

Takashi Washio
Einoshin Suzuki
Kai Ming Ting
Akihiro Inokuchi (Eds.)

LNAI 5012

Advances in Knowledge Discovery and Data Mining

12th Pacific-Asia Conference, PAKDD 2008
Osaka, Japan, May 2008
Proceedings

 Springer

Lecture Notes in Artificial Intelligence 5012

Edited by R. Goebel, J. Siekmann, and W. Wahlster

Subseries of Lecture Notes in Computer Science

Takashi Washio Einoshin Suzuki
Kai Ming Ting Akihiro Inokuchi (Eds.)

Advances in Knowledge Discovery and Data Mining

12th Pacific-Asia Conference, PAKDD 2008
Osaka, Japan, May 20-23, 2008
Proceedings

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

Takashi Washio
Akihiro Inokuchi
Osaka University
The Institute of Scientific and Industrial Research
8-1 Mihogaoka, Ibaraki, Osaka 567-0047, Japan
E-mail: {washio, inokuchi}@ar.sanken.osaka-u.ac.jp

Einoshin Suzuki
Kyushu University
Graduate School of Information Science and Electrical Engineering
Department of Informatics
744 Motooka, Nishi, Fukuoka 819-0395, Japan
E-mail: suzuki@i.kyushu-u.ac.jp

Kai Ming Ting
Monash University
Gippsland School of Information Technology
Gippsland Campus, Churchill, Victoria 3842, Australia
E-mail: kaiming.ting@infotech.monash.edu.au

Library of Congress Control Number: 2008926516

CR Subject Classification (1998): I.2, H.2.8, H.3, H.5.1, G.3, J.1, K.4

LNCS Sublibrary: SL 7 – Artificial Intelligence

ISSN 0302-9743
ISBN-10 3-540-68124-8 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-68124-3 Springer Berlin Heidelberg New York

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.

Springer is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2008
Printed in Germany

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 12270828 06/3180 5 4 3 2 1 0

Preface

The Pacific-Asia Conference on Knowledge Discovery and DataMining (PAKDD) has been held every year since 1997. PAKDD 2008, the 12th in the series, was held at Osaka, Japan during May 20–23, 2008. PAKDD is a leading international conference in the area of data mining. It provides an international forum for researchers and industry practitioners to share their new ideas, original research results, and practical development experiences from all KDD-related areas including data mining, data warehousing, machine learning, databases, statistics, knowledge acquisition, automatic scientific discovery, data visualization, causal induction, and knowledge-based systems.

This year we received a total of 312 research papers from 34 countries and regions in Asia, Australia, North America, South America, Europe, and Africa. Every submitted paper was rigorously reviewed by two or three reviewers, discussed by the reviewers under the supervision of an Area Chair, and judged by the Program Committee Chairs. When there was a disagreement, the Area Chair and/or the Program Committee Chairs provided an additional review. Thus, many submissions were reviewed by four experts. The Program Committee members were deeply involved in a highly selective process. As a result, only approximately 11.9% of the 312 submissions were accepted as long papers, 12.8% of them were accepted as regular papers, and 11.5% of them were accepted as short papers.

The PAKDD 2008 conference program also included four workshops. They were a workshop on Algorithms for Large-Scale Information Processing in Knowledge Discovery (ALSIP 2008), a workshop on Web Mining and Web-Based Application 2008 (WMWA 2008), a workshop on Data Mining for Decision Making and Risk Management (DMDRM 2008), and a workshop on Interactive Data Mining (IDM 2008). PAKDD 2008 would not have been successful without the support of many people and organizations. We wish to thank the members of the Steering Committee for their invaluable suggestions and support throughout the organization process. We are indebted to the Area Chairs, Program Committee members, and external reviewers for their effort and engagement in providing a rich and rigorous scientific program for PAKDD 2008. We wish to express our gratitude to our General Workshop Co-chairs Sharma Chakravarthy and Sanjay Chawla for selecting and coordinating the exciting workshops, to our Tutorial Co-chairs Achim Hoffmann and Akihiro Yamamoto for coordinating the fruitful tutorials, and to the distinguished keynote speakers, invited speakers, and tutorial presenters for their wonderful talks and lectures. We are also grateful to the Local Arrangement Chair Takashi Okada, the Local Arrangement Co-chairs Katsutoshi Yada and Kouzou Ohara, and the Local Arrangement Committee, whose great effort ensured the success of the conference.

We greatly appreciate the support from various institutions. The conference was organized by the Institute of Scientific and Industrial Research, Osaka University, Osaka, Japan and co-organized by the School of Science and Technology, Kwansai Gakuin University, Hyogo, Japan and the Faculty of Commerce, Kansai University, Osaka, Japan in cooperation with the Japanese Society of Artificial Intelligence. It was sponsored by Osaka Convention and Tourism Bureau, Commemorative Organization for the Japan World Exposition '70, Kayamori Foundation of Informational Science, the Air Force Office of Scientific Research/Asian Office of Aerospace Research and Development (AFOSR/AOARD), Future Systems Corp., Salford Systems, and Mathematical Systems Inc.

We also want to thank all authors and all conference participants for their contribution and support. We hope all participants took this opportunity to share and exchange ideas with one another and enjoyed PAKDD 2008.

March 2008

Takashi Washio
Einoshin Suzuki
Kai Ming Ting

Organization

PAKDD 2008 Organization Committee

General Co-chairs

Shusaku Tsumoto
Huan Liu

Shimane University, Japan
Arizona State University, USA

Program Committee Chair

Takashi Washio

Osaka University, Japan

Program Committee Co-chairs

Einoshin Suzuki
Kai Ming Ting

Kyushu University, Japan
Monash University, Australia

Local Arrangements Chair

Takashi Okada

Kwansei Gakuin University, Japan

Local Arrangements Co-chairs

Katsutoshi Yada
Kouzou Ohara

Kansai University, Japan
Osaka University, Japan

Workshop Co-chairs

Sharma Chakravarthy
Sanjay Chawla

The University of Texas at Arlington, USA
University of Sydney, Australia

Tutorial Co-chairs

Achim Hoffmann

University of New South Wales in Sydney,
Australia

Akihiro Yamamoto

Kyoto University, Japan

Publicity Chair

Kouzou Ohara

Osaka University, Japan

Publication Chair

Akihiro Inokuchi

Osaka University, Japan

PAKDD 2008 Conference Steering Committee

Chair

David Cheung
Rao Kotagiri

University of Hong Kong, China
University of Melbourne, Australia (Life-long member)

Treasurer

Graham Williams

Australian Taxation Office, Australia

Members

Arbee L.P. Chen
Ming-Syan Chen
Tu Bao Ho

National Chengchi University, Taiwan
National Taiwan University, Taiwan
Japan Advanced Institute of Science and Technology, Japan

Masaru Kitsuregawa
Huan Liu
Ee-Peng Lim
Hiroshi Motoda
Jaideep Srivastava
Takao Terano
Kyu-Young Whang

University of Tokyo, Japan
Arizona State University, USA
Nanyang Technological University, Singapore
AFOSR/AOARD, Japan (Life-long member)
University of Minnesota, USA
Tokyo Institute of Technology, Japan
Korea Advanced Institute of Science and Technology, Korea

Chengqi Zhang
Ning Zhong
Zhi-Hua Zhou

University of Technology Sydney, Australia
Maebashi Institute of Technology, Japan
Nanjing University, China

PAKDD 2008 Program Committee

Chair

Takashi Washio

Osaka University, Japan

Co-chairs

Einoshin Suzuki
Kai Ming Ting

Kyushu University, Japan
Monash University, Australia

Area Chairs

Naoki Abe
Luc De Raedt
Tapio Elomaa
Johannes Fürnkranz
Joao Gama

IBM T.J. Watson Research Center, USA
Katholieke Universiteit Leuven, Belgium
Tempere University of Technology, Finland
TU Darmstadt, Germany
LIACC-University Porto, Portugal

Tu Bao Ho	Japan Advanced Institute of Science and Technology, Japan
Achim Hoffmann	The University of New South Wales, Australia
Eamonn Keogh	University of California, Riverside, USA
Wee Keong Ng	Nanyang Technological University, Singapore
Hang Li	Microsoft Research Asia, China
Jinyan Li	Institute for Infocomm Research, Singapore
Hiroshi Motoda	AFOSR/AOARD and Osaka University, Japan
Jian Pei	Simon Fraser University, Canada
Hannu Toivonen	University of Helsinki, Finland
Wei Wang	University of North Carolina at Chapel Hill, USA
Charles X. Ling	University of Western Ontario, Canada
Qiang Yang	Hong Kong University of Science and Technology, China
Mohammed Zaki	Rensselaer Polytechnic Institute, USA
Zhi-Hua Zhou	Nanjing University, China

Members

David Albrecht	Diane Cook
Aijun An	Bruno Cremilleux
Vo Ngoc Anh	Alfredo Cuzzocrea
Hiroki Arimura	Dao-Qing Dai
Michael Bain	Honghua Dai
Hideo Bannai	Floriana Esposito
Michael Berthold	Wei Fan
Hendrik Blockeel	Eibe Frank
Francesco Bonchi	Ada Waichee Fu
Ulf Brefeld	Dragan Gamberger
Rui Camacho	Junbin Gao
Longbing Cao	Fosca Giannotti
Tru Hoang Cao	Aristides Gionis
Sanjay Chawla	Bart Goethals
Arbee Chen	Shyam Kumar Gupta
Ming-Syan Chen	Sung Ho Ha
Phoebe Chen	Jiawei Han
Shu-Ching Chen	Shoji Hirano
Songcan Chen	Wynne Hsu
Yixin Chen	Xiaohua Hu
Zheng Chen	Fuchun Huang
William K. Cheung	Jimmy Huang
Yiu-ming Cheung	Jin Huang
Sungzoon Cho	Kaiqi Huang
Vic Ciesielski	San-Yih Hwang

Daisuke Ikeda
Akihiro Inokuchi
Sanjay Jain
Szymon Jaroszewicz
Daxin Jiang
LiCheng Jiao
Ye Jieping
Huidong Jin
Rong Jin
Ruoming Jin
Alipio M. Jorge
George Karypis
Hisashi Kashima
Hiroyuki Kawano
Boonserm Kijsirikul
Masaru Kitsuregawa
Marzena Kryszkiewicz
Ravi Kumar
James Kwok
Wai Lam
Jonathan Lawry
Sang Ho Lee
Vincent C.S. Lee
Wee Sun Lee
Yoon-Joon Lee
Philippe Lenca
Chun-hung Li
Gang Li
Jianzhong Li
Tao Li
Xiao-Lin Li
Xue Li
Xuelong Li
Chih-Jen Lin
Xuemin Lin
Tie-Yan Liu
Xiaohui Liu
Woong-Kee Loh
Chang-Tien Lu
Jixin Ma
Marco Maggini
Yutaka Matsuo
Sameep Mehta
Wagner Meira Jr.
Xiaofeng Meng

Taneli Mielikainen
Toshiro Minami
Pabitra Mitra
Yang-Sae Moon
Yasuhiko Morimoto
Tsuyoshi Murata
Atsuyoshi Nakamura
Richi Nayak
Wilfred Ng
Hung Son Nguyen
Ngoc Thanh Nguyen
Zaiqing Nie
Tadashi Nomoto
Zoran Obradovic
Kouzou Ohara
Salvatore Orlando
Satoshi Oyama
Sankar K. Pal
Yanwei Pang
Adrian Pearce
Dino Pedreschi
Wen-Chih Peng
Yonghong Peng
Jean-Marc Petit
Bernhard Pfahringer
Vincenzo Piuri
Joel Quinqueton
Naren Ramakrishnan
Sanjay Ranka
Patricia Riddle
Fabio Roli
Stefan Raping
Kenji Satou
Joern Schneidewind
Michele Sebag
Dou Shen
Jialie Shen
Yi-Dong Shen
Daming Shi
Zhongzhi Shi
Akira Shimazu
Masashi Shimbo
Arno Siebes
Andrzej Skowron
Mingli Song

Ashok Srivastava
 Aixin Sun Nanyang
 Ah-Hwee Tan Nanyang
 Chew Lim Tan
 Pang-Ning Tan
 Zhaohui Tang
 David Taniar
 Luis Torgo
 Ivor W. Tsang
 Tomoyuki Uchida
 Jeffrey D. Ullman
 Takeaki Uno
 Guoyin Wang
 Haixun Wang
 Hui Wang
 Huiqiong Wang
 Jason T.L. Wang
 Lipo Wang
 Wenjia Wang
 Zhihai Wang
 Graham Williams
 Limsoon Wong
 Qingxiang Wu
 Xindong Wu
 Xintao Wu
 Hui Xiong
 Zhuoming Xu
 Takehisa Yairi
 Seiji Yamada
 Chunsheng Yang
 Hui Yang
 Ying Yang

Min Yao
 Yiyu Yao
 Dit-Yan Yeung
 Jian Yin
 Kennichi Yoshida
 Tetsuya Yoshida
 Clement Yu
 Jeffrey Xu Yu
 Jian Yu
 Yuan Yuan
 Bo Zhang
 Changshui Zhang
 Chengqi Zhang
 Daoqiang Zhang
 Du Zhang
 Harry Zhang
 Junping Zhang
 Mengjie Zhang
 Shichao Zhang
 Weixiong Zhang
 Zhongfei (Mark) Zhang
 Zili Zhang
 Zijian Zheng
 Ning Zhong
 Sheng Zhong
 Aoying Zhou
 Huiyu Zhou
 Shuigeng Zhou
 Xiaofang Zhou
 Yan Zhou
 Xingquan Zhu

PAKDD 2008 External Reviewers

Michael Armella
 Mafruz Z. Ashrafi
 Anneleen Van Assche
 Yingyi Bu
 Bin Cao
 Huanhuan Cao
 Rattachat Chatpatanasiri
 Kasturi Chatterjee
 Jianhui Chen
 Tingting Chen

Kurt De Grave
 Fabien De Marchi
 Nicola Di Mauro
 Frédéric Flouvat
 Ling Guo
 Hao He
 Qi He
 Peter Hebden
 Shuiwang Ji
 Xing Jiang

Tom Johnsten	Liang Sun
Marius Kloft	Giorgio Terracina
Stephane Lallich	Quan Thanh Tho
Hui Liu	Joaquin Vanschoren
Yang Liu	Qian Wan
Yiming Ma	Raymond Wan
James Malone	Jinlong Wang
Nguyen Le Minh	Raymond Chi Wing Wong
Alberto Paccanaro	Tao Wu
Tanasanee Phienthrakul	Evan Xiang
Jiangtao Ren	Dan Xiao
Peng Ren	Ming Xu
Khalid Saleem	Ghim-Eng Yap
Ray Dos Santos	Xiaowei Ying
Baohong Shen	Huaifeng Zhang
Dou Shen	Ke Zhang
Al Shorin	Na Zhao
Junilda Spirollari	Yanchang Zhao
Anantaporn Srisawat	Victor Zou
Chad M.S. Steel	
Jan Struyf	

PAKDD 2008 Local Arrangements Committee

Chair

Takashi Okada Kwansei Gakuin University, Japan

Co-chairs

Katsutoshi Yada Kansai University, Japan
Kouzou Ohara Osaka University, Japan

Members

Noriaki Kawamae NTT, Japan
Koichi Moriyama Osaka University, Japan
Tomonobu Ozaki Kobe University, Japan

Organized by

I.S.I.R., Osaka University

Co-organized by

School of Science and Technology, Kwansei Gakuin University
Faculty of Commerce, Kansai University

In Cooperation with

The Japanese Society of Artificial Intelligence

Sponsored by

Osaka Convention & Tourism Bureau

Commemorative Organization for the Japan World Exposition '70

Kayamori Foundation of Informational Science

The Air Force Office of Scientific Research/Asian Office of Aerospace Research and Development (AFOSR/AOARD)

Future Systems Corp.

Salford Systems

Mathematical Systems Inc.



独立行政法人日本万国博覧会記念機構

Commemorative Organization for The Japan World Exposition '70

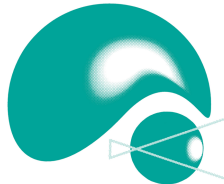




Table of Contents

Keynote Speech

Graph Mining: Laws, Generators and Tools	1
--	---

Invited Speeches

Efficient Algorithms for Mining Frequent and Closed Patterns from Semi-structured Data	2
--	---

Supporting Creativity: Towards Associative Discovery of New Insights	14
--	----

Cost-Sensitive Classifier Evaluation Using Cost Curves	26
--	----

Prospective Scientific Methodology in Knowledge Society	30
---	----

Long Papers

SubClass: Classification of Multidimensional Noisy Data Using Subspace Clusters	40
---	----

Mining Quality-Aware Subspace Clusters	53
--	----

A Decremental Approach for Mining Frequent Itemsets from Uncertain Data	64
---	----

Multi-class Named Entity Recognition Via Bootstrapping with Dependency Tree-Based Patterns	76
--	----

Towards Region Discovery in Spatial Datasets	88
--	----

Accurate and Efficient Retrieval of Multimedia Time Series Data Under Uniform Scaling and Time Warping	100
Feature Construction Based on Closedness Properties Is Not That Simple	112
On Addressing Accuracy Concerns in Privacy Preserving Association Rule Mining	124
Privacy-Preserving Linear Fisher Discriminant Analysis	136
Unsupervised Change Analysis Using Supervised Learning	148
ANEMI: An Adaptive Neighborhood Expectation-Maximization Algorithm with Spatial Augmented Initialization	160
Minimum Variance Associations — Discovering Relationships in Numerical Data	172
An Efficient Unordered Tree Kernel and Its Application to Glycan Classification	184
Generation of Globally Relevant Continuous Features for Classification	196
Mining Bulletin Board Systems Using Community Generation	209
Extreme Support Vector Machine Classifier	222
LCM over ZBDDs: Fast Generation of Very Large-Scale Frequent Itemsets Using a Compact Graph-Based Representation	234
Unusual Pattern Detection in High Dimensions	247
Person Name Disambiguation in Web Pages Using Social Network, Compound Words and Latent Topics	260

Mining Correlated Subgraphs in Graph Databases	272
A Minimal Description Length Scheme for Polynomial Regression	284
Handling Numeric Attributes in Hoeffding Trees	296
Scaling Record Linkage to Non-uniform Distributed Class Sizes	308
Large-Scale k-Means Clustering with User-Centric Privacy Preservation	320
Semi-Supervised Local Fisher Discriminant Analysis for Dimensionality Reduction	333
An Efficient Algorithm for Finding Similar Short Substrings from Large Scale String Data	345
Ambiguous Frequent Itemset Mining and Polynomial Delay Enumeration	357
Characteristic-Based Descriptors for Motion Sequence Recognition	369
Protecting Privacy in Incremental Maintenance for Distributed Association Rule Mining	381
SEM: Mining Spatial Events from the Web	393
BOAI: Fast Alternating Decision Tree Induction Based on Bottom-Up Evaluation	405
Feature Selection by Nonparametric Bayes Error Minimization	417
A Framework for Modeling Positive Class Expansion with Single Snapshot	429

A Decomposition Algorithm for Learning Bayesian Network Structures from Data 441

Learning Classification Rules for Multiple Target Attributes 454

A Mixture Model for Expert Finding 466

On Privacy in Time Series Data Mining 479

Regular Papers

Exploiting Propositionalization Based on Random Relational Rules for Semi-supervised Learning 494

On Discrete Data Clustering 503

Automatic Training Example Selection for Scalable Unsupervised Record Linkage 511

Analyzing PETs on Imbalanced Datasets When Training and Testing Class Distributions Differ 519

Improving the Robustness to Outliers of Mixtures of Probabilistic PCAs 527

Exploratory Hot Spot Profile Analysis Using Interactive Visual Drill-Down Self-Organizing Maps 536

Maintaining Optimal Multi-way Splits for Numerical Attributes in Data Streams 544

Efficient Mining of High Utility Itemsets from Large Datasets 554

Tradeoff Analysis of Different Markov Blanket Local Learning Approaches 562

Forecasting Urban Air Pollution Using HMM-Fuzzy Model	572
Relational Pattern Mining Based on Equivalent Classes of Properties Extracted from Samples	582
Evaluating Standard Techniques for Implicit Diversity	592
A Simple Characterization on Serially Constructible Episodes	600
Bootstrap Based Pattern Selection for Support Vector Regression	608
Tracking Topic Evolution in On-Line Postings: 2006 IBM Innovation Jam Data	616
PAID: Packet Analysis for Anomaly Intrusion Detection	626
A Comparison of Different Off-Centered Entropies to Deal with Class Imbalance for Decision Trees	634
FISViz: A Frequent Itemset Visualizer	644
A Tree-Based Approach for Frequent Pattern Mining from Uncertain Data	653
Connectivity Based Stream Clustering Using Localised Density Exemplars	662
Learning User Purchase Intent from User-Centric Data	673
Query Expansion for the Language Modelling Framework Using the Naïve Bayes Assumption	681

Fast Online Estimation of the Joint Probability Distribution	689
Fast Most Similar Neighbor Classifier for Mixed Data Based on Approximating and Eliminating.....	697
Entity Network Prediction Using Multitype Topic Models	705
Using Supervised and Unsupervised Techniques to Determine Groups of Patients with Different Doctor-Patient Stability	715
Local Projection in Jumping Emerging Patterns Discovery in Transaction Databases	723
Applying Latent Semantic Indexing in Frequent Itemset Mining for Document Relation Discovery	731
G-TREACLE: A New Grid-Based and Tree-Alike Pattern Clustering Technique for Large Databases	739
A Clustering-Oriented Star Coordinate Translation Method for Reliable Clustering Parameterization	749
Constrained Clustering for Gene Expression Data Mining	759
Concept Lattice-Based Mutation Control for Reactive Motifs Discovery	767
Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases	777
Designing a System for a Process Parameter Determined through Modified PSO and Fuzzy Neural Network	785

Data-Aware Clustering Hierarchy for Wireless Sensor Networks	795
A More Topologically Stable Locally Linear Embedding Algorithm Based on R^* -Tree	803
Sparse Kernel-Based Feature Weighting	813
Term Committee Based Event Identification within News Topics	821
Locally Linear Online Mapping for Mining Low-Dimensional Data Manifolds	830
A Creditable Subspace Labeling Method Based on D-S Evidence Theory	839
Short Papers	
Discovering New Orders of the Chemical Elements through Genetic Algorithms	849
What Is Frequent in a Single Graph?	858
A Cluster-Based Genetic-Fuzzy Mining Approach for Items with Multiple Minimum Supports	864
A Selective Classifier for Incomplete Data	870
Detecting Near-Duplicates in Large-Scale Short Text Databases	877
Customer Churn Time Prediction in Mobile Telecommunication Industry Using Ordinal Regression	884
Rule Extraction with Rough-Fuzzy Hybridization Method	890
I/O Scalable Bregman Co-clustering	896

Jumping Emerging Patterns with Occurrence Count in Image Classification	904
Mining Non-coincidental Rules without a User Defined Support Threshold	910
Transaction Clustering Using a Seeds Based Approach	916
Using Ontology-Based User Preferences to Aggregate Rank Lists in Web Search	923
The Application of Echo State Network in Stock Data Mining	932
Text Categorization of Multilingual Web Pages in Specific Domain	938
Efficient Joint Clustering Algorithms in Optimization and Geography Domains	945
Active Learning with Misclassification Sampling Using Diverse Ensembles Enhanced by Unlabeled Instances	951
A New Model for Image Annotation	958
Unmixed Spectrum Clustering for Template Composition in Lung Sound Classification	964
Forward Semi-supervised Feature Selection	970
Automatic Extraction of Basis Expressions That Indicate Economic Trends	977
A New Framework for Taxonomy Discovery from Text	985
R-Map: Mapping Categorical Data for Clustering and Visualization Based on Reference Sets	992

Mining Changes in Patent Trends for Competitive Intelligence	999
Seeing Several Stars: A Rating Inference Task for a Document Containing Several Evaluation Criteria	1006
Structure-Based Hierarchical Transformations for Interactive Visual Exploration of Social Networks	1015
CP-Tree: A Tree Structure for Single-Pass Frequent Pattern Mining	1022
Combining Context and Existing Knowledge When Recognizing Biological Entities – Early Results	1028
Semantic Video Annotation by Mining Association Patterns from Visual and Speech Features	1035
Cell-Based Outlier Detection Algorithm: A Fast Outlier Detection Algorithm for Large Datasets	1042
Fighting WebSpam: Detecting Spam on the Graph Via Content and Link Features	1049
A Framework for Discovering Spatio-temporal Cohesive Networks	1056
Efficient Mining of Minimal Distinguishing Subgraph Patterns from Graph Databases	1062
Combined Association Rule Mining	1069
Enriching WordNet with Folksonomies	1075
A New Credit Scoring Method Based on Rough Sets and Decision Tree	1081

Analyzing the Propagation of Influence and Concept Evolution in Enterprise Social Networks through Centrality and Latent Semantic Analysis	1090
Author Index	1099

Graph Mining: Laws, Generators and Tools

Christos Faloutsos

Carnegie Mellon University
5000 Forbes Ave,
Pittsburgh PA 15213, USA
`christos@cs.cmu.edu`

Extended Abstract

How do graphs look like? How do they evolve over time? How can we generate realistic-looking graphs? We review some static and temporal 'laws', and we describe the "Kronecker" graph generator, which naturally matches all of the known properties of real graphs. Moreover, we present tools for discovering anomalies and patterns in two types of graphs, static and time-evolving. For the former, we present the 'CenterPiece' subgraphs (CePS), which expects q query nodes (eg., suspicious people) and finds the node that is best connected to all q of them (eg., the master mind of a criminal group). We also show how to compute CenterPiece subgraphs efficiently. For the time evolving graphs, we present tensor-based methods, and apply them on real data, like the DBLP author-paper dataset, where they are able to find natural research communities, and track their evolution.

Finally, we also briefly mention some results on influence and virus propagation on real graphs.

Biographical Note

Christos Faloutsos is a Professor at Carnegie Mellon University. He has received the Presidential Young Investigator Award by the National Science Foundation (1989), the Research Contributions Award in ICDM 2006, ten "best paper" awards, and several teaching awards. He has served as a member of the executive committee of SIGKDD; he has published over 160 refereed articles, 11 book chapters and one monograph. He holds five patents and he has given over 20 tutorials and over 10 invited distinguished lectures. His research interests include data mining for streams and networks, fractals, indexing for multimedia and bioinformatics data, and database performance.

Efficient Algorithms for Mining Frequent and Closed Patterns from Semi-structured Data

Hiroki Arimura

Hokkaido University, Kita 14-jo, Nishi 9-chome, Sapporo 060-0814, Japan
arim@ist.hokudai.ac.jp

Abstract. In this talk, we study efficient algorithms that find frequent patterns and maximal (or closed) patterns from large collections of semi-structured data. We review basic techniques developed by the authors, called the *rightmost expansion* and the *PPC-extension*, respectively, for designing efficient frequent and maximal/closed pattern mining algorithms for large semi-structured data. Then, we discuss their applications to design of polynomial-delay and polynomial-space algorithms for frequent and maximal pattern mining of sets, sequences, trees, and graphs.

1 Introduction

By rapid progress of high-speed networks and large-scale storage technologies, a huge amount of electronic data of new types, called *semi-structured data*, have emerged in the late 1990s. Web Pages, XML documents, and genome data are typical examples of such semi-structured data. Therefore, there have been potential demands for efficient methods that extract useful information from these semi-structured data.

Traditionally, data mining mainly deals with well-structured data, e.g., transaction databases or relational databases, for which data is arranged in a table-like regular structure. On the other hand, these semi-structured data are (i) huge, (ii) heterogeneous collections of (iii) weakly-structured data that do not have rigid structures. Thus, we cannot directly apply these traditional data mining technologies to semi-structured data. Hence, our goal is to develop efficient methods that discover interesting or useful rules from large collections of semi-structured data, namely, [\[13, 15, 17, 18, 20, 21, 25, 29\]](#).

In this paper, we present efficient semistructured data mining algorithms for discovering rules and patterns from structured data such as sequence, trees, and graphs. In Section 2, we consider tree mining and sequence mining in the framework of frequent pattern mining. We present the rightmost expansion technique [\[9, 11, 17\]](#). Then, in Section 3, we extend them to closed or maximal pattern mining by the PCC-expansion technique [\[3, 4, 5, 7\]](#), where each pattern is a representative of an equivalence class of patterns having the same occurrences in a given database. Finally, in Section 4, we conclude.

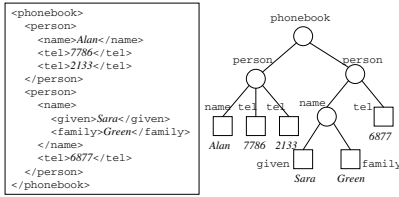


Fig. 1. An XML document (left) as a labeled ordered tree (right)

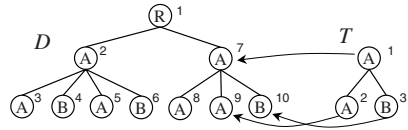


Fig. 2. A data tree D and a pattern tree T on the label set $\mathcal{L} = \{A, B\}$

2 Efficient Frequent Pattern Mining Algorithms

In this section, we introduce a framework of frequent pattern mining for semi-structured data and present efficient tree mining algorithms.

2.1 Framework of Semi-structured Data Mining

A general framework. In our framework, a semi-structured data mining problem is specified by a 4-tuple $(\mathcal{P}, \mathcal{T}, \mathcal{O}, L)$, where \mathcal{P} is a class of \dots , \mathcal{T} is a class of \dots , \mathcal{O} is a class of \dots (or \dots), and $L : \mathcal{P} \times \mathcal{D} \rightarrow 2^{\mathcal{O}}$ is a function called \dots , where $L(P, T) = \{o_1, \dots, o_n\} \subseteq \mathcal{O}$ is the set of all occurrences of a pattern $P \in \mathcal{P}$ in a given database $T \in \mathcal{T}$. We also assume that a partial order \sqsubseteq over patterns, called the \dots order (or the \dots) such that for every patterns P, Q , $P \sqsubseteq Q$ implies $L(P, T) \supseteq L(Q, T)$. If $P \sqsubseteq Q$ then we say that P subsumes Q or P is more general than Q . For most classes of semi-structured patterns, $L(\cdot, \cdot)$ and \sqsubseteq are defined by the notion of matching or embedding of patterns.

Now, we define the frequent pattern mining problem for \mathcal{P} as the problem of, given a database $T \in \mathcal{T}$ and a minimum frequency threshold $0 \leq \sigma \leq |T|$, finding all frequent patterns $P \in \mathcal{P}$ appearing in T such that $|L(P, T)| \geq \sigma$ without duplicates.

As the theoretical framework to study the computational complexity of semi-structured data mining, we adopt the theory of enumeration algorithms. Hence, our goal is to design \dots [16], where the \dots is the maximum computation time between the successive outputs and the \dots is the total time divided by the number of output patterns.

Ordered and unordered trees. For example, we gives the definition of frequent ordered tree mining. In tree mining, data and patterns are modeled by \dots as shown in Fig. 2. Let Σ be a countable alphabet of symbols. A labeled ordered tree is a rooted tree $T = (V_T, E_T, \leq_T, root_T, lab_T)$, where $V = V_T$ is the set of \dots , $E_T \subseteq V^2$ is a set of \dots called the \dots , $\leq_T \subseteq V^2$ is a binary relation called the \dots , which orders children of each internal node left to right, $root_T \in V$ is the root of T , and $lab_T : V \rightarrow \Sigma$ is a function called the \dots . A \dots

is a rooted tree $T = (V_T, E_T, root_T, lab_T)$, where the order \leq_T of children is not relevant. We denote by \mathcal{OT} and \mathcal{UT} the classes of labeled ordered trees and of labeled unordered trees.

For a ordered trees P and T , P matches Q , denoted by $P \sqsubseteq Q$, if there exists a matching function $\phi : V_P \rightarrow V_T$ from P to T that satisfies the following conditions (i) – (iv): (i) ϕ is one-to-one; (ii) ϕ preserves the parent-child relation; (iii) ϕ preserves the sibling relation; (iv) ϕ preserves the node label. We denote by $\Phi(P, T)$ the set of all matching functions from P to T . An occurrence list of a pattern tree P in a text tree T is the set of the occurrences of P in T defined by $L(P, T) = \{ \phi(root_P) : \phi \in \Phi(P, T) \}$. Then, the support of P is defined by the number of occurrences $1 \leq |L(P, T)| \leq |T|$.

2.2 Rightmost Expansion Technique for Frequent Pattern Mining

For frequent ordered tree mining problem, we developed algorithm FREQT [9] that finds all frequent ordered tree patterns in a database. One of the keys of the algorithm is efficient enumeration of labeled ordered trees. Our FREQT, as well as TREEMINER by Zaki [34], are in the first generation of depth-first tree and graph mining algorithms. The key of these algorithms is the rightmost expansion technique explained below, which is independently proposed by our group [9], Nakano [23], and Zaki [34].

A basic idea of the method is to build a spanning tree G over the search space of ordered tree patterns, called a family tree for labeled ordered trees as shown in Fig. 3. By using the family tree, we can enumerate all the distinct ordered tree patterns without duplicates in a unique way starting at the root pattern called the empty tree \perp and expanding (or growing) an already generated tree of size $k - 1$ (a tree of size $k - 1$) by attaching a new node to yield larger tree of size k (a tree of size k).

However, a straightforward implementation of this idea leads exponential number of the duplication for one tree resulting G to be a DAG. The rightmost expansion [9, 23, 34] is a technique to avoid duplicates, where we grow a pattern tree by attaching a new node to only the rightward positions on the rightmost branch of the parent tree so that the attached node becomes the rightmost leaf of the resulting tree.

This is equivalent to encode each labeled ordered tree T with n nodes by the sequence $code(T) = (X_1, \dots, X_n)$, called the depth-label sequence, where for each $1 \leq i \leq n$, $X_i = (depth_i, lab_i) \in \mathbf{N} \times \Sigma$ are the pair of the depth and the label of the i -th nodes of T in the preorder traversal of T . Then, we grow a tree by attaching a new depth-label pair at the tail of $code(T)$ [11, 23, 34]. Optimization techniques such as occurrence-deliver [31] and duplicate detection achieve significant speed-up of the order of magnitude.

2.3 Frequent Unordered Tree Miner UNOT

For frequent unordered tree mining problem, we developed algorithm UNOT [11] that finds all frequent unordered tree patterns in a database. Some real-world

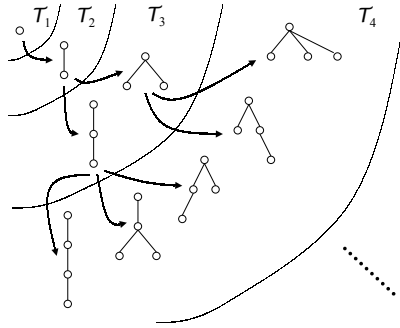


Fig. 3. A family tree for (unlabeled) ordered trees

applications requires more general classes of graph patterns than ordered trees. However, from theory point of view, graph mining with general graph patterns seems intractable due to the NP-completeness of the subgraph isomorphism problem for general graphs. Our algorithm UNOT, and the algorithm independently proposed by Nijssen and Kok [26], are ones of the first output-polynomial time tree/graph mining algorithms for nontrivial subclasses of graphs larger than ordered trees, which finds all frequent unordered tree patterns from a given collections of trees in time polynomial time per pattern [11].

A difficulty comes from the fact that an unordered tree can have exponentially many equivalent representations as ordered trees (Fig. 4). To overcome this difficulty, we introduced the unique canonical representation of an unordered tree, called a left-heavier tree representation has a monotonicity that if an unordered tree $T(a, \dots)$ is left-heavier then the tree S obtained from T (the \dots) by removing the rightmost leaf is also left-heavier. Thus, We developed an efficient method to enumerate such canonical representation without duplicates [11, 24] by generalizing rightmost expansion technique of [1].

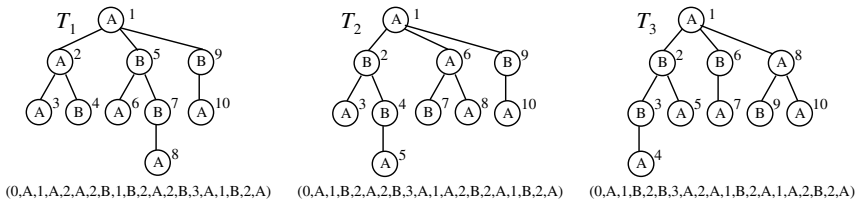


Fig. 4. Equivalent unordered trees. Tree T_3 is the canonical representation among three equivalent trees.

2.4 Applications of Frequent and Optimized Pattern Mining

\dots is a variant of frequent pattern mining, where a database is a collection $D = \{T_1, \dots, T_m\} \subseteq \mathcal{T}$ of data together with labeling function $\chi : D \rightarrow \{+1, -1\}$ that assigns a binary classification label $\chi(T_i)$ to

each text $T_i \in D$ and the goal is to find a best pattern $P \in \mathcal{P}$ that optimizes a given statistical score function such as the classification error or the Shannon entropy in D . In weighted frequent pattern mining, we optimize the sum $score(P, D, \chi) = \sum_{T \in L(P, D)} \chi(T)$.

An interesting application of the optimized tree miners are feature extraction for statistical machine learning over semi-structured data, such as boosting [19], SVM, statistical clustering [30] for tree and graph structures. Our tree mining algorithm FREQT is applied to text mining from natural language texts for log analysis at the call center and customer services [22], where a collection of Japanese sentences are transformed into a collection of labeled unordered trees by lexical and dependency analyses, and then FREQT is applied to find top- K best patterns in the MDL measure [22].

3 Efficient Maximal Pattern Mining Algorithms

3.1 Maximal Pattern Discovery

Maximal pattern discovery (also known as frequent maximal pattern discovery [27]) is one of the most important topics in recent studies of data mining [4, 5, 7, 8, 12, 27, 29, 33, 32]. A frequent maximal pattern is such a pattern that is maximal with respect to the subsumption ordering (or the generalization relation) among an equivalence class of patterns having the same set of occurrences in a database. For some known classes of patterns, such as itemsets and sequence motifs [3], it is known that the number of frequent maximal patterns is much smaller than that of frequent patterns on most realworld datasets, while the frequent maximal patterns still contain the complete information of the frequency of all frequent patterns. Thus, Maximal pattern discovery is useful to increase the performance and the comprehensivity of data mining.

Formally, for a class \mathcal{P} of patterns, we define the associated class \mathcal{C} of frequent maximal patterns (also known as frequent maximal patterns in [27]) as follows [31, 4, 5, 7]. Recall that the class of maximal patterns \mathcal{C} is specified by a 4-tuple $(\mathcal{P}, \mathcal{T}, \mathcal{O}, L)$ of a pattern class \mathcal{P} , a database class \mathcal{T} , an occurrence class, and an occurrence mapping $L : \mathcal{P} \times \mathcal{T} \rightarrow 2^{\mathcal{O}}$. We define patterns P and Q are equivalent each other, denoted by $P \equiv Q$, iff $L(P, T) = L(Q, T)$ holds, and the equivalence class for pattern P by $[P] = \{Q \in \mathcal{P} : P \equiv Q\}$. Then, a pattern P is frequent in a database T if P is a maximal element in $[P]$ w.r.t. \sqsubseteq . Equivalently, P is maximal if there exists no strictly more specific pattern $Q \in \mathcal{P}$ than P equivalent to P , i.e., $P \sqsubset Q$ and $L(P, T) = L(Q, T)$ hold.

3.2 Depth-First Algorithms for Maximal Pattern Discovery

Efficient Algorithms. For maximal pattern discovery, we have developed the following efficient algorithms for finding all maximal patterns from a given collection of data. Let Σ be an alphabet of symbols.

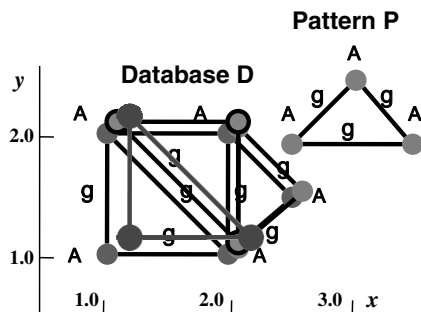


Fig. 5. Geometric Graph Mining, where patterns and a database are graphs having points in the 2-D plane as their vertices, and are invariant under translation, rotation, and enlargement

- The algorithm LCM (Labeled Closed Mining) [31] (Fig. 6), where a closed set [27], also called a *closed itemset* here, is an itemset over Σ which is maximal w.r.t. set inclusion among those itemsets having the same set of *supporting transactions* in a given databases.
- The algorithm MAXMOTIF (Maximal Motif Mining) [4] for mining maximal motifs with wildcards [28]. A motif with wildcards [28] is a sequence of constant symbols in Σ and special wildcards \circ for single letters such as $AB\circ B\circ ABC$.
- The algorithm CLOATT (Closed Labeled Attribute Tree Mining) [5] for mining maximal trees [29]. An attribute tree [29] is a labeled unordered tree where out-going edges starting from any internal node mutually distinct symbols in Σ as their labels. This class can be seen as a fragment of description logic having deterministic roles only.
- The algorithm MAXGEO (Maximal Geometric Graph Mining) [7] for mining maximal geometric graphs [7]. A geometric graph is an undirected graph whose vertices are points in the 2-D plane \mathbf{R}^2 and labeled with symbols in Σ , as shown in Fig. 5. The matching among geometric graphs is defined to be invariant under a set of geometric transformations such as translation, rotation, and enlargement.

All of the above algorithms are polynomial time polynomial space algorithms based on depth-first search. To achieve this requirement, we developed the PPC-extension (prefix-preserving closure extension) technique below.

Reverse search technique. Our PPC-extension can be viewed as an instance of the *reverse search* technique [16], which is a technique for designing efficient enumeration algorithms for complex combinatorial objects, such as perfect matching and maximal cliques. Let \mathcal{S} be the set of solutions on an instance of a given combinatorial enumeration problem. In reverse search, for every non-root solutions Y , we arbitrarily assign the parent $X = Pa(X)$ in a systematic way by using a *parent function* Pa . The mapping Pa is designed so that $Pa(P)$ is uniquely determined and the *rank* of P is properly decreasing. Then, it follows

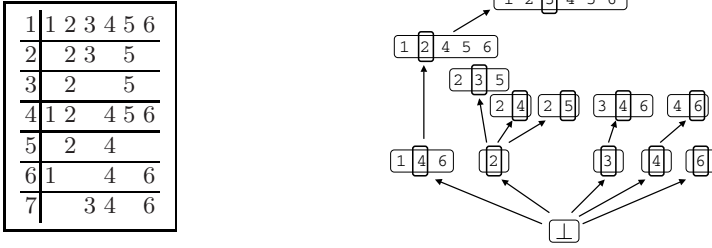


Fig. 6. An example of a transaction database (left) and maximal (closed) item sets on the database generated by the PPC-extension (right), where \perp is the smallest closed set and each arrow indicates the parent-child relationship according to the PPC-extension. A small box on an item of each closed set indicates newly added item as a seed of the PPC-extension.

that the directed graphs with \mathcal{S} as the set of vertices and Pa as the set of directed edges, called the \perp -spanning tree, forms a spanning tree over the solutions. Finally, we can apply depth-first search to enumerate all solutions by starting from the root and then by expanding the current solution to obtain its children. If a child no longer belongs to \mathcal{S} , then we backtrack to its parent and continue the search. Repeating the above process, we can enumerate all the solutions in \mathcal{S} in depth-first manner with small delay and small memory.

3.3 PPC-Extension for Maximal Semi-structured Patterns

In this subsection, we will present the PPC-extension framework for designing an efficient maximal pattern mining algorithm for a given class \mathcal{P} of semi-structured patterns. In particular, we give an algorithmic schema PPC-MaxMine, from which we can derive polynomial delay and polynomial time enumeration algorithms LCM [31], MAXMOTIF [4], CLOATT [5], and MAXGEO [7] for maximal pattern mining.

Merge and Closure operations. For the class of maximal patterns \mathcal{C} specified by a 4-tuple $(\mathcal{P}, \mathcal{T}, \mathcal{O}, L)$, we require that any set S of patterns has the unique greatest lower bound $\sqcap S$ of S w.r.t. \sqsubseteq , or the least common subsumer (LCS) of all patterns in S . Although this property of the existence of the unique LCS does not hold in general, the four classes of patterns that we have considered, i.e., the classes of itemsets, motifs with wildcards, attribute trees, and geometric graphs, enjoy this property.

Under the assumption of the unique LCS, we give the merge operation $\oplus : \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{P}$ that computes the least common subsumer $Q_1 \oplus Q_2 \in \mathcal{P}$ of a pair of patterns $Q_1, Q_2 \in \mathcal{P}$. We suppose that a pattern P occurs in a database $T \in \mathcal{T}$ at occurrence or position $p \in L(P, T) \in \mathcal{O}$. Then, an occurrence or position of T at the occurrence p , denoted by $(T - p) \in \mathcal{T}$, is a copy of the data T where

¹ This graph sometimes has several roots, and then called a spanning forest.

the origin of the copy is set to the position p . Then, we define the p -closure of a possibly non-maximal pattern P by the pattern $Clo_T(P) = \bigoplus_{p \in \mathcal{L}(P)} (T - p)$.

Lemma 1.

$$\begin{aligned}
 & \mathcal{P} \text{ is closed w.r.t. } T \iff T \in \mathcal{P} \\
 & P, Q \in \mathcal{P} \\
 & \bullet P \sqsubseteq Q \iff Clo_T(P) \sqsubseteq Clo_T(Q) \\
 & Clo_T(Clo_T(P)) = Clo_T(P) \\
 & \bullet P \sqsubseteq Q \iff P \sqsubseteq Q, \text{ iff } L(P, T) \supseteq L(Q, T) \\
 & \bullet P \sqsubseteq Q \iff T, \text{ iff } Clo_T(P) = P
 \end{aligned}$$

Defining the parent function. Now, we define the parent function for the maximal patterns in \mathcal{C} by using the closure operator Clo_T as follows. Firstly, we introduce an adequate representation scheme for patterns of \mathcal{P} , where each pattern P of size n is encoded by a sequence $code(P) = (X_1, \dots, X_n)$ over an alphabet \mathcal{X} of components. Then, we assume that the encoding is prefix-closed, i.e., for any pattern P , any prefix of the encode of P is a proper encoding of some pattern in \mathcal{P} . The increasing sequence representation for itemsets and the depth-label sequence representations are examples of such encoding schema. In what follows, we identify a pattern P and its encoding $code(P) = (X_1, \dots, X_n)$ if it is clear from the context. If $P = (X_1, \dots, X_n)$, $1 \leq k \leq n$, and $Z \in \mathcal{X}$ then we define the insertion of Z at the index k by $P[k \leftarrow Z] = (X_1, \dots, X_{k-1}, Z, X_k, \dots, X_n)$.

We define the root pattern by $\perp = Clo_T(\varepsilon) \in \mathcal{C}$, which is the closure of the empty pattern ε that appears everyposition in T . Let $P = (X_1, \dots, X_n) \in \mathcal{C}$ be any non-root pattern that is maximal in T . For $k \leq n$, we denote by $P(k) = (X_1, \dots, X_k)$ the prefix of P with length k . The k -*fix* of P in T is the shortest prefix $P(k)$ ($0 \leq k \leq n$) of P that has the same occurrence list with the original, i.e., $L(P(k), T) = L(P, T)$ holds. Then, i is called the *critical index* of P w.r.t. T and denoted by $crit_idx(P) = i$. We define the *parent* of pattern Q in T by

$$Pa(Q) = Clo_T(Q(k-1)),$$

where $k = crit_idx(Q)$ is the critical index of Q w.r.t. T . Then, we can observe that since $Q(k)$ is the shortest prefix of Q having the same set of occurrences, the strictly shorter prefix $Q(k-1)$ has a properly larger set of occurrence $L(Q(k-1))$ ($L(Q) \subset L(Q(k-1))$). By the definition of the equivalence relation \equiv , we know that $Q(k-1)$ now belongs to an equivalence class $[Q(k-1)] \subseteq \mathcal{Q}$ disjoint with the previous equivalence class $[Q]$ for Q . Thus, it follows by Property 4 of Lemma 1 that the closure $P = Clo_T(Q(k-1)) \in \mathcal{P}$ of $Q(k-1)$ is a maximal pattern in \mathcal{C} . By this construction, for every non-root maximal pattern $Q \in \mathcal{C}$, we always associate as its parent the maximal pattern $P = Pa(Q) \in \mathcal{C}$. Furthermore, we can easily see from Lemma 1 that P is more general than Q , and furthermore, P is strictly shorter than Q in length. Combining the above discussion, we can see that the rooted directed graph $G = (\mathcal{C}, Pa, \perp)$ with the set of vertices \mathcal{C} , the set of reverse edges $Pa \subseteq \mathcal{C} \times \mathcal{C}$, and the root \perp is a *spanning tree* for \mathcal{C} , a spanning tree for all maximal patterns in \mathcal{C} .

Algorithm PPC-MaxMine(\mathcal{X} : component alphabet, $T \in \mathcal{T}$: database, $\sigma \geq 0$: min-freq):

- 1 global: T, σ ;
- 2 PPC-Expand($\perp, 0$);

Procedure PPC-Expand(P, i):

- 1 if $|L(P, T)| < \sigma$ then return; //in-frequent
- 2 Output P ;
- 3 for any $X \in \mathcal{X}$ and any $k > i$ such that $P[k \leftarrow X] \in \mathcal{P}$ do begin
- 4 $Q = Clo_T(P[k \leftarrow X])$; //PPC-extension: closure computation
- 5 if $P(k-1) = R(k-1)$ then //PPC-extension: prefix check
- 6 PPC-Expand(Q, k);
- 7 end

Fig. 7. An algorithm scheme PPC-MaxMine for enumerating all maximal patterns in \mathcal{P} in a database

Depth-first mining by PPC-extension. The remaining thing is to perform the depth-first search over the family tree G for \mathcal{C} by inverting the direction of the reverse edges in Pa . Suppose that P is a parent of Q , i.e., $P = Pa(Q) = Clo_T(Q(k-1))$ for the critical index $k = crit_idx(Q)$ and $Z = X_k \in \mathcal{X}$ is the k -th component of Q . Then, we can show that we can recover Q from P by computing $Q = Clo_T(P[k \leftarrow Z])$ provided that the , . fb , . , $P(k-1) = Q(k-1)$ succeeds, i.e., the prefixes of P and Q of length $k-1$ coincide, where $P[k \leftarrow Z]$ is the encoding obtained from P by inserting Z at position k . Then, Q is said to be a PPC-extension of P . Furthermore, for different selection of $(k, Z) \in \mathbf{N} \times \mathcal{X}$ generates distinct children of P .

Fig 6 shows the search structure of PPC-extension in LCM algorithm in the case of maximal itemset mining 31. In Fig. 7, we give a simple backtracking algorithm PPC-MaxMine based on the PPC-extension technique. The algorithm PPC-MaxMine finds all maximal patterns in the class \mathcal{C} appearing in T without duplicates in the depth-first manner.

Complexity analysis. From the view from enumeration algorithms, we showed that all instances of the algorithm scheme PPC-MaxMine, namely, LCM 31, MAXMOTIF 4, CLOATT 5, and MAXGEO 7, are actually polynomial delay and polynomial time enumeration algorithms that computes all maximal patterns for the classes of itemsets, motifs with wildcards, attribute trees, and geometric graphs, respectively. To our knowledge, these are the first results on efficient output-sensitive algorithms for maximal pattern discovery for semi-structured data.

4 Conclusion

In this talk, we reviewed techniques for designing efficient frequent and maximal/closed mining algorithms for large semi-structured data. We capture the

notion of high-throughput and light-weight pattern mining algorithms by the class of polynomial-delay and polynomial space enumeration algorithms for pattern mining problems. Firstly, we study efficient frequent tree mining based on the rightmost expansion technique developed by [9,23,34], which enable us to efficiently enumerate complex semi-structured patterns in a systematic way without duplicates. Then, we present frequent mining algorithms FREQT [9], OPTT [1], and UNOT [11]. Secondly, we study efficient maximal/closed pattern mining for classes sets, sequences, trees, and graphs based on the prefix-preserving closure extension (PPC-extension) technique. Based on this technique, we present frequent mining algorithms LCM [31], MAXMOTIF [4], CLOATT [5], and MAX-GEO [7]. All the above algorithms that we presented have polynomial-delay and polynomial space enumeration complexity for the corresponding semi-structured data mining problems.

It is an interesting future problem to apply the above frameworks for other classes of complex pattern mining problems such as maximal/closed pattern mining for sequential patterns (sequences of itemsets) or generalized itemsets (sets of elements of a concept hierarchy). On maximal/closed pattern mining, we mainly considered the pattern classes of “ $\langle \cdot, \cdot, \cdot, \dots, \cdot \rangle$ ”, e.g., items sets, motifs with wild cards and attribute trees, where the LCS and the closure are uniquely determined from the occurrences. On the other hands, there are a few results on efficient maximal/closed pattern mining for the classes of “ $fl \langle \cdot, \cdot, \cdot \rangle$ ” patterns such as ordered trees, unordered trees, and subsequences patterns for which no closure operator is known. Thus, it is a future problem to study a generic framework for mining maximal flexible patterns.

Acknowledgment

The results presented in this talk are obtained in the joint works with Takeaki Uno, Shin-ichi Nakano, Tatsuya Asai, Hiroshi Sakamoto, Shinji Kawasoe, Kenji Abe, and Yuzo Uchida. The author would like to express sincere thanks to them. This research was partly supported by the Ministry of Education, Science, Sports and Culture, Grant-in-Aid for Specially Promoted Research, 17002008, 2007 on “semi-structured data mining”, and the MEXT/JSPS Global COE Program, “Center for Next-Generation Information Technology based on Knowledge Discovery and Knowledge Federation,” at Graduate School of Information Science and Technology, Hokkaido University.

References

1. Abe, K., Kawasoe, S., Asai, T., Arimura, H., Arikawa, S.: Optimized substructure discovery for semi-structured data. In: Elomaa, T., Mannila, H., Toivonen, H. (eds.) PKDD 2002. LNCS (LNAI), vol. 2431, pp. 1–14. Springer, Heidelberg (2002)
2. Abiteboul, S., Buneman, P., Suciu, D.: Data on the Web. Morgan Kaufmann, San Francisco (2000)

3. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: *Advances in Knowledge Discovery and Data Mining*, ch. 12, AAAI Press / The MIT Press (1996)
4. Arimura, H., Uno, T.: An Efficient Polynomial Space and Polynomial Delay Algorithm for Enumeration of Maximal Motifs in a Sequence. Special issue on bioinformatics, *Journal of Combinatorial Optimization* 13, 243–262 (2006)
5. Arimura, H., Uno, T.: An output-polynomial time algorithm for mining frequent closed attribute trees. In: Kramer, S., Pfahringer, B. (eds.) *ILP 2005*. LNCS (LNAI), vol. 3625, pp. 1–19. Springer, Heidelberg (2005)
6. Arimura, H., Uno, T.: A Polynomial Space and Polynomial Delay Algorithm for Enumerating Maximal Two-Dimensional Patterns with Wildcards. Technical Report TCS-TR-A-06-19, Division of Computer Science, Hokkaido University, July 18 (2006)
7. Arimura, H., Uno, T., Shimozono, S.: Time and Space Efficient Discovery of Maximal Geometric Graphs. In: Corruble, V., Takeda, M., Suzuki, E. (eds.) *DS 2007*. LNCS (LNAI), vol. 4755, Springer, Heidelberg (2007)
8. Arimura, H., Uno, T.: Mining Maximal Flexible Patterns in a Sequence. In: *Proc. 5th Workshop on Learning with Logics and Logics for Learning (LLLL 2007)*, 25th JSAI 2007. LNCS (LNAI), vol. 4914, Springer, Heidelberg (2008)
9. Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., Arikawa, S.: Efficient Substructure Discovery from Large Semi-structured Data. In: *Proc. the 2nd SIAM Int'l Conf. on Data Mining (SDM 2002)*, pp. 158–174 (2002)
10. Asai, T., Arimura, H., Abe, K., Kawasoe, S., Arikawa, S.: Online algorithms for mining semi-structured data stream. In: *Proc. ICDM 2002*, pp. 27–34 (2002)
11. Asai, T., Arimura, H., Uno, T., Nakano, S.: Discovering frequent substructures in large unordered trees. In: Grieser, G., Tanaka, Y., Yamamoto, A. (eds.) *DS 2003*. LNCS (LNAI), vol. 2843, Springer, Heidelberg (2003)
12. Chi, Y., Yang, Y., Xia, Y., Muntz, R.R.: Cmtreeminer: Mining both closed and maximal frequent subtrees. In: Dai, H., Srikant, R., Zhang, C. (eds.) *PAKDD 2004*. LNCS (LNAI), vol. 3056, Springer, Heidelberg (2004)
13. Cong, G., Yi, L., Liu, B., Wang, K.: Discovering Frequent Substructures from Hierarchical Semi-structured Data. In: *Proc. SIAM SDM (2002)*
14. Freund, Y., Schapire, R.E.: A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* 55(1), 119–139 (1997)
15. Dehaspe, L., De Raedt, L.: Mining association rules with multiple relations. In: Džeroski, S., Lavrač, N. (eds.) *ILP 1997*. LNCS, vol. 1297, pp. 125–132. Springer, Heidelberg (1997)
16. Avis, D., Fukuda, K.: Reverse Search for Enumeration. *Discrete Applied Mathematics* 65(1–3), 21–46 (1996)
17. Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: mining graph data. *Machine Learning* 50(3), 321–354 (2003)
18. Kuramochi, M., Karypis, G.: Frequent Subgraph Discovery. In: *Proc. ICDM 2001 (2001)*
19. Kudo, T., Matsumoto, Y.: A Boosting Algorithm for Classification of Semi-Structured Text. In: *Proc. of EMNLP*, pp. 301–308 (2004)
20. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovering Frequent Episode in Sequences. In: *Proc. KDD 1995*, pp. 210–215 (1995)
21. Miyahara, T., Shoudai, T., Uchida, T., Takahashi, K., Ueda, H.: Discovery of Frequent Tree Structured Patterns in Semistructured Web Documents. In: Cheung, D., Williams, G.J., Li, Q. (eds.) *PAKDD 2001*. LNCS (LNAI), vol. 2035, pp. 47–52. Springer, Heidelberg (2001)

22. Morinaga, S., Arimura, H., Ikeda, T., Sakao, Y., Akamine, S.: Key Semantics Extraction by Dependency Tree. In: Proc. KDD 2005, pp. 666–671. ACM, New York (2005)
23. Nakano, S.: Efficient generation of plane trees. *Information Processing Letters* 84, 167–172 (2002)
24. Nakano, S., Uno, T.: Constant time generation of trees with specified diameter. In: Hromkovič, J., Nagl, M., Westfechtel, B. (eds.) WG 2004. LNCS, vol. 3353, pp. 33–45. Springer, Heidelberg (2004)
25. Nestrov, S., Abiteboul, S., Motwani, R.: Extracting Schema from Semistructured Data. In: Proc. SIGKDD 1998, pp. 295–306 (1998)
26. Nijssen, S., Kok, J.N.: Efficient Discovery of Frequent Unordered Trees. In: Proc. the 1st Int’l Workshop on Mining Graphs, Trees and Sequences (MGTS 2003) (September 2003)
27. 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)
28. Pisanti, N., Crochemore, M., Grossi, R., Sagot, M.-F.: A basis of tiling motifs for generating repeated patterns and its complexity for higher quorum. In: Rovan, B., Vojtáš, P. (eds.) MFCS 2003. LNCS, vol. 2747, pp. 622–631. Springer, Heidelberg (2003)
29. Termier, A., Rousset, M.-C., Sebag, M.: TreeFinder: a First Step towards XML Data Mining. In: Proc. IEEE ICDM 2002, pp. 450–457 (2002)
30. Tsuda, K., Kudo, T.: Clustering graphs by weighted substructure mining. In: ICML 2006, pp. 953–960 (2006)
31. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An efficient algorithm for enumerating closed patterns in transaction databases. In: Suzuki, E., Arikawa, S. (eds.) DS 2004. LNCS (LNAI), vol. 3245, pp. 16–30. Springer, Heidelberg (2004)
32. Yan, X., Han, J., Afshar, R.: CloSpan: mining closed sequential patterns in large databases. In: Proc. SDM 2003, SIAM, Philadelphia (2003)
33. Wang, J., Han, J.: BIDE: efficient mining of frequent closed sequences. In: Proc. ICDE 2004 (2004)
34. Zaki, M.J.: Efficiently mining frequent trees in a forest. In: Proc. SIGKDD 2002, ACM Press, New York (2002)

Supporting Creativity: Towards Associative Discovery of New Insights

Michael R. Berthold, Fabian Dill, Tobias Kötter, and Kilian Thiel

University of Konstanz, Fach M712, 78484 Konstanz, Germany
Michael.Berthold@uni-Konstanz.de

Abstract. In this paper we outline an approach for network-based information access and exploration. In contrast to existing methods, the presented framework allows for the integration of both semantically meaningful information as well as loosely coupled information fragments from heterogeneous information repositories. The resulting Bisociative Information Networks (BisoNets) together with explorative navigation methods facilitate the discovery of links across diverse domains. In addition to such “chains of evidence”, they enable the user to go back to the original information repository and investigate the origin of each link, ultimately resulting in the discovery of previously unknown connections between information entities of different domains, subsequently triggering new insights and supporting creative discoveries.

Keywords: BisoNet, Bisociative Information Networks, Bisociation, Discovery Support Systems.

1 Motivation: The Need for Information Exploration

Data collection and generation methods continue to increase their ability to fill up information repositories at an alarming rate. In many industries it is nowadays commonly accepted – although often not openly admitted – that only a fraction of available information is taken into account when making decisions or trying to uncover interesting, potentially crucial links between previously unconnected pieces of information.

In order to allow users to be able to find important pieces of information it is necessary to replace classical question answering systems with tools that allow for the interactive exploration of potentially related information – which can often trigger new insights and spark new ideas which the user did not expect at start and was therefore unable to formulate as a query initially. It is especially crucial for such systems to enable the seamless crossing of repository boundaries to trigger new discoveries across domains. Since we will not know at the start which types of information are needed or which kind of questions will be asked throughout this explorative process, the system always needs to be able to provide access to heterogeneous information repositories. These can be structured, well annotated repositories, such as an ontology or a database of

human annotations (“known facts”) but it needs to incorporate other types of information as well, such as experimental data or the vast amounts of results from the mining of e.g. published texts (“pieces of evidence”). The real challenge lies in providing the user with easy access to all of this information so that she can quickly discard uninteresting paths to information that is not currently relevant and at the same time focus on areas of interest. Similar to drill down operations in Visual Data Mining, such a system will need to be able to show summarizations according to different dimensions or levels of detail and allow parallel changes of focus to enable the user to ultimately navigate to the information entities that explain the connections of interest. Of course, the system cannot be static but will require not only means for continuous updating of the underlying information repositories to accommodate new data, but also new and better methods to extract connections. In [1] we have argued that such a system will truly support the discovery of new insights. Related work investigating the nature of creativity (see [2] among others) describes similar requirements for creative discoveries, based on broad but at the same time context dependent more focused exploration of associations as the underlying backbone.

In this paper we outline an approach to realize such a system using a network-based model to continuously integrate and update heterogeneous information repositories and at the same time allow for explorative access to navigate both semantic and evidential links. Before describing our prototypical system in more detail we review existing network-based systems for knowledge or information modeling. We conclude the paper by discussing open issues and challenges.

2 State of the Art: Network-Based Information Access

Different network-based models have been applied to Information Retrieval, such as artificial neural networks, probabilistic inference networks, Hopfield or knowledge networks [3]. The first two are mainly used to match documents to queries and to find relevant documents related to a certain query. Documents and index terms, which are the most discriminative terms, are represented as vertices in these networks. Edges can be created to connect documents citing each other, documents with their index terms, as well as cooccurring index terms. Hopfield and knowledge networks are additionally used for automatic thesaurus creation and consultation [4]. In this case only vertices of index terms cooccurring in documents or sentences are connected via edges. Another connectionist approach, Adaptive Information Retrieval (AIR), creates additional vertices for each document author and connects them by their author co-author relationships [5,6].

The majority of these approaches use weighted networks. In these networks a weight is assigned to each edge, which depends on the underlying network model as well as the computation and interpretation of the relation. In probabilistic inference networks the weights represent probabilities of terms occurring in documents being relevant to a certain query [3,7]. Whereas the weights of knowledge or Hopfield networks as discussed in [4] represent the relatedness of cooccurring terms. Usually the weights of these approaches are only computed once and not

changed afterwards. In contrast to these approaches, Belew enables each user of an AIR model to adapt the weights according to their relevance feedback [5]. After initialization of the weights where the edges between documents and terms are weighted with the term's inverse document frequency, a user can send queries to the network. The user then rates the resulting nodes, representing terms, documents or authors, as relevant or irrelevant. This relevance feedback is passed to the network again in order to adjust the edge weight and process another query. This kind of iterative process is continued until the result fits the users needs. One essential disadvantage of such an adaptive system is that it adapts to the user's opinion of which documents are more relevant than others related to a certain query. This means that the network will, over time, be strongly biased by the opinion of the majority of the users.

In a number of other domains, networks have been applied to combine, represent, integrate and analyze information, such as bioinformatics and life science, with a strong emphasis on the extraction of pharmacological targets [8], protein functions [9], gene-gene [10], gene-protein [11] or protein-protein interactions [12,13] from different biological databases and biomedical literature [14]. To mine texts and find this kind of interaction Blaschke et al. [12] proposed to parse the sentences into grammatical units. Patterns or regular expressions have been used as well to extract genes, proteins and their relations in texts [10,13].

Once the units of information and their relations are found, they can be represented in a network. Additional algorithms can be used to cluster and analyze these networks in order to identify meaningful subnetworks (communities) [15,13]. The analysis of network structures also reveals new insights into complex processes such as regulator strategies in yeast cells [16]. Additionally the edges can be evaluated and their quality can be specified based on several features like edge reliability, relevance and rarity [17]. Note that also the increasingly popular social networks fall into this category. In general much work has been done when it comes to methods for network analysis [18].

2.1 Adaptive and Explorative Approaches

To visually analyze graphs, different layout algorithms such as the force-directed Fruchterman-Reingold algorithm [19] have been developed. But large networks with several million vertices and many more edges cannot be visualized completely in a reasonable manner. Therefore the visualization has to be focused on a subgraph or at least summarized to match the current user's interest or give an overview. Various visualization techniques have been developed to address this problem. Examples are the generalized Fisheye views [20], the splitting of a network into several smaller semantical distinct regions [21] or the interactive navigation through different levels of abstractions [22].

Another way to analyze large networks is to extract subgraphs that contain most of the relevant information. One way to do this is to query a graph. On the one hand queries can be generated by manually drawing a sub-graph or by using a particular query language, i.e. GenoLink [23]. The results of such queries are represented as sub-graphs which themselves could be the starting point of further

analyses. On the other hand Spreading Activation techniques are very common techniques to explore networks and handle queries [24]. In general the idea of activity spreading is based on assumed mechanisms of human cognitive memory operations, originated from psychological studies [25]. These techniques are adopted to many different areas such as Cognitive Science, Databases, Artificial Intelligence, Psychology, Biology and Information Retrieval. The basic activity spreading technique is quite simple. First, one or more vertices, representing the query terms, are activated. The initial activation is distributed (spread) over the outgoing edges and activates in subsequent iterations the adjacent vertices. This iterative process will continue until a certain termination condition, such as a maximum number of activated nodes or iterations or a minimum edge or vertex weight is reached. The activation itself can also be weighted and can decrease over time or when propagating over certain edges. Furthermore different activation functions can be used for the vertices [24]. In [4] the networks are explored by usage of a branch-and-bound search and a Hopfield net activation. Due to the restriction that a Hopfield activation algorithm only guarantees to converge if the graph's adjacency matrix is symmetric, meaning that the graph is undirected, this technique is only applicable for certain kinds of networks. Other approaches cope with the complexity by clustering or pruning the graph based on their topology [26] or based on additional information such as a given ontology [27].

2.2 Combining Heterogeneous Information Repositories

The integration of heterogeneous data sources facilitates insights across different domains. Such insights are important especially in complex application areas such as life sciences, which deal with different kinds of data, e.g. gene expression experiments, gene ontologies, scientific literature, expert notes, etc. During the last few years several approaches have been developed that attempt to tackle this problem. The authors of [28] classified these systems into three general classes: navigational integration, mediator-based integration and warehouse integration.

Navigational integration approaches like SRS [29], Entrez [30] and LinkDB [20] aim to integrate heterogeneous data by providing links between units of information derived from different sources. Links can be created based on database entries as well as on the similarity of the units of information, or manually by experts [20]. Most of the applications consist of one or more indexed flat files containing the relations between the different concepts.

The second category is the mediator-based integration systems such as DiscoveryLink [31], BioMediator [32], Kleisli [33] and its derivatives like TAMBIS [34] or K2 [35]. These systems act as a mediator, which maps the schema of different data sources onto a unified schema. Each query is converted and split up into a set of sub-queries, which are then redirected to the wrapper of the integrated data source. Finally the results of the sub-queries are combined to a single result and returned by the mediator.

Warehouse approaches like GUS [35], Atlas [36], BIOZON [37] and BNDB [38] are similar to the mediator-based approach since they also provide a unified

schema for all data sources. But instead of creating a sub-query for each data source the data itself is loaded into the unified schema.

Navigational integration and mediator-based approaches do not integrate all the detailed data of a concept. The amount and complexity to handle additional data is much smaller in comparison to systems that integrate the detailed information of a concept like the warehouse approach. The advantage of this kind of light integration is the ability to keep the detailed information up to date since it is stored in the external sources itself. The drawback of such an integration is the dependency on all the integrated systems with respect to reliability and performance. In contrast, the warehouse approach also integrates all the detailed information from the distributed repositories. The data can be preprocessed and enriched with additional information such as similarity measures or user annotations. The local storage of all data leads to a better performance and system reliability. However the huge amount of data itself and continued maintenance to detect changes and inconsistencies are the major drawback of such systems.

In summary, warehouse and mediator-based approaches provide the user with a unified, mostly relational schema. This allows professional users the ability to use powerful query languages like SQL to perform complex joins and queries. The unification leads mostly to a complex data model including link tables to combine the different data sources. Navigational approaches only maintain link information between concepts and provide simple point and click interfaces visualizing links between them. These interfaces are also manageable by semi professional users but restricted in their query capabilities like the lack of complex joins. A common goal of all the mentioned integration approaches is the combination of equal or similar concepts from different data sources. An obvious approach to link these concepts is the usage of a flexible graph structure. An example of integrating high confidence biological data is PathSys [39]. PathSys is a graph-based data warehouse, which is used to analyze relations between genes and proteins. To predict protein-protein interactions several approaches adopted Bayesian Networks to model the mostly noisy or uncorrelated evidences of biological experiments [40,41].

3 BisoNets: Bisociative Information Networks

As we have suggested above, simply finding classical associations is not sufficient to detect interesting connections across different information repositories and contexts. Existing systems either tend to be to application focussed or restricted to only a few type of information sources or types. However, in order to support creative discoveries across domains we cannot assume that we know from the beginning which information repositories will need to be combined in which way.

In 1964 Arthur Koestler introduced the term *bisociation* [42] to indicate the process of connecting two previously unconnected concepts. Using this terminology we use the term Bisociative Information Networks, or short *BisoNets* to denote a type of information network addressing the above concerns, fusing the following requirements:

- **Heterogeneous Information:** BisoNets integrate information from various information repositories, representing both semantically solid knowledge (such as from an ontology or a human annotated semantic net) and imprecise and/or unreliable knowledge such as derived from automatic analysis methods (e.g. results from text mining or association rule analyses) or other experimental results (e.g. correlations derived from protein expression experiments).
- **Merging Evidence and Facts:** BisoNets provide a unifying mechanism to combine these different types of information and assign and maintain edge weights and annotations in order to allow the mixing of links with different degrees of certainty.
- **Continuous Update:** BisoNets can be refined online and continuously integrate updated or new information.
- **Exploration/Navigation:** Finally, in order to allow access to the resulting information structure, BisoNets provide explorative navigation methods, which show summarizations of (sub-) networks, and allow the changing of focus and quick zooming operations.

There is strong evidence that such a complex system of loosely, not necessarily semantically coupled information granules exhibits surprisingly sophisticated features. In [43] Hecht-Nielsen describes a network which generates grammatically correct and semantically meaningful sentences purely based on links created from word co-occurrence without any additional syntactical or semantical analysis. In addition, [2] discusses requirements for creativity, supporting this type of domain bridging bisociations.

3.1 First Steps: A BisoNet Prototype

In order to evaluate the concept of BisoNets, we have implemented a first prototype and so far have mainly applied it to life science related data. However, the toolkit is not restricted to this type of data. The BisoNet prototype creates one vertex for each arbitrary unit of information, i.e. a gene or protein name, a specific molecule, an index term or a document, and other types of named entities. Relations between vertices are represented by edges. Vertices are identified by their unique name and edges by the vertices they connect. In order to model not only facts but also more or less precise pieces of evidence, edges are weighted to reflect the degree of certainty and specificity of the relation.

Due to the uniqueness of a vertex name, a vertex can be ambiguous and represent different units of information, i.e. a vertex can represent a term extracted from a document and a gene or protein name derived from a certain database. For example a vertex could represent the animal “jaguar” or the make of car. To distinguish the different kinds of meanings, an annotation can be applied to vertices and edges. An annotation specifies the origin and the type of the information unit. A vertex representing different units of information will contain different annotations: one annotation for each meaning. Edges with different annotations represent relations derived from different data sources. Each

annotation of an edge contains its own weight in order to specify the evidence of the relation according to the data sources it was derived from.

The structure of the knowledge network is rather lightweight, that is it simply consists of vertices and edges, but contains no detailed information of the vertices or edges itself. In order to access this valuable, more detailed information as well, so-called data agents have been implemented. For each annotation, representing a particular kind of information of a certain data source, a data agent is available, which can be used to access the corresponding data source and extract the detailed information for a particular vertex or edge annotation.

To analyze and explore the network in order to find new and hopefully useful information, potentially uninteresting information has to be filtered. The prototype provides several filtering methods. One method allows particular annotation types of vertices and edges to be hidden, such as terms, species, or chemical compounds to focus on a specific context. Another one filters edges by their weight to filter out all relations below a certain degree of evidence. To extract information related to a particular issue, an activity spreading algorithm has been implemented, similar to the branch-and-bound algorithm of [4], which is able to extract subgraphs consisting of the most relevant vertices related to a specified set of initially activated vertices.

We implemented the BisoNet prototype within the modular information mining platform KNIME [44] due to the large set of data preprocessing and analysis methods available already. Each procedure and algorithm dealing with the network was implemented as a module or KNIME node respectively. This allows them to be used and combined individually and networks can be created, analyzed and explored in a flexible manner. Figure 1 shows an example KNIME workflow in which a network was created consisting of PubMed [45] abstracts as text data, gene subgroup information derived from gene expression data, gene-gene interaction data from Genetwork [46] and Gene Ontology [47] information. One by one all data sources are integrated into the network and at the end of the pipeline various filters can be applied to concentrate on a particular subgraph.

To visualize the network we used Cytoscape [48] an open source software platform for graph visualization. Note that this graph visualization toolkit does not offer sophisticated means to navigate the underlying BisoNet.

To create the complete network PubMed abstracts, related to the drug Plavix, treating thrombotic events, were analyzed and all content bearing index terms, gene and compound names were extracted and inserted into the network as vertices. Co-occurring terms above a certain frequency are connected by an edge. In addition gene-gene interaction data of Genetwork was integrated and, by applying different filters such as gene annotation filter or edge weight filter, the subgraph shown in Figure 2 can be extracted. The graph consists of 27 vertices representing gene names and 33 edges representing gene-gene interactions. The green vertices stem from the Genetwork data, the brown vertices from PubMed text data. In the subgraph illustrated in Figure 2 the four genes derived from text data connect and supplement the gene subgraphs of the Genetwork data nicely. Note how connections between subgraphs based on one data source are connected by information derived from a second source.

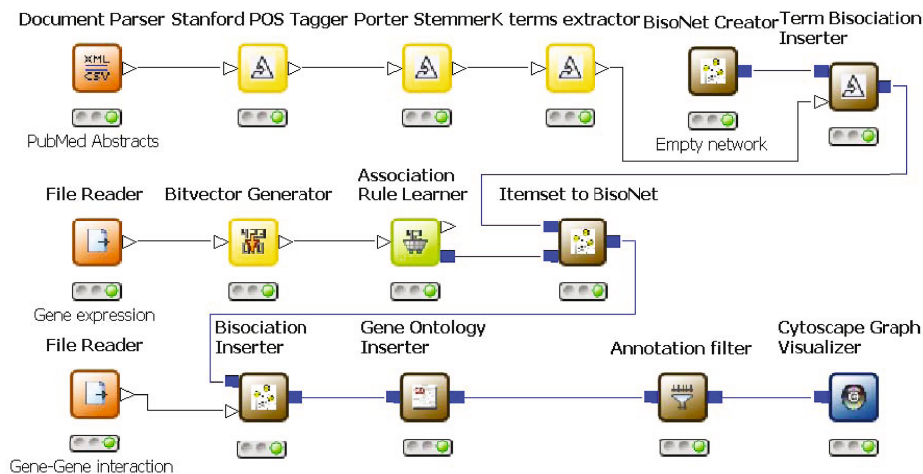


Fig. 1. A KNIME workflow which creates a network consisting of text and gene data. See text for details.

3.2 Open Issues and Challenges

The BisoNet prototype as described above is a first attempt at implementing the concepts listed in Section 1. Many open issues and challenges are still awaiting solutions and usable realizations. Within the EU Project “BISON” many of these challenges will be tackled over the coming years, focussing among others on issues related to:

- Scalability: addressing problems related to the increasing size of the resulting networks demanding new approaches for the storage, access, and subgraph operations on distributed representations of very large networks,
- Weight and Network Aggregation: that is, issues related to information sources of vastly different context and levels of certainty but also presumably simple problems of different versions of the same information repository, which also requires dealing with outdated information.
- Graph Abstraction: relating to methods that are especially crucial for problems related to exploration and navigation. In order to support zoom in and out operations, we need sophisticated methods for graph summarization and abstraction allowing for the offering, creation, and formalization of different views along different dimensions and at different levels of granularity on (sub) graphs.
- Disambiguation: that is, the differentiation of named entities with different meaning will also be critical to avoid nonsensical paths. Some of this will manifest automatically by supporting links of different domains but some means of at least semi automatic detection of ambiguous terms will be needed.

Without doubt, many other issues will be encountered along the way and soon cognitive issues will also become increasingly important, i.e., developing interfaces

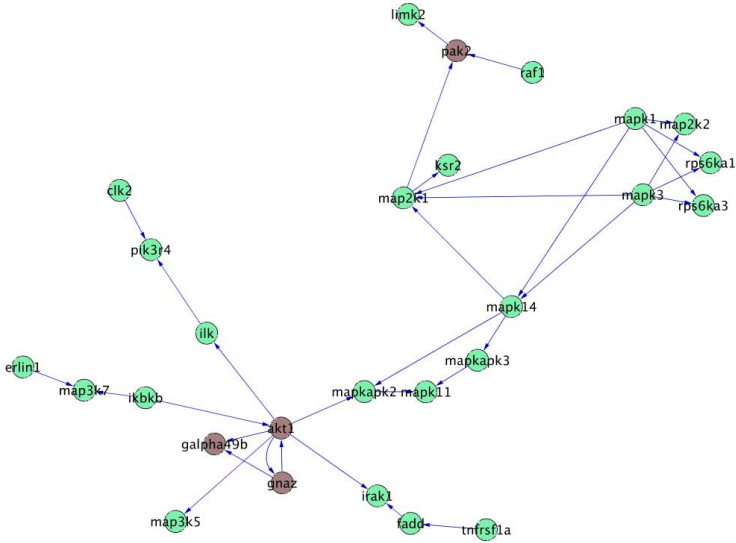


Fig. 2. A gene subgraph extracted from a network. See text for details.

that are adopted to the way humans think and work and therefore truly support human creativity instead of asking the user to adopt to the way the system has been designed.

4 Summary

In this paper we have outlined a new approach to support associative information access, enabling the user to find links across different information repositories and contexts. The underlying network combines pieces of information of various degrees of precision and reliability and allows for the exploration of both connections and original information fragments. We believe these types of bisociative information networks are a promising basis for the interactive exploration of loosely connected, semi- or unstructured information repositories, ultimately leading to fully fledged Discovery Support Systems.

We would like to thank the members of the European Framework 7 project BISON for many stimulating discussions, which have helped to refine the concept of BisoNets.

References

1. Berthold, M.R., Nürnbergger, A.: Towards associative information access. In: Proceedings of AISB 2006 (Adaptation in Artificial and Biological Systems), Society for the Study of Artificial Intelligence and the Simulation of Behaviour, University of Bristol, UK, vol. 3, pp. 98–101 (2006)

2. Sternberg, R.J. (ed.): *Handbook of Creativity*. Cambridge University Press, Cambridge (1999)
3. Cunningham, S., Holmes, G., Littin, J., Beale, R., Witten, I.: Applying connectionist models to information retrieval. In: Amari, S., Kasobov, N. (eds.) *Brain-Like Computing and Intelligent Information Systems*, pp. 435–457. Springer, Heidelberg (1997)
4. Chen, H., Ng, T.: An algorithmic approach to concept exploration in a large knowledge network (automatic thesaurus consultation): symbolic branch-and-bound search vs. connectionist hopfield net activation. *J. Am. Soc. Inf. Sci.* 46, 348–369 (1995)
5. Belew, R.K.: Adaptive information retrieval: using a connectionist representation to retrieve and learn about documents. In: *SIGIR 1989: Proceedings of the 12th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 11–20. ACM Press, New York (1989)
6. Belew, R.K. (ed.): *Finding Out About: A Cognitive Perspective on Search Engine Technology and the WWW*, 1st edn. Cambridge University Press, Cambridge (2000)
7. Fuhr, N.: Probabilistic models in information retrieval. *The Computer Journal* 35(3), 243–255 (1992)
8. Paolini, G., Shapland, R., van Hoorn, W., Mason, J., Hopkins, A.: Global mapping of pharmacological space. *Nature Biotechnology* 24, 805–815 (2006)
9. Sharan, R., Ulitsky, I., Shamir, R.: Network-based prediction of protein function. *Mol Syst Biol* 3, 88 (2007)
10. Natarajan, J., Berran, D., Dubitzky, W., Hack, C., Zhang, Y., DeSesa, C., Brocklyn, J.R.V., Bremer, E.G.: Text mining of full-text journal articles combined with gene expression analysis reveals a relationship between sphingosine-1-phosphate and invasiveness of a glioblastoma cell line. *BMC Bioinformatics* 7, 373 (2006)
11. Chiang, J.H., Yu, H.C.: Meke: discovering the functions of gene products from biomedical literature via sentence alignment. *Bioinformatics* 19(11), 1417–1422 (2003)
12. Blaschke, C., Andrade, M.A., Ouzounis, C., Valencia, A.: Automatic extraction of biological information from scientific text: protein-protein interactions. In: *Proc. Int. Conf. Intell. Syst. Mol. Biol.*, pp. 60–67 (1999)
13. Hu, X., Wu, D.D.: Data mining and predictive modeling of biomolecular network from biomedical literature databases. *IEEE/ACM Trans Comput Biol Bioinform* 4(2), 251–263 (2007)
14. Roberts, P.M.: Mining literature for systems biology. *Brief Bioinform* 7(4), 399–406 (2006)
15. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45, 167 (2003)
16. Lee, T.I., Rinaldi, N.J., Robert, F., Odom, D.T., Bar-Joseph, Z., Gerber, G.K., Hannett, N.M., Harbison, C.T., Thompson, C.M., Simon, I., Zeitlinger, J., Jennings, E.G., Murray, H.L., Gordon, D.B., Ren, B., Wyrick, J.J., Tagne, J.B., Volkert, T.L., Fraenkel, E., Gifford, D.K., Young, R.A.: Transcriptional regulatory networks in *saccharomyces cerevisiae*. *Science* 298, 799–804 (2002)
17. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: *Leser, U., Naumann, F., Eckman, B. (eds.) DILS 2006. LNCS (LNBI)*, vol. 4075, pp. 35–49. Springer, Heidelberg (2006)
18. Brandes, U., Erlebach, T.: *Network Analysis: Methodological Foundations*. Springer, Heidelberg (2005)

19. Fruchterman, T.M.J., Reingold, E.M.: Graph drawing by force-directed placement. *Software-Practice And Experience* 21, 1129–1164 (1991)
20. Fujibuchi, W., Goto, S., Migimatsu, H., Uchiyama, I., Ogiwara, A., Akiyama, Y., Kanehisa, M.: Dbget/linkdb: an integrated database retrieval system. *Pac. Symp. Biocomput.*, 683–694 (1998)
21. Shneiderman, B., Aris, A.: Network visualization by semantic substrates. *IEEE Trans Vis Comput Graph* 12(5), 733–740 (2006)
22. Abello, J., Abello, J., Korn, J.: Mgv: a system for visualizing massive multidigraphs. *Transactions on Visualization and Computer Graphics* 8(1), 21–38 (2002)
23. Durand, P., Labarre, L., Meil, A., Divol, J.L., Vandenbrouck, Y., Viari, A., Wojcik, J.: Genolink: a graph-based querying and browsing system for investigating the function of genes and proteins. *BMC Bioinformatics* 7(1), 21 (2006)
24. Crestani, F.: Application of spreading activation techniques in informationretrieval. *Artif. Intell. Rev.* 11(6), 453–482 (1997)
25. Rumelhart, D., McClelland, J.: *Parallel distributed processing: explorations in the microstructure of cognition, foundations*, vol. 1. MIT Press, Cambridge (1986)
26. van Ham, F., van Ham, F., van Wijk, J.: Interactive visualization of small world graphs. In: van Wijk, J. (ed.) *Proc. IEEE Symposium on Information Visualization INFOVIS 2004*, pp. 199–206 (2004)
27. Shen, Z., Ma, K.L., Eliassi-Rad, T.: Visual analysis of large heterogeneous social networks by semantic and structural abstraction. *IEEE Trans Vis Comput Graph* 12(6), 1427–1439 (2006)
28. Hernandez, T., Kambhampati, S.: Integration of biological sources: current systems and challenges ahead. *SIGMOD Rec* 33(3), 51–60 (2004)
29. Etzold, T., Argos, P.: Srs—an indexing and retrieval tool for flat file data libraries. *Comput Appl Biosci* 9(1), 49–57 (1993)
30. Schuler, G., Epstein, J., Ohkawa, H., Kans, J.: Entrez: molecular biology database and retrieval system. *Methods Enzymol* 266, 141–162 (1996)
31. Haas, L., Schwarz, P., Kodali, P., Kotlar, E., Rice, J., Swope, W.: Discoverylink: A system for integrated access to life sciences data sources. *IBM Systems Journal* 40(2), 489–511 (2001)
32. Wang, K., Tarczy-Hornoch, P., Shaker, R., Mork, P., Brinkley, J.F.: Biomediator data integration: beyond genomics to neuroscience data. In: *AMIA Annu Symp Proc*, pp. 779–783 (2005)
33. Chung, S.Y., Wong, L.: Kleisli: a new tool for data integration in biology. *Trends Biotechnol* 17(9), 351–355 (1999)
34. Stevens, R., Baker, P., Bechhofer, S., Ng, G., Jacoby, A., Paton, N.W., Goble, C.A., Brass, A.: Tambis: transparent access to multiple bioinformatics information sources. *Bioinformatics* 16(2), 184–185 (2000)
35. Davidson, S., Crabtree, J., Brunk, B., Schug, J., Tannen, V., Overton, G., Stoeckert, C.: K2/kleisli and gus: Experiments in integrated access to genomic data sources. *IBM Systems Journal* 40(2), 512–531 (2001)
36. Shah, S.P., Huang, Y., Xu, T., Yuen, M.M.S., Ling, J., Ouellette, B.F.F.: Atlas - a data warehouse for integrative bioinformatics. *BMC Bioinformatics* 6, 34 (2005)
37. Birkland, A., Yona, G.: Biozon: a system for unification, management and analysis of heterogeneous biological data. *BMC Bioinformatics* 7, 70 (2006)
38. Kuentzer, J., Backes, C., Blum, T., Gerasch, A., Kaufmann, M., Kohlbacher, O., Lenhof, H.P.: Bndb - the biochemical network database. *BMC Bioinformatics* 8, 367 (2007)

39. Baitaluk, M., Qian, X., Godbole, S., Raval, A., Ray, A., Gupta, A.: Pathsys: integrating molecular interaction graphs for systems biology. *BMC Bioinformatics* 7, 55 (2006)
40. Jansen, R., Yu, H., Greenbaum, D., Kluger, Y., Krogan, N.J., Chung, S., Emili, A., Snyder, M., Greenblatt, J.F., Gerstein, M.: A bayesian networks approach for predicting protein-protein interactions from genomic data. *Science* 302(5644), 449–453 (2003)
41. Figeys, D.: Combining different 'omics' technologies to map and validate protein-protein interactions in humans. *Briefings in Functional Genomics and Proteomics* 2, 357–365 (2004)
42. Koestler, A.: *The Act of Creation*. London Hutchinson (1964)
43. Hecht-Nielsen, R.: 3. In: *Confabulation Theory*, pp. 73–90. Springer, Heidelberg (2007)
44. Berthold, M.R., Cebron, N., Dill, F., Gabriel, T.R., Kötter, T., Meinel, T., Ohl, P., Sieb, C., Thiel, K., Wiswedel, B.: KNIME: The Konstanz Information Miner. In: *Studies in Classification, Data Analysis, and Knowledge Organization (GfKL 2007)*, Springer, Heidelberg (to appear)
45. U.S. National Library of Medicine: Pubmed (last accessed January 11, 2008), <http://www.ncbi.nlm.nih.gov/sites/entrez/>
46. Franke, L., van Bakel, H., Fokkens, L., de Jong, E.D., Egmont-Petersen, M., Wijmenga, C.: Reconstruction of a functional human gene network, with an application for prioritizing positional candidate genes. *Am J Hum Genet* 78, 1011–1025 (2006)
47. Gene Ontology Consortium: Creating the gene ontology resource: design and implementation. *Genome Res* 11(8), 1425–1433 (2001)
48. Cytoscape: Cytoscape (last accessed January 11, 2008), <http://www.cytoscape.org/>

Cost-Sensitive Classifier Evaluation Using Cost Curves

Robert C. Holte¹ and Chris Drummond²

¹ Computing Science Department, University of Alberta, Edmonton, Alberta, Canada, T6G 2E8

holte@cs.ualberta.ca

² Institute for Information Technology, National Research Council, Ontario, Canada, K1A 0R6

Chris.Drummond@nrc-cnrc.gc.ca

Abstract. The evaluation of classifier performance in a cost-sensitive setting is straightforward if the operating conditions (misclassification costs and class distributions) are fixed and known. When this is not the case, evaluation requires a method of visualizing classifier performance across the full range of possible operating conditions. This talk outlines the most important requirements for cost-sensitive classifier evaluation for machine learning and KDD researchers and practitioners, and introduces a recently developed technique for classifier performance visualization – the cost curve – that meets all these requirements.

1 Introduction

Methods for creating accurate classifiers from data are of central interest to the data mining community [2,15,16]. The focus of this talk is on binary classification, *i.e.* classification tasks in which there are only two possible classes, which we will call *positive* and *negative*. In binary classification, there are just two types of error a classifier can make: a *false positive* is a negative example that is incorrectly classified as positive, and a *false negative* is a positive example that is incorrectly classified as negative. In general, the cost of making one type of misclassification will be different—possibly very different—than the cost of making the other type.□

Methods for evaluating the performance of classifiers fall into two broad categories: numerical and graphical. Numerical evaluations produce a single number summarizing a classifier’s performance, whereas graphical methods depict performance in a plot that typically has just two or three dimensions so that it can be easily inspected by humans. Examples of numerical performance measures are accuracy, expected cost, precision, recall, and area under a performance curve (AUC). Examples of graphical performance evaluations are ROC curves [18,19], precision-recall curves [6], DET curves [17], regret graphs [13], loss difference plots [1], skill plots [4], prevalence-value-accuracy plots [21], and the method presented in this talk, cost curves [7,11].

Graphical methods are especially useful when there is uncertainty about the misclassification costs or the class distribution that will occur when the classifier is deployed. In this setting, graphical measures can present a classifier’s actual performance for a wide variety of different operating points (combinations of costs and class distributions),

¹ We assume the misclassification cost is the same for all instances of a given class; see [12] for a discussion of performance evaluation when the cost can be different for each instance.

whereas the best a numerical measure can do is to represent the average performance across a set of operating points.

Cost curves are perhaps the ideal graphical method in this setting because they directly show performance as a function of the misclassification costs and class distribution. In particular, the x-axis and y-axis of a cost curve plot are defined as follows.

The x-axis of a cost curve plot is defined by combining the two misclassification costs and the class distribution—represented by $p(+)$, the probability that a given instance is positive—into a single value, $PC(+)$, using the following formula:

$$PC(+) = \frac{p(+)\mathcal{C}(-|+)}{p(+)\mathcal{C}(-|+) + (1 - p(+))\mathcal{C}(+|-)} \tag{1}$$

where $\mathcal{C}(-|+)$ is the cost of a false negative and $\mathcal{C}(+|-)$ is the cost of a false positive. $PC(+)$ ranges from 0 to 1.

Classifier performance, the y-axis of a cost curve plot, is “normalized expected cost” (NEC), defined as follows:

$$NEC = FN * PC(+) + FP * (1 - PC(+)) \tag{2}$$

where FN is a classifier’s false negative rate, and FP is its false positive rate. NEC ranges between 0 and 1.

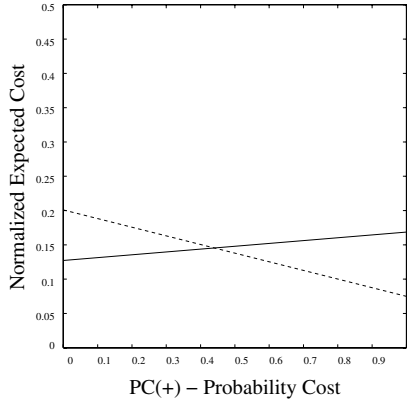


Fig. 1. Japanese credit - Cost curves for 1R (dashed line) and C4.5 (solid line)

To draw the cost curve for a classifier we draw two points, $y = FP$ at $x = 0$ and $y = FN$ at $x = 1$, and join them by a straight line. The cost curve represents the normalized expected cost of the classifier over the full range of possible class distributions and misclassification costs. For example, the dashed line in Figure 1 is the cost curve for the decision stump produced by 1R [14] for the Japanese credit dataset from the UCI repository and the solid line is the cost curve for the decision tree C4.5 [20] learns from the same training data. In this plot we can instantly see the relation between 1R and C4.5’s performance across the full range of deployment situations. The vertical

difference between the two lines is the difference between their normalized expected costs at a specific operating point. The intersection point of the two lines is the operating point where 1R's stump and C4.5's tree perform identically. This occurs at $PC(+) = 0.445$. For larger values of $PC(+)$ 1R's performance is better than C4.5's, for smaller values of $PC(+)$ the opposite is true.

Mathematically, cost curves are intimately related to ROC curves: they are “point-line duals” of one another. However, cost curves have the following advantages over ROC curves (see [11] for details):

- Cost curves directly show performance on their y-axis, whereas ROC curves do not explicitly depict performance. This means performance and performance differences can be easily seen in cost curves but not in ROC curves.
- When applied to a set of cost curves the natural way of averaging two-dimensional curves produces a cost curve that represents the average of the performances represented by the given curves. By contrast, there is no agreed upon way to average ROC curves, and none of the proposed averaging methods produces an ROC curve representing average performance.
- Cost curves allow confidence intervals to be estimated for a classifier's performance, and allow the statistical significance of performance differences to be assessed. The confidence interval and statistical significance testing methods for ROC curves do not relate directly to classifier performance.

For these reasons, we have gained insights into classifier performance using cost curves that would likely not have been possible using other methods [8,9,10] and other data mining researchers are using cost curves in their analyses [3,5,22,23].

Acknowledgments

We thank the Natural Sciences and Engineering Research Council of Canada and the Alberta Ingenuity Centre for Machine Learning for their support of this research.

References

1. Adams, N.M., Hand, D.J.: Comparing classifiers when misclassification costs are uncertain. *Pattern Recognition* 32, 1139–1147 (1999)
2. Antonie, M.-L., Zaiane, O.R., Holtex, R.C.: Learning to use a learned model: A two-stage approach to classification. In: *Proceedings of the 6th IEEE International Conference on Data Mining (ICDM 2006)*, pp. 33–42 (2006)
3. Bosin, A., Dessi, N., Pes, B.: Capturing heuristics and intelligent methods for improving micro-array data classification. In: *IDEAL 2007*. LNCS, vol. 4881, pp. 790–799. Springer, Heidelberg (2007)
4. Briggs, W.M., Zaretzki, R.: The skill plot: a graphical technique for the evaluating the predictive usefulness of continuous diagnostic tests. *Biometrics*, OnlineEarly Articles (2007)
5. Chawla, N.V., Hall, L.O., Joshi, A.: Wrapper-based computation and evaluation of sampling methods for imbalanced datasets. In: *Workshop on Utility-Based Data Mining held in conjunction with the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 179–188 (2005)

6. Davis, J., Goadrich, M.: The relationship between precision-recall and ROC curves. In: Proceedings of the 23rd International Conference on Machine Learning (ICML 2006), pp. 233–240 (2006)
7. Drummond, C., Holte, R.C.: Explicitly representing expected cost: An alternative to ROC representation. In: Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 198–207 (2000)
8. Drummond, C., Holte, R.C.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: Workshop on Learning from Imbalanced Datasets II, held in conjunction with ICML 2003 (2003)
9. Drummond, C., Holte, R.C.: Learning to live with false alarms. In: Workshop on Data Mining Methods for Anomaly Detection held in conjunction with the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 21–24 (2005)
10. Drummond, C., Holte, R.C.: Severe class imbalance: Why better algorithms aren't the answer. In: Gama, J., Camacho, R., Brazdil, P.B., Jorge, A.M., Torgo, L. (eds.) ECML 2005. LNCS (LNAI), vol. 3720, pp. 539–546. Springer, Heidelberg (2005)
11. Drummond, C., Holte, R.C.: Cost curves: An improved method for visualizing classifier performance. *Machine Learning* 65(1), 95–130 (2006)
12. Fawcett, T.: ROC graphs with instance-varying costs. *Pattern Recognition Letters* 27(8), 882–891 (2006)
13. Hilden, J., Glasziou, P.: Regret graphs, diagnostic uncertainty, and Youden's index. *Statistics in Medicine* 15, 969–986 (1996)
14. Holte, R.C.: Very simple classification rules perform well on most commonly used datasets. *Machine Learning* 11(1), 63–91 (1993)
15. Jumi, M., Suzuki, E., Ohshima, M., Zhong, N., Yokoi, H., Takabayashi, K.: Spiral discovery of a separate prediction model from chronic hepatitis data. In: Sakurai, A., Hasida, K., Nitta, K. (eds.) JSAI 2003. LNCS (LNAI), vol. 3609, pp. 464–473. Springer, Heidelberg (2007)
16. Liu, T., Ting, K.M.: Variable randomness in decision tree ensembles. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 81–90. Springer, Heidelberg (2006)
17. Liu, Y., Shriberg, E.: Comparing evaluation metrics for sentence boundary detection. In: Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2007), vol. 4, pp. IV–185–IV–188 (2007)
18. Provost, F., Fawcett, T.: Robust classification for imprecise environments. *Machine Learning* 42, 203–231 (2001)
19. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: Comparison under imprecise class and cost distributions. In: Proceedings of the Third International Conference on Knowledge Discovery and Data Mining, Menlo Park, CA, pp. 43–48 (1997)
20. Quinlan, J.R.: Induction of decision trees. *Machine Learning* 1, 81–106 (1986)
21. Remaley, A.T., Sampson, M.L., DeLeo, J.M., Remaley, N.A., Farsi, B.D., Zweig, M.H.: Prevalence-value-accuracy plots: A new method for comparing diagnostic tests based on misclassification costs. *Clinical Chemistry* 45, 934–941 (1999)
22. Ting, K.M.: Issues in classifier evaluation using optimal cost curves. In: Proceedings of The Nineteenth International Conference on Machine Learning, pp. 642–649 (2002)
23. Zhou, Z.-H., Liu, X.-L.: Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering* 18(1), 63–77 (2006)

Prospective Scientific Methodology in Knowledge Society

Genshiro Kitagawa

The Institute of Statistical Mathematics
4-6-7 Minami-Azabu, Minato-ku, Tokyo 106-8569 Japan
kitagawa@ism.ac.jp

Abstract. Due to rapid development of information and communication technologies, the methodology of scientific research and the society itself is changing. The present grand challenge is the development of the fourth methodology for scientific researches to create knowledge based on large scale massive data. To realize this, it is necessary to develop a method of integrating various types of information and of personalization, and the Bayes modeling is becoming the key technology. In the latter half of the paper, several time series examples are presented to show the importance of careful modeling that can take into account of essential information.

Keywords and Phrases: Information society, knowledge society, data centric science, active modeling, time series analysis.

1 Change of Scientific Research and Society Due to the Development of Information Technology

1.1 Change of Society Due to Informationization

Due to the progress of information and communication technologies (IT), large-scale massive data are accumulating in various fields of scientific researches and in society. As examples, we may consider the microarray data in life science, POS data in marketing, high-frequency data in finance, all-sky CCD image in astronomy, and various data obtained in environmental science, earth science, etc.

Rapid development of information and communication technologies influenced the research methodologies of science and technology and also society itself. In the information society, the information became as worthy as the substances and the energy, and the quantity of information decides the success and failure in the society. However, in the 21st century, *ubiquitous society* is approaching. In other words, ubiquitous society for every body, is going to be realized, where everybody can access to huge amount of information anywhere and anytime. If such post-IT society actually realized, the value of information itself will be depreciated, because huge amount of information can be shared by everybody. The success and failure in the post-IT society depends on whether one can extract

essential or useful information or knowledge from massive data. Therefore, in the post-IT society, the development of the methods and technologies for knowledge discovery and knowledge creation are very important.

Informationization also strongly influenced society. According to P. E. Drucker (1993), the capitalism has been moved to the post-capitalist society shortly after the World War II, due to the productivity revolution. This is because, the knowledge became the real, controlling resource and the absolutely decisive factor of production. According to him,

1.2 Expansion of Research Object and Change in Scientific Methodology Due to Informationization

The scientific research until the 19th century has developed basically under Newton-Descartes paradigm based on a mechanic view of the world. In the deductive approach, or in theoretical sciences, mathematics played an important role as the language of the science. However, the theory of evolution advocated by C. Darwin in mid 19th century means that every creature in real world evolves and changes with time.

Motivated by such changes of view of real world, in 1891, K. Pearson declared that everything in the real world can be an object of scientific research, and advocated *statistical science* (Tsubaki (2002)). It can be considered that the descriptive statistics and subsequent inferential statistics have developed as methodologies of achieving the grammar of science. By the establishment of the method of experimental sciences, not only biology but also many stochastic phenomena in real world such as economy and psychology, became the objects of scientific research.

In the latter half of the 20th century, the computation ability has increased rapidly by the development of the computers. As a result, numerical computation and Monte Carlo computation are applied to the nonlinear dynamics, complex systems, and intertwined high degree of freedom systems that have been difficult to handle by conventional analytic approach based on theoretical science, and the computational science has developed very rapidly.

However, development of IT became a trigger of another development of utilizing the information in rapidly exploding cyber-world. The development of the information technology resulted in accumulation of large-scale massive data in various fields of scientific researches and society, and a huge cyber-world is being created. It is not possible to talk about future development of the science and the technology without establishing methods of effective use of large-scale data. In this article, the scientific methodology supported by the technology of utilizing large-scale data set will be called the “fourth science” (Figure 1).

Needless to say, the first and the second methodologies are the theoretical sciences and the experimental sciences. These sciences are called the deductive method (or principle driven approach) and the inductive method (or data driven approach), and became mainsprings that promoted the scientific researches in the 20th century. However, in the latter half of the 20th century, the computing

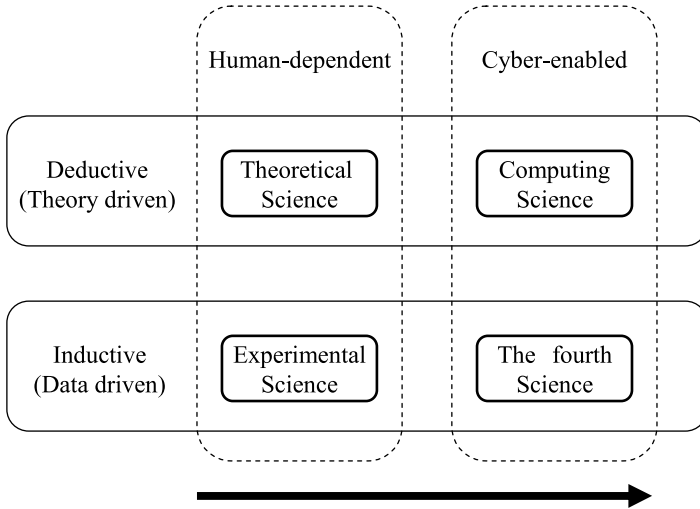


Fig. 1. Four methodologies that drive scientific researches

science was established as a method of alleviating the limit of the theoretical science based on analytic method, and succeeded in the prediction and simulation of nonlinear dynamics, complex systems or intertwined systems.

The computing science and “the fourth science” are newly establishing cyber-enabled deductive and inductive methods while the conventional methodologies, theoretical science and experimental science, relies on the researcher’s knowledge and experiences. Now having been developed the computing science, it is indispensable to promote this fourth science strategically to realize well-balanced scientific researches in the information era. It is notable that the U.S. National Science Foundation set “Cyber-enabled discovery and innovation” as a new priority area in the fiscal year 2008 (NSF(2008)).

In the field of global simulation etc., the data assimilation that integrates information obtained from the theoretical model and observations from satellite are becoming popular. In general, this can be considered as a technology to integrate the principle driven approach and the data driven approach. So far, in some area of scientific researches, the integration of two methodologies has been intentionally avoided. However, it is an important subject for the development of the knowledge society in the future. Actually, it can be considered as the filtering method from the standpoints of statistical science or control engineering. A rather natural way of thinking for researchers in methodologies can become the key technology for the science and technology in the future.

The statistics before the inferential statistics, such as the descriptive statistics was based on the observations of the object. On the other hand, the inferential statistics aims at performing scientific reasoning based on carefully designed rather small number of experimental data. However, due to the informationization in recent years, huge amount of heterogeneous data are accumulating, and knowledge discovery from massive large-scale data that are not necessarily

designed strictly, became important again. In spite of significant difference of amount of data, it may be said that it is a kind of atavism to descriptive statistics.

In relation to this, Dr. Hotta, President of the Research Organization of Information and Systems, stated an interesting thing about the transition of biology. According to him; “biology that used to be a kind of natural history, became an area of experimental sciences by adopting the scientific methodology in the 20th century. However, now it is becoming possible to decode entire genome of living bodies. In a sense, biology is returning to a kind of natural history in the modern age.”

1.3 Active Modeling

In parallel to the changes of the society and expansion of object of the scientific researches, our images of “knowledge” is also radically changing. In the past, a typical definition of the knowledge is “justified true belief,” that used to be applied to $\text{D}, \text{P}, \text{V}$. However, with the progress of modern age, the knowledge is becoming to applied to $\text{I}, \text{P}, \text{V}$ and brought productivity reevaluation and management revolution (Drucker (1993)). Now, approaching to the knowledge society, the knowledge discovery and the knowledge creation are becoming important. Corresponding to these changes, the definition of the knowledge is also changing to “information effective in action and information focused on results.”

In the area of statistical science, the role of the model is changing along with the change of the scientific methodology and the image of the knowledge. In the conventional setting of the mathematical statistics, by assuming that the data is obtained from the true distribution, we aimed at performing an objective inference concerning the true structure. However, in the statistical modeling for information processing or for information extraction, it is rather natural to consider that the model is not true or close replica of the truth but is an useful “tool” for information extraction.

Once the statistical model is considered like this, a flexible standpoint of model construction is obtained, namely, in statistical modeling we should use not only the present data but also the theory on that subject, empirical knowledge, and any other data that have been obtained so far, and even the objective of the modeling. Once the model is obtained, the information extraction, knowledge discovery, prediction, simulation, control, and management, etc. can be achieved straightforwardly or deductively (Figure 2). Needless to say, the result of knowledge acquisition using the model leads to refined modeling. In this article, such a process will be called active modeling. Therefore, active modeling forms a spiral of the knowledge development.

To establish the “fourth science” for large-scale massive data, we have the following grand challenges:

1. Prediction and knowledge discovery based on large-scale data,
2. Quantitative risk science, i.e., modeling uncertainty and managing risks,
3. Real world simulation,
4. Service science, i.e., innovations in medical care, pharmacology, marketing, education, etc.

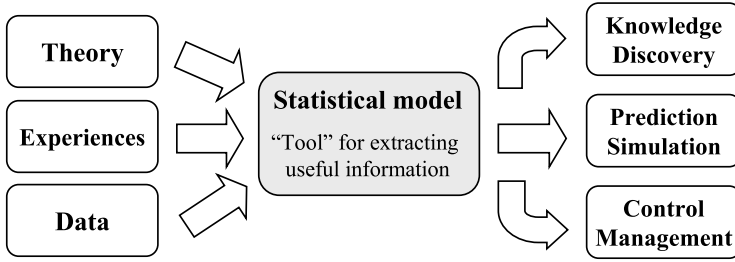


Fig. 2. Active modeling and the use of identified model

In addition, as element technologies for these problems solving, the technologies for the knowledge integration and for the personalization are needed. For the personalization, it is necessary to convert from the formulation of the past statistical inference to an inference about individual object. Of course, this does not mean to abandon the main feature of the statistical inference, namely, the standpoint of capturing stochastic phenomena based on distribution, and can be realized by an appropriate conditioning on the distribution. However, in the modeling for personalization, ultimate conditioning is required, and the difficult problem called “new NP problem” arise, in which the number of variables is much more than the number of observations.

Anyways, the technology that becomes a key to achieve information integration and an ultimate conditioning is the Bayes modeling. It is because various prior information and information from data can be integrated by the use of the Bayes model. Although the Bayes’ theorem was discovered middle in the 18th century, and the superiority of the inference based on the Bayes’ theorem was well-known, application to real problems was rather rare, due to philosophical controversy, difficulty in determining the prior distributions, and the difficulty in computing the posterior distribution, etc. However, owing to the development of statistical science such as the change in the viewpoint of modeling, the model evaluation criterion that objectively evaluates models that are introduced subjectively, and the development of statistical computing methods such as MCMC and sequential Monte Carlo methods (Kitagawa and Gersch (1996), Doucet et al. (2001)), now the Bayes method becomes the main tool in information extraction, information integration, and information retrieval, etc. (Higuchi et al. (2007)).

Although the Bayes modeling is becoming of practical use, there is one difficulty in the achievement of modeling. Namely, there is no established methodology to derive appropriate class of models for particular problem. Therefore, the researcher’s art is still demanded in the most important part of statistical modeling, i.e., the presentation of the model family. The raison d’être of the researchers, in particular of the statisticians in a cyber world can be found here.

2 Active Modeling of Time Series: Some Examples

So far, we have discussed importance of integrating various kind of information considering the characteristics of the object and objective of the modeling. In this section, we shall show several examples of time series modeling. In time series analysis, the nonlinear non-Gaussian state-space model

$$x_n = f(x_{n-1}, v_n), \quad y_n = h(x_n, w_n) \quad (1)$$

is a useful tool for information extraction and information integration (Kitagawa and Gersch (1996)). Here, y_n , x_n , v_n and w_n are time series, unknown state vector, system noise and observation noise, respectively. The functions $f(x, v)$ and $h(x, w)$ are, in general, nonlinear functions and the distributions of v_n and w_n are not necessarily Gaussian. This general state-space model is a powerful platform for integrating various types of information in time series analysis.

The ordinary state-space model used to be popular in time series modeling because of the presence of the computationally efficient Kalman filter. However, development of sequential Monte Carlo methods for filtering and smoothing with general state-space model opened the door to flexible nonlinear non-Gaussian modeling of time series (Kitagawa and Gersch (1996), Doucet et al. (2001)).

2.1 Prediction and Interpolation by Time Series Modeling

Figure 3 shows the results of increasing horizon prediction of BLSALLFOOD data, the number of food industry workers in US (Kitagawa and Gersch (1996)), based on autoregressive (AR) models with various orders. This time series has apparent seasonal component. In the following prediction, the AR models are estimated based on the first 110 observations and predict the succeeding 46 observations, $y_n, n = 111, \dots, 156$.

The upper left plot shows the case when AR model with order 1, hereafter denoted as AR(1), was fitted by the Yule-Walker method and obtain the increasing horizon predictive distributions by the Kalman filter. The smooth curve shows the predicted values, i.e. the means of the increasing horizon predictive distributions, and two dotted curves above and below this mean function are the ± 1 standard error interval. Almost all actual observations are in these bounds that suggests slight over estimation of the prediction variance. Except for this problem, the prediction looks reasonable. However, the seasonal pattern is not considered at all and this cannot be a good prediction.

The upper right plot shows the results by the AR(3). In this case, the first one cycle was reasonably predicted. But the prediction over one year period is too smooth and is almost the same as the AR(1). The lower left plot shows the results by AR(11). In this case, cyclic behavior was predicted in entire period of four years and the prediction errors are significantly reduced. On the other hand, the lower right plot shows the results by AR(15), the minimum AIC model. Comparing with the prediction by AR(11), it can be seen that much more precise prediction was attained by this model. Actually, it is remarkable that details of the seasonal pattern were successfully predicted by this model.

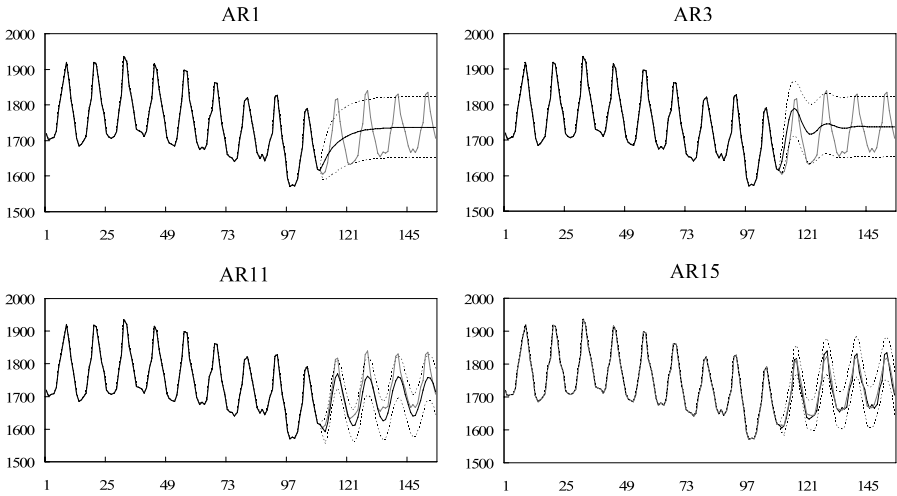


Fig. 3. Increasing horizon prediction of BLSALLFOOD data by AR models with various orders (Kitagawa (2005))

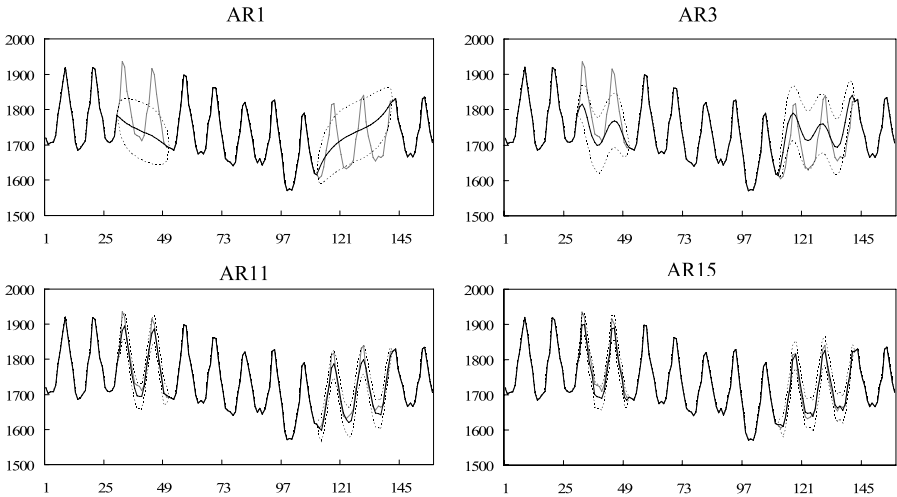


Fig. 4. Interpolation of missing observations by AR models with various orders (Kitagawa and Gersch (1996))

From these results, it can be concluded that even though we obtain the best predictors by the Kalman filter, if the model order is inappropriate, we cannot get good predictive distribution.

Figure 4 show the results of interpolating missing observations by AR models. In this example, 50 observations, y_{41}, \dots, y_{60} and y_{111}, \dots, y_{140} , are assumed to be missing and are estimated by the fixed interval smoothing algorithm

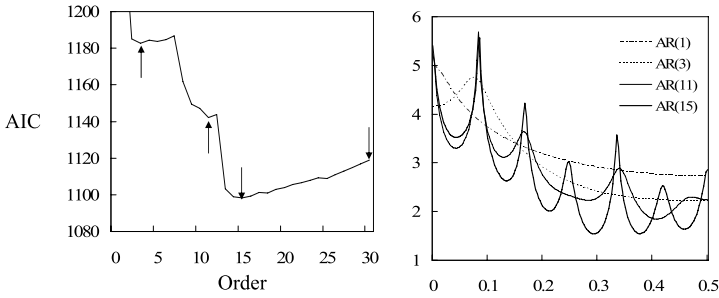


Fig. 5. AIC's and power spectra of the fitted AR models

(Kitagawa and Gersch (1996)). The upper left plot shows the result by AR(1). Approximately 75% of missing observations are included in the ± 1 confidence interval. However, similar to the case of increasing horizon prediction, the seasonal pattern of the time series is not used at all for prediction.

The upper right plot shows the result by AR(3). In this case, moderate cyclic pattern was obtained. But the width of the confidence interval is not reduced significantly. In the case of AR(11) shown in bottom left plot, reasonable estimates are obtained by incorporating the annual cyclic pattern. The AR(15) also yields the similar results.

Here, we shall consider from the point of view of the power spectra. The right plot of Figure 5 shows the power spectra obtained by four AR models use for increasing horizon prediction and interpolation. The spectrum by AR(1) is very smooth curve that falls in the right. It can capture the characteristic that the time series has stronger long period components but cannot capture any cyclic behavior. The spectrum by AR(3) has a peak around at $f = 0.08$, corresponding to one year cycle. However, its peak is very broad indicating that it does not capture very definite cyclic pattern. In the case of AR(11), sharp peaks with one year, 6 months and 3 months cycles appeared, but the ones with 4 months and 2.4 months period did not. This means that by AR(11), it is possible to express one year cycle but cannot reproduce every details within the one cycle. On the other hand, in the case of AR(15), any cyclic pattern with one year cycle can be expressed since the spectrum by AR(15) can express 6 periodic components and one direct current component, i.e., $f = 0$.

Incidentally, if we use too higher order models, the spectrum may have more than 6 peaks and it may deteriorate the accuracy of increasing horizon prediction and interpolation. The left plot of Figure 5 shows the values of AIC for various orders of AR models (Konishi and Kitagawa (2007)). The minimum of the AIC was attained at order 15. AR(3) and AR(11) are local minima of AIC but corresponding AIC are significantly larger than that of AR(15). AICs of the models with order higher than 15 are larger than that of AR(15), suggesting poor prediction abilities than the AR(15).

These results suggest an obvious thing that given a specific model, we cannot breakthrough the limitation of that model. In other words, even if we use the

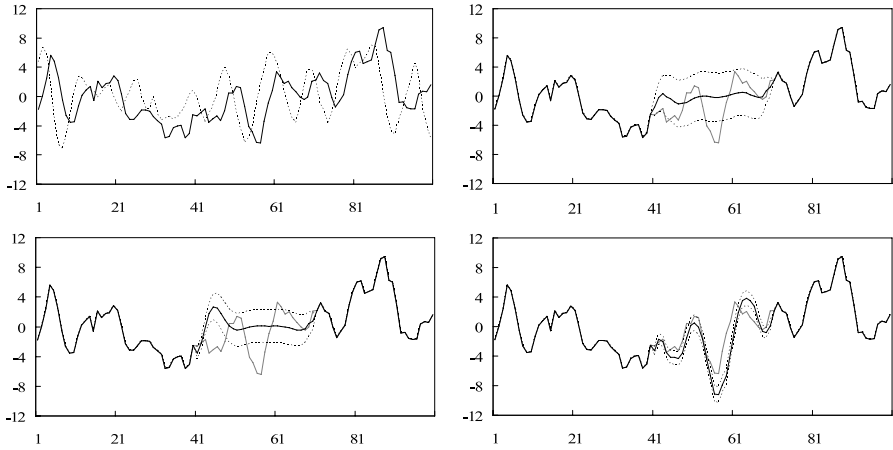


Fig. 6. Interpolation by multivariate AR models

best prediction or best interpolation, it does not guarantee the optimality of the estimation.

2.2 Use of Multivariate Structure

In this subsection, we shall consider interpolation of 2-variate time series (y_n, x_n) and exemplify that the interpolation may significantly improved by incorporating information from other time series. Figure 6 shows an artificially generate 2-variate time series. The problem here is to estimate the data y_{41}, \dots, y_{70} by assuming that they are missing.

Figure 6 show the result by obtained by using univariate AR model. The best model selected by AIC was AR(5). Since the periodicity is not so strong as the time series of considered in the previous subsection, the interpolated values are very smooth and good reproduction of the missing observations are not achieved even with the AIC best model. This result shows a possible limitation of the univariate time series model for recovering missing data.

To mitigate this limitation, we shall consider the use of information from other time series. Two plots in Figure 7 show the scatter plots of two time series. The left plot show the scatter plot between y_n and x_n , and the right one between y_n and x_{n-2} . Almost no correlation between two time series is seen between y_n and x_n . On the other hand, in the right plot, significant correlation exists between y_n and previous values of other time series, x_{n-2} is seen. These suggest the possibility of improving the prediction or interpolation by using the information about the past values of x_n .

The bottom left plot of Figure 6 show the result obtained by interpolating the missing observations by using the bivariate AR model. It is assumed that on the missing interval both of y_n and x_n are not observed. Although the confidence interval is slightly reduced, the estimated values are similar to those of univariate AR model. On the other hand, the bottom right plot shows the case when we

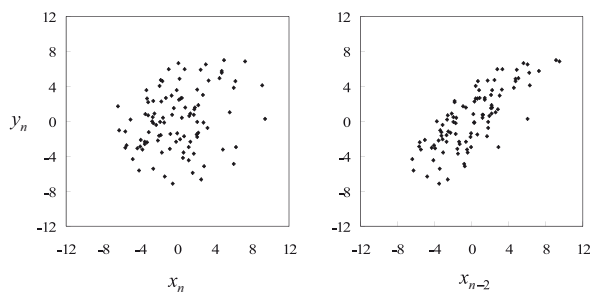


Fig. 7. Scatter plots of two time series

can use the observations of the time series x_n on this interval. Even though we used the same bivariate AR model, very good reproduction of the missing observations of y_n is achieved by using the information of x_n . Although, it is rather obvious, this example clearly shows that we should utilize the all available information appropriate for the current purpose.

References

1. Doucet, A., Freitas, F., Gordon, N.: Sequential Monte Carlo Methods in Practice. Springer, New York (2001)
2. Drucker, P.E.: Post-Capitalist Society. Harper Business, New York (1993)
3. Kitagawa, G., Gersch, W.: Smoothness Priors Analysis of Time Series. Springer, New York (1996)
4. Konishi, S., Kitagawa, G.: Information Criteria and Statistical Modeling. Springer, New York (2007)
5. Matsumoto, N., Kitagawa, G., Roeloffs, E.A.: Hydrological response to earthquake in Haibara well, central Japan–1. Geophysical Journal International 155(3), 885–898 (2003)
6. NSF, NSF wide investment (2008), http://www.nsf.gov/news/priority_areas/index.jsp
7. Tsubaki, H.: Statistical science aspects of business (in Japanese). In: Proceeding of the Japan Society of Applied Science, vol. 16, pp. 26–30 (2002)

SubClass: Classification of Multidimensional Noisy Data Using Subspace Clusters

Ira Assent¹, Ralph Krieger¹, Petra Welter¹, Jörg Herbers², and Thomas Seidl¹

¹ Data Management and Exploration Group

RWTH Aachen University

Aachen, Germany

{assent, krieger, welter, seidl}@cs.rwth-aachen.de

² INFORM GmbH

Pascalstraße 23

Aachen, Germany

joerg.herbers@inform-ac.com

Abstract. Classification has been widely studied and successfully employed in various application domains. In multidimensional noisy settings, however, classification accuracy may be unsatisfactory. Locally irrelevant attributes often occlude class-relevant information. A global reduction to relevant attributes is often infeasible, as relevance of attributes is not necessarily a globally uniform property. In a current project with an airport scheduling software company, locally varying attributes in the data indicate whether flights will be on time, delayed or ahead of schedule. To detect locally relevant information, we propose combining classification with subspace clustering (*SubClass*). Subspace clustering aims at detecting clusters in arbitrary subspaces of the attributes. It has proved to work well in multidimensional and noisy domains. However, it does not utilize class label information and thus does not necessarily provide appropriate groupings for classification. We propose incorporating class label information into subspace search. As a result we obtain locally relevant attribute combinations for classification. We present the SubClass classifier that successfully exploits classifying subspace cluster information. Experiments on both synthetic and real world datasets demonstrate that classification accuracy is clearly improved for noisy multidimensional settings.

1 Introduction

Data produced in application domains like life sciences, meteorology, telecommunication, and multimedia entertainment is rapidly growing, increasing the demand for data mining techniques which help users generate knowledge from data. Many applications require incoming data to be classified according to models derived from labeled historic data. In a current project, we investigate flight delays for airport scheduling purposes. The significance of flight delays can e.g. be studied in reports of the Bureau of Transportation Statistics in the U.S. [7] and the Central Office for Delay Analysis of Eurocontrol [11]. Extensive flight data is recorded by flight information systems at all major airports. Using such databases, we classify flights as on time, delayed or ahead

of schedule. This classification is essential in refining robust scheduling methods for airport resources and ground staff (like the one presented in [6]).

For classification, numerous techniques exist. For our noisy database that contains nominal attributes, numerical classifiers are not applicable. Neural networks or support vector machines do not allow users to easily understand the decision model for flight classification [20][17]. Bayes classifiers, decision trees, and nearest neighbor classifiers provide explanatory information, yet assume globally uniform relevance of attributes [20][18][3]. It has been shown that each type of classifier has its merit; there is no inherent superiority of any classifier [10]. However, classification is difficult in the presence of noise. Moreover, patterns may not show across all data attributes for all classes to be learned. In multidimensional data only a subgroup of attributes may be relevant for classification. This relevance is not globally uniform, but differs from class to class and from instance to instance.

We have validated the assumption of local relevance of attributes for the flight classification project by training several types of classifiers. When using only attributes which are determined as relevant by standard statistical tests, classification accuracy actually drops. This suggests that globally irrelevant attributes are nonetheless locally relevant for individual patterns. We therefore target at grouping flights with similar characteristics and identifying structure on the attribute level. In the flight domain, several aspects support the locality of flight delay structures. As an example, passenger figures may only influence departure delays when the aircraft is parked at a remote stand, i.e. when bus transportation is required. At some times of the day, these effects may be superposed by other factors like runway congestion. Weather conditions and other influences not recorded in the data cause significant noise.

Recent classification approaches like [9] use local weighting in nearest neighbor classification to overcome this drawback. Combining relevant attributes hierarchically a subspace is constructed for nearest neighbor classification. However, locally adaptive nearest neighbor methods do not consider the correlation of different attribute sets. Association rules have been extended to classification [16]. Recent approaches adopt subspace clustering methods to identify relevant subspaces for rule based classification [21].

In this work, we propose a nearest neighbor classifier which directly uses the result of our new subspace clustering method. Note that our approach is different from semi-supervised learning where unlabeled data is used for training [22]. Our approach assumes class labels that are directly incorporated into subspace clustering. Clustering is helpful for understanding the overall structure of a data set. Its aim is automatic grouping of the data in absence of any known class labels in historic data [13]. Since class labels are not known in advance (“unsupervised learning”), they are not used to classify according to given groupings (“supervised learning”). Hence clustering is not appropriate for classification purposes by its very nature [13]. However, the structures detected by clustering may be helpful for detecting local relevance of attributes. For noisy and high-dimensional data, clustering is often infeasible as clusters are hidden by irrelevant attributes. Different attribute combinations might show different clustering structures, thus the aim of subspace clustering is to detect clusters in arbitrary projections (“subspaces”) of the attributes. As the number of subspaces is exponential in the number of

attributes, most approaches try to prune the subspace search space [18,4]. Subspace clustering has been shown to successfully detect locally relevant attribute combinations [5,5].

We propose combining both worlds, supervised learning and unsupervised learning by incorporating class label information into subspace search and clustering. Classification based on these classifying subspace clusters exploits both class and local correlation information. The flight classification problem is used to evaluate our model. Its applicability, however, goes beyond this scenario. In fact, there are many more application areas where classification has to handle noisy multidimensional data with locally relevant attributes.

This paper is structured as follows: we define interesting subspaces for subspace classification in Section 2.1. Classifying subspace clusters and the overall classification scheme are discussed in Sections 2.2 and 2.3 respectively. Algorithmic concepts are presented in Section 3. The proposed method is evaluated in the experiments in Section 4 on both synthetic and flight data, before concluding the paper.

2 Subspace Classification

Subspace clustering is a recent research area which tries to detect local structures in the presence of noise or high-dimensional data where meaningful clusters can no longer be detected in all attributes [18,15]. As searching all possible subspaces is usually intractable, subspace clustering algorithms try to focus on promising subspace regions. The challenge is a suitable notion of interestingness for subspaces to find all relevant clusters. Subspace clustering is a technique well-suited to identify relevant regions of historic data, however, it is not suited for classification "as is". Our classification approach is capable of exploiting local patterns in the data for classification. This requires detecting subspaces and subspace clusters that are also based on class structure. Our *SubClass* model thus comprises three steps:

- Step 1: **interesting subspaces** for classifying clusters: Section 2.1
- Step 2: **classifying subspace clusters**: Section 2.2
- Step 3: a **classification** scheme: Section 2.3

2.1 Step 1: Interesting Subspaces

Interesting subspaces for classifying clusters exhibit a clustering structure in their attributes as well as coherent class label information. Such a structure is reflected by homogeneity in the attribute values or class labels of that subspace. Homogeneity can be measured using Shannon Entropy [19], or entropy for short. From an information theoretic perspective, Shannon entropy is the minimum number of bits required for encoding information. More frequently occurring events are encoded with fewer bits than less frequent ones. The sum over logarithmic probabilities weighted by their probability, measures the amount of information, i.e. the heterogeneity of the data.

Definition 1. Shannon Entropy. Given a random variable X and its possible events v_1, \dots, v_m the Shannon Entropy $H(X)$ is defined as:

$$H(X) = - \sum_{i=1}^m p(v_i) \cdot \log_2 p(v_i)$$

Transferring the entropy notion to the clustering or classification domain, an attribute can be seen as a random variable whose domain is the set of all possible events. In case of continuous domains, the entropy requires discretization of attributes. Entropy according to a set of attributes with respect to a set of class labels is then:

Definition 2. Attribute Entropy. Given a set of attributes X_1, \dots, X_m , their possible values v_1, \dots, v_m , and class labels $C = \{c_1, \dots, c_n\}$, attribute entropy is defined as:

$$H(X_1, \dots, X_m | C) = - \sum_{c_i \in C} \sum_{v_1 \in X_1} \dots \sum_{v_m \in X_m} p(c_i) \cdot H(X_1, \dots, X_m | C = c_i)$$

Attribute entropy is thus the sum over all conditional attribute entropy value combinations weighted by the class label probabilities. It is a measure for the clustering tendency for all class labels c_i of a subspace in terms of the attributes. To measure the clustering tendency in terms of individual class labels, we define class entropy according to conditional entropy $H(C|X)$ (as e.g. in [18]).

Definition 3. Class Entropy. Given a set of attributes X_1, \dots, X_m , their possible values v_1, \dots, v_m , and class label C the conditional entropy of a segmentation along these attribute values is defined as:

$$H(C | X_1, \dots, X_m) = - \sum_{v_1 \in X_1} \dots \sum_{v_m \in X_m} p(v_1, \dots, v_m) \cdot H(C | X_1 = v_1, \dots, X_m = v_m)$$

Class entropy is thus the sum over all conditional class entropy value combinations for individual class labels C . It corresponds to investigating the data for individual classes instead of aggregated as for attribute entropy.

We are interested in subspaces that exhibit both a distinct class structure as well as a clear clustering structure. Since entropy measures homogeneity, we are interested in low entropy values that reflect a non-uniform distribution of class or attribute values.

However, comparing subspaces using entropy is clearly biased with respect to the number of attributes. Subspaces with more attributes typically have lower entropy values. This is due to the fact that with increasing attribute number, objects tend to be less similar: each attribute contributes potential dissimilarity [5]. Thus, we have to normalize entropy with respect to the number of attributes. Normalization to a range of [0,1] can be achieved by taking the maximum possible entropy value for a given number of attributes into account. Maximum entropy means all values are equally likely, i.e. a uniform distribution. $H_{uniform}(X_1, \dots, X_m | C)$ for $d = |X_1 \times \dots \times X_m|$ possible attribute combinations is determined as: $H_{uniform}(X_1, \dots, X_m | C) = -d \cdot \frac{1}{d} \cdot \log_2 \frac{1}{d} = -\log_2 \frac{1}{d} = \log_2 d$, since in uniform distribution, each attribute value occurs $1/d$ times. For larger numbers of attributes, the theoretical upper bound of $\log_2 d$ cannot be reached, as the actual number of instances is smaller than the number of possible attribute value combinations d .

To account for this, we the number of instances $|I|$ is used in this case:

$$H_N(X_1, \dots, X_m|C) = \frac{H(X_1, \dots, X_m|C)}{\min\{\log_2|I|, \log_2 d\}}$$

In a similar spirit, we use the overall class distribution to normalize class entropy:

$$H_N(C|X_1, \dots, X_m) = \frac{H(C|X_1, \dots, X_m)}{H(C)}$$

Since those subspaces are interesting that cover both aspects, we define interestingness as a convex combination of attribute and class entropy, provided that each of the two is within reasonable bounds:

Definition 4. Subspace Interestingness. Given attributes X_1, \dots, X_m , a class attribute C , and a weighting factor $0 \leq w \leq 1$, a subspace is interesting with respect to thresholds β, λ iff:

$$\begin{aligned} w \cdot H_N(X_1, \dots, X_m|C) + (1 - w) \cdot H_N(C|X_1, \dots, X_m) &\leq \beta \\ \wedge H_N(X_1, \dots, X_m|C) &\leq \lambda \wedge H_N(C|X_1, \dots, X_m) &\leq \lambda \end{aligned}$$

Thus, a subspace is interesting for subspace classification if it shows low normalized class and attribute entropy as an indication of class and cluster structure. w allows assigning different weights to these two aspects for different applications, while λ is set to fairly relaxed threshold values to ensure that both aspects fulfill minimum entropy requirements.

2.2 Step 2: Classifying Subspace Clusters

Having defined interesting subspaces, the next step is detecting *classifying subspace clusters*. On discretized data, clusters can be defined as frequent attribute value combinations. To incorporate class information, these groupings should be homogeneous with respect to class label. We defined the absolute frequency

$$AbsFreq(v_1, \dots, v_m) = |\{o, o|_S = (v_1, \dots, v_m)\}|$$

as the number of objects o which exhibit the attribute values (v_1, \dots, v_m) in subspace S (projection $o|_S$ contains those attribute values v_i from o where $X_i \in S$).

To ensure that non-trivial clusters are mined, we normalize frequency with respect to the expected frequency of uniformly distributed subspaces. The expected frequency

$$ExpFreq(v_1, \dots, v_m) = AbsFreq(v_1, \dots, v_m) * d/|I|$$

is the number of cluster objects in comparison to the number of instances $|I|$ per attribute combination under uniform distribution. Classifying subspace clusters exceed minimum frequency for both absolute and relative (expected) frequency. Note that minimum absolute frequency simply ensures that a cluster exceeds a minimum size even for very small expected frequency values:

Definition 5. Classifying Subspace Cluster. Given a subspace S of attributes X_1, \dots, X_m , a classifying subspace cluster SC with respect to attribute values v_1, \dots, v_m , minimum frequency thresholds ϕ_1, ϕ_2 , and maximum entropy γ is defined as follows:

- $H_N(C|X_1 = v_1, \dots, X_m = v_m) \leq \gamma$
- $AbsFreq(v_1, \dots, v_m) \geq \phi_1$
- $ExpFreq(v_1, \dots, v_m) \geq \phi_2$

Classifying subspace clusters have low normalized class entropy, as well as high frequency in terms of attribute values. Thus, they are homogeneous in terms of class and show local attribute correlations.

2.3 Step 3: Classification

Classification of a given object o is based on the class label distribution of similar classifying subspace clusters. For nominal values as they occur in our flight data, an object o is typically contained in several subspace clusters and similarity is reduced to containment. Let $CSC(o) = \{SC_i | v_k = o_k \forall v_k \in SC_i\}$ denote the set of all classifying subspace clusters containing object o . Simply assigning the majority class label from this set $CSC(o)$ would be biased with respect to very large and redundant subspace clusters, where redundancy means similar clusters in slightly varying projections [5]. We therefore propose an iterative procedure that takes the *information gain* into account to build the decision set $DS_k(o)$.

Just as in the subspace clustering step we measure class homogeneity using the conditional class entropy. Starting with an empty decision set and apriori knowledge about class distribution $H(C)$ we select up to k subspace clusters with maximal information gain on the class label as long as more than ϕ_1 objects are contained in the decision space, i.e. the projection to the union of dimensions of the subspace clusters in the decision set.

Definition 6. Classification. Given a dataset D , parameter k , an object $o = (o_1, \dots, o_d)$ is classified to the majority class label of decision set DS_k . DS_k is iteratively constructed from $DS_0 = \emptyset$ by selecting the subspace cluster $SC_j \in CSC(o)$ which maximizes the information gain about the class label:

$$DS_j = DS_{j-1} \cup SC_j, SC_j = \left\{ \underset{SC_i \in CSC(o)}{\text{argmax}} \{H(C|DS_{j-1}) - H(C|DS_{j-1} \cup SC_i)\} \right\}$$

under the constraints that the decision space contains at least ϕ_1 objects:

$$|\{v \in D, v|_{DS_k} = o|_{DS_k}\}| \geq \phi_1$$

and that the information gain is positive

$$H(C|DS_{j-1}) - H(C|DS_{j-1} \cup SC_i) > 0$$

Hence, the decision set of an object o is created by choosing those k subspace clusters containing o that provide most information on the class label, as long as more than a

minimum number of objects are in the decision space. o is then classified according to the majority in the decision set DS_k . The decision set is then the set of locally relevant attributes that were used to classify object o . The attributes in the decision set are helpful for users wishing to understand the information that led to classification.

3 Algorithmic Concept

Our algorithmic concept focuses on step 1 that is the computationally most complex. A simple brute-force search would require evaluating all 2^N subspaces which is not acceptable for high dimensionality N . We thus propose lossless pruning of subspaces based on two entropy monotonicities.

Theorem 1. Upward Monotony of the Class Entropy. *Given a set of m attributes, subspace $S = \{X_1, \dots, X_m\}$, $e \in \mathbb{R}^+$ and $T \subseteq S$, the class entropy in subspace T is less than or at most equal to the class entropy of its superspace S :*

$$H(C|T) < e \quad \Rightarrow \quad H(C|S) < e$$

Proof. The theorem follows immediately from $H(X|X_i, X_j) \leq H(X|X_i)$ [12].

This theorem states that the class entropy decreases monotonically with growing number of attributes. Conversely, attribute entropy increases monotonically with the number of attributes.

Theorem 2. Downward Monotony of the Attribute Entropy. *Given a set of m attributes, subspace $S = \{X_1, \dots, X_m\}$, $e \in \mathbb{R}^+$ and $T \subseteq S$, the attribute entropy in subspace T is greater than or at most equal to the class entropy of its superspace S :*

$$H(S|C) < e \quad \Rightarrow \quad H(T|C) < e$$

Proof. The theorem follows immediately from $H(X_i, X_j|C) \geq H(X_i|C)$ [12].

We exploit monotonicity by pruning

- all those subspaces T whose superspaces $S \supset T$ fail the class entropy threshold. This is correct since the normalization factor $H(C)$ is independent of the subspace.
- Prune all those superspaces T whose subspaces $S \subset T$ fail the attribute entropy threshold if $\log_2|I| \geq \log_2|S|$. This is correct since the normalization factor is independent of the subspace if $\min\{\log_2|I|, \log_2|S|\} = \log_2|I|$.

Our proposed algorithm alternately determines lower dimensional and higher dimensional *one-sided homogeneous* subspaces, i.e. subspaces that are homogeneous w.r.t. to class or attribute entropy, respectively. In each step new candidates are created from the set of one-sided homogeneous subspaces mined in the last step.

Figure 1 illustrates pruning in a subspace lattice of four attributes. The solid line is the boundary for pruning according to attribute entropy and the dashed line according to class entropy. Each subspace below the attribute boundary and above the class boundary is homogeneous with respect to the entropy considered. The subspaces between

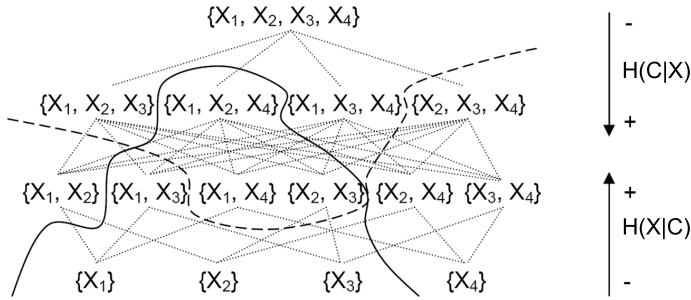


Fig. 1. Lattice of Subspaces and their projections used for up- and downward pruning

both boundaries are interesting subspace candidates, whose combined entropy has to be computed in the next step.

For the bottom up case, the apriori property, originally from association rule mining, can be used to create new candidates [2][8][15]. Following the apriori approach, we join two attribute homogeneous subspaces of size m with identical prefixes (e.g. in lexicographic ordering) to create a candidate subspace of size $m + 1$. After this, each new candidate is checked for entropy validity, i.e. if all of its possible subspace of cardinality m are contained in the set of attribute homogeneous candidate subspaces.

We suggest a similar method for top down candidate generation using class monotonicity. From the set of class homogeneous subspaces of dimensionality m , we generate all subspace candidates of dimensionality $m - 1$. We develop a method that ensures that each subspace candidate is only generated once. Based on the lexicographic order, our method uniquely generates a subspace of dimensionality $m - 1$ from its smallest supspace. Note that this guarantees that all candidates but no superfluous candidates are generated (see example below). After this, just as with apriori, we check whether all superspaces containing the newly generated candidates are class homogeneous subspaces. Otherwise the new generated subspace is removed from the candidate set.

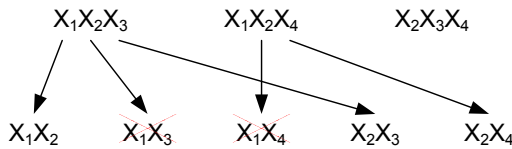


Fig. 2. Example top down generation

Example. Assume four attributes X_1, \dots, X_4 from the previous step subspaces $X_1X_2X_3$, $X_1X_2X_4$, and $X_2X_3X_4$ that satisfy the class entropy criterion. In order to generate candidates, we iterate over these subspaces in lexicographic order. The first three-dimensional subspace $X_1X_2X_3$ generates the two-dimensional subspaces X_1X_2 (drop X_3), X_1X_3 (drop X_2), X_2X_3 (drop X_1). Next, $X_1X_2X_4$ generates X_1X_4 and

X_2X_4 . X_1X_2 is not generated, because dropping X_4 is not possible, as it is preceded by X_3 which is not contained in this subspace. The last three-dimensional subspace $X_2X_3X_4$ does not generate any two-dimensional subspace since the leading X_1 is not contained; its subsets X_2X_3 and X_2X_4 have been generated by other three-dimensional subspaces. After candidate generation, we check their respective supersets. For example, for X_1X_2 , its supersets $X_1X_2X_3$ and $X_1X_2X_4$ exist. For X_1X_3 , its superset $X_1X_2X_3$ exists, but $X_1X_3X_4$ does not, so it is removed from further consideration following monotony pruning. Likewise, X_1X_4 is removed as $X_1X_3X_4$ is missing, but X_2X_3 and X_2X_4 are kept.

As we use two entropies, one with downward, one with upward pruning, subspaces may need to be considered twice. Minimizing computations is thus a trade-off. Figure 3 illustrates these effects. A missing candidate in S_{Down} (e.g. X_1X_2) means that this candidate has an attribute entropy above β . According to the attribute monotony, superspaces (e.g. $X_1X_2X_3$) have an attribute entropy above β and thus the combined entropy is also greater than β . Even though the subspace could be pruned according to combined entropy, it is still required for valid class entropy candidate generation. There is thus a trade off between avoiding computations and reducing the search space

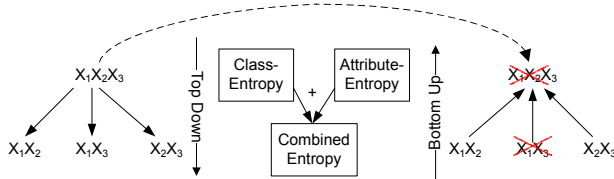


Fig. 3. Pruning of subspace $X_1X_2X_3$

by pruning high entropy subspaces. A good heuristic is to evaluate the entropy of those subspaces for which larger subspaces already had a high entropy. Randomly picking subspaces for additional evaluation also performs quite well in practice.

If the bottom up approach has not pruned the investigated subspace, the top down approach computes the entropy of the subspace. If the weighted normalized entropy is below β the subspaces is added to the result set and marked as one-sided homogeneous. The algorithm finally computes the combined entropy of all subspaces for which both subspaces are marked one-sided homogeneous in the result sets.

Once subspaces have been evaluated for **step 1**, the most complex algorithmic task has been solved. Having reduced the potentially exponential number of subspaces to the interesting ones, the actual clustering (**step 2**) is performed for each of these subspaces. This is done by computing the frequency and class entropy for all attribute value combinations in these subspaces. The resulting classifying subspace clusters then provide the model that is used for the actual classification (**step 3**). For incoming objects, compute the most similar classifying subspace clusters according to relative Hamming distance. If tied, compute reverse class entropy. The decision is then based on their class label distribution.

4 Experiments

Experiments were run on both synthetic and real world data. Synthetic data is used to show the correctness of our approach. Local patterns are hidden in a data set of 7.000 objects and eight attributes. As background noise, each attribute of the synthetic data set is uniformly distributed over ten values. On top of this, 16 different local patterns (subspace clusters) with different dimensionalities and different numbers of objects are hidden in the data set. Each local pattern contains two or three class labels among which one class label is dominating. We randomly picked 7.000 objects for training and 1.000 objects for testing.

The flight data contains historic data from a large European airport. For a three-month period, we trained the classifier on arrivals of two consecutive months and tested on the following month. Outliers with delays outside $[-60, 120]$ minutes have been eliminated. In total, 11.072 flights have been used for training and 5.720 flights for testing. Each flight has a total of 13 attributes, including e.g. the airline, flight number, aircraft type, routing, and the scheduled arrival time within the day. The class labels are "ahead of schedule", "on time" and "delayed". Finally we use two well-known real world data sets from the UCI KDD archive (Glass and Iris [14]), as a general benchmark.

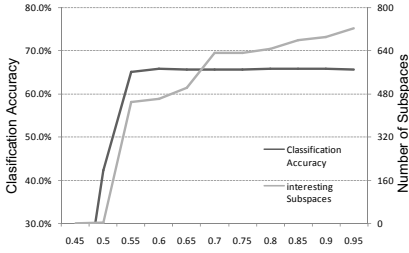
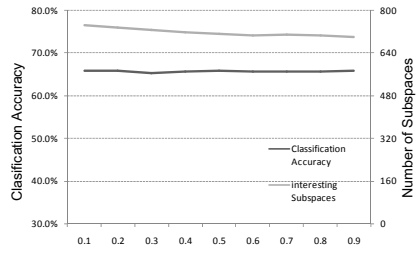
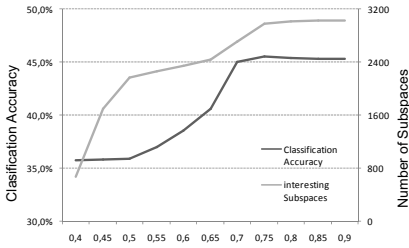
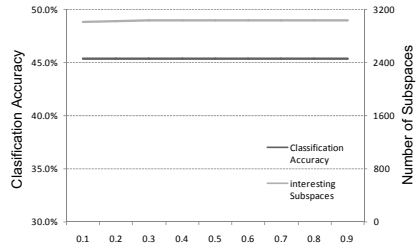
As mentioned before, preliminary experiments on the flight data indicate that no global relevance of attributes exist. Moreover, the data is inherently noisy, and important influences like weather conditions are not collected from scheduling. For realistic testing as in practical application, classifiers can only draw from existing attributes. Missing or not collected parameters are not available for training or testing neither in our experiments nor during the actual scheduling process.

We have conducted prior experiments to evaluate the effect of ϕ and γ for minimum frequency and maximum entropy thresholds, respectively. For each data set we used a cross validation to chose ϕ_1 (absolute frequency), ϕ_2 (relative frequency) and γ . For λ we have chosen 0.9. This value corresponds to a rather relaxed setting as we only want to remove completely inhomogeneous subspaces from consideration. To restrict the search space β can be set to a low value.

In our first experiments we develop a heuristic to set up reasonable parameters for the threshold β of the interestingness and the weight w of the class and attribute entropy, respectively.

Figure 4(a) illustrates varying β from 0.45 to 0.95 on the synthetic data, measuring classification accuracy and the number of classifying subspaces. The weight w for interestingness was set to 0.5. As expected, the number of classifying subspaces (CSS) decreases when lowering the threshold β . At the same time, the classification accuracy does not change substantially or even increases slightly when less subspaces are used. This effect may be related to the effect of overfitting. Using too many subspaces patterns are not sufficiently generalized, and noise is not removed. To set up the threshold β , slowly increasing β until the number of classifying subspace clusters shows a rapid rise, allows adjusting β to a point between generalization and overfitting. For both our data sets, a value around 0.65 obtains produces good results.

The effect of slightly increasing classification accuracy when reducing the number of subspaces can also be observed on the flight delay data (see Figure 4(c)). This confirms that the flight data contains local patterns for classification.

(a) Varying β on synthetic data(b) Varying w on synthetic data(c) Varying β on flight delay data(d) Varying w on flight delay data**Fig. 4.** Parameter evaluation using synthetic and real world data set

Varying parameter w yields the results depicted in the left part of Figure 4(b) and 4(d). The number of classifying subspaces decreases when giving more weight to attribute entropy. At the same time, classification accuracy does not change significantly. This robustness is due to the ensuing subspace clustering phase. As classification accuracy does not change this confirms that our classifying subspace cluster definition selects the relevant patterns. Setting $w = 0.5$ gives equivalent weight to the class and attribute entropy and hence is a good choice for pruning subspaces. We summarize our heuristics used to setup the parameters for our *SubClass* algorithm in Figure 5.

Next, we evaluate classification accuracy by comparing *SubClass* with other well-established classifiers that are applicable on nominal attributes: the k -NN classifier with Manhattan distance, the C4.5 decision tree that also uses a class and attribute entropy model [18], and a Naive Bayes classifier, a probabilistic classifier that assumes independence of attributes. Parameter settings use the best values from the preceding experiments.

Figure 6 illustrates the classification accuracy using four different data sets. In the noisy synthetic data set, our *SubClass* approach outperforms other classifiers. The large degree of noise and the varying class label distribution within the subspace clusters make this a challenging task. From the real world experiment on the flight data, depicted in Figure 6, we see that the situation is even more complex. Still, our *SubClass* method performs better than its competitors. This result supports our analysis that locally relevant information for classification exists that should be used for model building. Experts

Subspace Search Parameter

	Parameter	Value
β	Threshold for combined subspace interestingness	0.8-0.9
λ	Threshold for subspace interestingness	0.85-0.95
w	Weight for combined subspace interestingness	≥ 0.5

Subspace Clustering Parameter

	Parameter	Value
γ	Threshold for class information	0.6-0.7
ϕ_1	Threshold for absolute frequency	$0.01-0.005 \cdot I $
ϕ_2	Threshold for expected frequency	$3 \cdot 5 \cdot C $

Fig. 5. Parameters used by *SubClass*

	Flight Data	Synthetic Data	Iris	Glass
SubClass	45.4%	65.9%	96.27%	70.9%
C4.5	43.9%	58.0%	95.94%	66.8%
K-NN	42.4%	54.3%	93.91%	71.1%
Naive Bayes	42.8%	64.1%	95.27%	46.7%

Fig. 6. Classification accuracy on four data sets

from flight scheduling confirm that additional information on further parameters, e.g. weather conditions, is likely to boost classification. This information is inexistent in the current scheduling data that is collected routinely. *SubClass* exploits all the information available, especially locally relevant attribute and value combinations, for the best classification in this noisy scenario. Finally we evaluated the performance of *SubClass* on Glass and Iris [14]. The results indicate that even in settings containing no or little noise *SubClass* performs well.

5 Conclusion

Classification in noisy data with locally varying attribute relevance, as for our project in scheduling at airports, requires an approach that detects local patterns. Our *SubClass* method automatically detects classifying subspace clusters by incorporating class structure into the subspace search and the subspace clustering process. The general concept requires a definition of interesting subspaces for classification, of classifying subspace clusters and a classification scheme. Based on class and attribute value entropy, our *SubClass* ensures that clusters contain class-relevant information. Working both bottom-up and top-down on the lattice of subspaces, *SubClass* prunes irrelevant subspaces from the mining process. Our experiments on synthetic and real world data demonstrate that local structures are successfully detected and employed for classification, even in extremely noisy data.

References

1. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of ACM SIGMOD International Conference on Management of Data, pp. 94–105 (1998)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proceedings of International Conference on Very Large Databases (VLDB), pp. 487–499 (1994)
3. Aha, D., Kibler, D., Albert, M.: Instance-based learning algorithms. *Machine Learning* 6(1), 37–66 (1991)
4. Assent, I., Krieger, R., Glavic, B., Seidl, T.: Spatial multidimensional sequence clustering. In: Proceedings of International Workshop on Spatial and Spatio-Temporal Data Mining (SSTDM), conjunction with IEEE International Conference on Data Mining (ICDM) (2006)
5. Assent, I., Krieger, R., Müller, E., Seidl, T.: DUSC: Dimensionality unbiased subspace clustering. In: Proceedings of IEEE International Conference on Data Mining (ICDM) (2007)
6. Bolat, A.: Procedures for providing robust gate assignments for arriving aircrafts. *European Journal of Operational Research* 120, 63–80 (2000)
7. Bureau of Transportation Statistics. Airline on-time performance data, <http://www.transtats.bts.gov>
8. Cheng, C., Fu, A., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 84–93 (1999)
9. Domeniconi, C., Peng, J., Gunopulos, D.: Locally adaptive metric nearest-neighbor classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24(9), 1281–1285 (2002)
10. Duda, R., Hart, P., Stork, D.: *Pattern Classification*, 2nd edn. Wiley, Chichester (2000)
11. Eurocontrol Central Office for Delay Analysis. Delays to air transport in europe, <http://www.eurocontrol.int/eCoda>
12. Gray, R.: *Entropy and Information Theory*. Springer, Heidelberg (1990)
13. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, San Francisco (2001)
14. Hettich, S., Bay, S.: The uci kdd archive. University of California, Department of Information and Computer Science, Irvine, CA (1999), <http://kdd.ics.uci.edu>
15. Kailing, K., Kriegel, H.-P., Kröger, P.: Density-connected subspace clustering for high-dimensional data. In: Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 246–257 (2004)
16. Li, W., Han, J., Pei, J.: CMAR: accurate and efficient classification based on multipleclass-association rules. In: Proceedings of IEEE International Conference on Data Mining (ICDM), pp. 369–376 (2001)
17. Platt, J.: Fast training of support vector machines using sequential minimal optimization. In: Schoelkopf, Burges, Smola (eds.) *Advances in Kernel Methods*, MIT Press, Cambridge (1998)
18. Quinlan, J.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1992)
19. Shannon, C., Weaver, W.: *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Illinois (1949)
20. Silva, L., de Sa, J.M., Alexandre, L.: Neural network classification using shannon’s entropy. In: Proceedings of European Symposium on Artificial Neural Networks (ESANN) (2005)
21. Washio, T., Nakanishi, K., Motoda, H.: Deriving Class Association Rules Based on Level-wise Subspace Clustering. In: Jorge, A.M., Torgo, L., Brazdil, P.B., Camacho, R., Gama, J. (eds.) *PKDD 2005. LNCS (LNAI)*, vol. 3721, pp. 692–700. Springer, Heidelberg (2005)
22. Zhu, X.: Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison (2005)

Mining Quality-Aware Subspace Clusters

Ying-Ju Chen, Yi-Hong Chu, and Ming-Syan Chen

Department of Electrical Engineering,
National Taiwan University, Taipei, Taiwan, ROC
{yjchen, yihong}@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

Abstract. In this paper, we study the quality issue of subspace clusters, which is an important but unsolved challenge in the literature of subspace clustering. After binning the data set into disjoint grids/regions, current solutions of subspace clustering usually invoke a grid-based apriori-like procedure to efficiently identify dense regions level by level according to the monotonic property in so defined subspace regions. A cluster in a subspace is intuitively considered as a set of dense regions that each one is connected to another dense region in the cluster. The measure of what is a dense region is successfully studied in recent years. However, the rigid definition of subspace clusters as connected regions still needs further justification in terms of the two principal measures of clustering quality, i.e., the intra-cluster similarity and the inter-cluster dissimilarity. A true cluster is likely to be separated into two or more clusters, whereas many true clusters may be merged into a fat cluster. In this paper, we propose an innovative algorithm, called the QASC algorithm (standing for Quality-Aware Subspace Clustering) to effectively discover accurate clusters. The QASC algorithm is devised as a general solution to partition dense regions into clusters and can be easily integrated into most of grid-based subspace clustering algorithms. By conducting on extensive synthetic data sets, the experimental results reveal that QASC is effective in identifying true subspace clusters.

1 Introduction

Clustering has been studied for decades and recognized as an important and valuable capability in various fields. Recently, instead of clustering in the full dimensions, research in data mining has been in the direction of finding clusters which are embedded in subspaces. The increase of research attention for subspace clustering comes from the recent report of "The Curse of Dimensionality" [1], which points out that the distances between data points will be indiscriminate in the high dimensional space. Due to the infeasibility of clustering in high dimensional data, discovering clusters in subspaces becomes the mainstream of cluster research, including the work of projected clustering [8] and subspace clustering [2][3][5].

The CLIQUE algorithm is one of the state-of-the-art methodology in the literature, which essentially relies on the monotonicity property in the partition of grid-based regions:

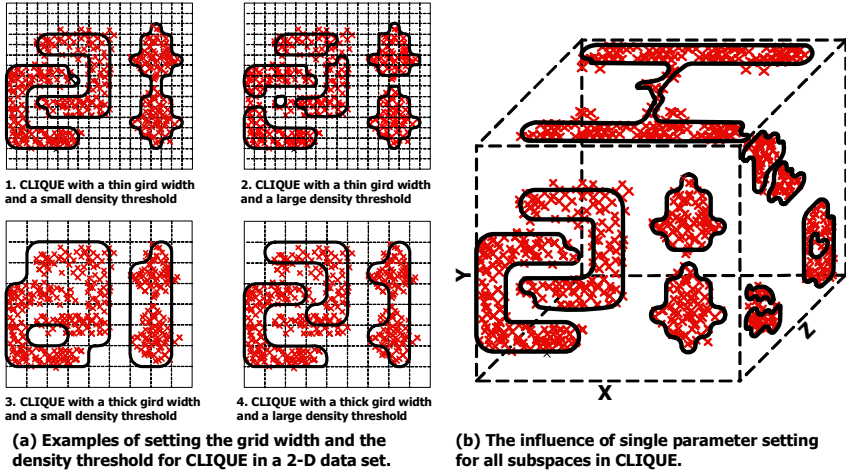


Fig. 1. Examples of quality issues in subspace clustering

Therefore, after binning input data into disjointed grids according to the coordinate of each data point, Apriori-based manners are able to efficiently identify dense grids level by level.

The measure of what is a dense region and the issue of how to efficiently and precisely identify dense regions have been comprehensively studied in recent years [2] [4] [5]. However, identifying clusters from connected dense grids, is deemed reasonable but does not be systematically evaluated yet. In fact, the rigid definition of subspace clusters as connected grids needs further justification in terms of two general criteria of clustering quality: (1) inter-cluster dissimilarity¹; and (2) intra-cluster similarity². We note that rashly connecting dense grids as clusters inevitably faces the compromise between inter-cluster dissimilarity and intra-cluster similarity, since the naive approach will amplify the side-effect from the misadjustment of two subtle input parameters, i.e., (1) the binning width of a grid and (2) the density threshold for identifying whether a region/grid is dense. With an inappropriate parameter setting, a true cluster is likely to be separated into two or more clusters, whereas many true clusters may be merged into a fat but improper cluster.

Consider the illustrative examples shown in Figure 1(a), which contain four situations in a two-dimensional space with different input parameters in CLIQUE. It is clear to see that different parameter settings result in highly divergent results when we straightforwardly link dense grids and construct clusters. Since

¹ The inter-cluster dissimilarity is used to reflect dissimilarity between two clusters. Different clusters are generally considered with dissimilar behavior and characters.

² The intra-cluster similarity refers to the measure of how similar the members in a cluster are. Intuitively, data within a valid cluster are more similar to each other than to a member belonging to a different cluster.

dense grids may distribute apart from each other when the connectivity between dense grids is relatively sparse, clusters could be separated into lots of slivers in CLIQUE, such as the case in Figure II(a).2, or the shapes of clusters could be distorted, such as the case in Figure II(a).4. As a result, the inter-cluster dissimilarity is strikingly sacrificed. On the other hand, true clusters could be merged into a few fat clusters when we have the crowded connection between dense grids, such as the result in Figure II(a).1 and Figure II(a).3. In such cases, we have the undesired loss of intra-cluster similarity in the clustering result.

Figure II(b) illustrates another critical limitation in current subspace clustering algorithms. Essentially, users could identify a set of parameters which is able to precisely discover all clusters embedded in a subspace, such as the result in the 2-dimensional subspace $\{X, Y\}$ shown in Figure II(b). However, there are numerous subspaces and using the same parameter setting is difficult to capture the best clustering characteristics for different subspaces due to variety of their distributions. Consider the 2-D subspaces $\{X, Z\}$ and $\{Y, Z\}$ in Figure II(b) as examples, where the result in $\{X, Z\}$ is expected to have two separated clusters without linkages, and the result in $\{Y, Z\}$ is expected to have three clusters with near-circular shapes instead of a set of small clusters with irregular shapes.

As a result, we present in this paper an approach, called QASC, (standing for **Q**uality-**A**ware **S**ubspace **C**lustering) to accurately construct subspace clusters from dense grids. Specifically, in order to conserve data characteristics within each subspace clusters, QASC takes the data distribution into consideration. Given a set of dense grids, QASC is devised as a two-phase algorithm to merge dense grids: (1) dense grids are partitioned into numerous small groups, where neighbor grids are located in the same group iff they are identified belonging to the same area influenced by a density function; (2) deliberately merge these small groups according to their distances and density functions by a hierarchical clustering manner.

The remaining of the paper is organized as follows. In Section 2, related works on subspace clustering are presented. In Section 3, we give the model and algorithm of QASC. Section 4 presents the experimental results. The paper concludes with Section 5.

2 Related Works

Without loss of generality, previous works on density-based subspace clustering for high dimensional data can be classified into two categories according to whether the grid structure is applied or not. Most of these algorithms utilize the grid structure, and the CLIQUE algorithm is the representative of such grid-based algorithms. On the other hand, a few works, e.g., the SUBCLU algorithm, can identify subspace clusters without use of grids.

Specifically, in the first step of CLIQUE, the data space is binned into equi-sized and axis-parallel units, where the width ξ of each dimension of a unit is one user-specified parameter. Afterward, the second step of CLIQUE exploits an apriori-like method to recursively identify all dense units in a bottom-up

way, where a dense unit is a unit whose density exceeds another user-specified threshold τ .

The use of grids can greatly reduce the computational complexity [6]. However, CLIQUE inevitably incurs many limitations from (1) using the support as a measure of interesting grids and (2) setting the subtle grid width. Consequently, the SUBCLU algorithm [3] and its extension utilize the idea of density-connected clusters from the DBSCAN algorithm without the use of grids. Giving two parameters ϵ and m in SUBCLU, the core objects are defined as the data points containing at least m data points in their ϵ -neighborhood. Since the definition of core objects also has the monotonicity property, clusters can be considered as a number of density-connected core objects with their surrounding objects, and identified in a bottom-up manner like CLIQUE. In general, SUBCLU can avoid the limitations of grids. However, the computation is higher than grid-based solutions. In addition, it also leaves the user with the responsibility of selecting subtle parameters. Even though users can empirically set parameter values that will lead to the discovery of acceptable clusters in a subspace, SUBCLU also has the problem illustrated in Figure 1(b) that clustering quality in other subspaces may be strikingly unsatisfactory.

Several variants of CLIQUE have been proposed to resolve the limitation of using the support as the measure of interesting grids. The ENCLUS algorithm in [2] utilizes entropy as a measure for subspace clusters instead of using support. The basic idea behind ENCLUS is that entropy of any subspace with clusters is higher than that of any subspace without clusters. The SCHISM algorithm is proposed to discover statistically "interesting" subspace clusters, where a cluster is interesting if the number of points it contains is statistically significantly higher than the number in the uniform distribution according to Chernoff-Hoeffding bound [7]. In addition, the MAFIA algorithm solves another limitation in CLIQUE. It uses adaptive, variable-sized grids whose widths are determined according to the distribution of data in each dimension [5]. As such, the side-effect from the rigid setting of grid widths in CLIQUE can be minimized. However, these new algorithms all merge dense/interesting grids as the same as CLIQUE. Depending on the connectivity between dense grids, they will face the same trade-off between inter-cluster dissimilarity and intra-cluster similarity in different subspaces as we show in Figure 1.

3 Quality Aware Subspace Clustering

3.1 Problem Description

We first introduce the notations used hereafter and then formalize the problem. Without loss of generality, we formalize the grid-based model by following the definition in CLIQUE. Specifically, let $S = A_1 \times A_2 \times \dots \times A_d$ be the d -dimensional data space formed by the d data attributes. A k -dimensional subspace is the space with the k dimensions drawn from the d attributes, where $k \leq d$.

In the grid-based subspace clustering, the data space S is first partitioned into a number of non-overlapping rectangular units by dividing each attribute

into δ intervals, where δ is an input parameter. Consider the projection of the dataset in a k -dimensional subspace. A k -dimensional interval u is defined as the intersection of one interval from each of the k attributes, and the support of u is defined as the number of data points contained in u . In CLIQUE, a grid is said a k -dimensional grid if its density exceeds a threshold τ , where τ is called "density threshold". Note that the definition of density grids is different between various subspace clustering algorithms, but subspace cluster is generally considered as disjointed sets of k -dimensional grids in CLIQUE and all its successors.

3.2 The QASC Algorithm

We then describe our algorithm, called QASC (the **Q**uality-**A**ware **S**ubspace **C**lustering algorithm), to deliver high-quality subspace clusters while considering the generality of the proposed model. We aim at improving the strategy of merging grids for the generality issue while conserving two general criteria of clustering quality, i.e., inter-cluster dissimilarity and intra-cluster similarity. To achieve this, the data distribution is taken into account. The basic idea behind our model is to construct small and disjointed groups of dense grids initially, where grids in each group are influenced by the same density function. Therefore, we are able to guarantee the intra-cluster similarity in the first phase. Afterward, we merge groups for improving the inter-cluster dissimilarity. We then formally present these two steps in the following sections, respectively.

Phase I of QASC: Identify Seed Clusters. The first step of QASC is to identify highly condensed group of dense grids, called seed clusters in this paper. We first have to present necessary definitions before introducing the solution to identify seed clusters.

Definition 1 (Grid Distance): Let $V_y = [a_1, a_2, \dots, a_k]$ and $V_{y'} = [a'_1, a'_2, \dots, a'_k]$ be two k -dimensional grids in subspace S^k , where a_i and a'_i are the i -th attribute values of y and y' respectively. The distance between y and y' is defined as:

$$Dist(y, y') = |V_y - V_{y'}|.$$

According to the definition, a grid y' is said a neighbor grid of y in S^k if $Dist(y, y') = 1$.

Definition 2 (Seed Grid): Let D be a density function defined on subspace S^k . A grid g is said a seed grid if g is a local maximum of D in S^k , i.e., $supp(g) \cap D$ is a local maximum of D .

Essentially, a seed grid is a local maximum in terms of the density in the k -dimensional space, and we are able to identify the set of seed grids in each subspace by a hill-climbing procedure.

Definition 3 (Density Function): Let y_{sg} be a seed grid in subspace S^k . The density function $f(y_{sg}, y) : S^d \rightarrow R_0^+$ is defined as:

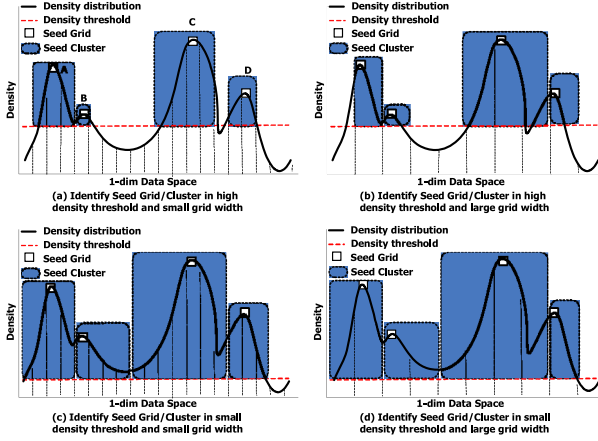


Fig. 2. Illustration of identifying seed grids and seed clusters in the 1-dim space with different parameters

$$f(y_{sg}, y) = \begin{cases} x, & x > 0, \text{ if } y \text{ is influenced by } y_{sg} \\ 0, & \text{if } y \text{ is not influenced by } y_{sg} \end{cases}.$$

In principle, the density function can be arbitrary. However, to conserve the nature characteristics in the input data without the assumption of the data distribution, the density function is specified according to the support distribution:

$$f(y_{sg}, y) = \begin{cases} \frac{\sup(y)}{\sup(y_{sg})}, & \text{if } \sup(y_{sg}) > \sup(y), \\ \exists y' \in D : \sup(y') \geq \sup(y), f(y_{sg}, y') > 0, \text{Dist}(y, y') = 1 & . \\ 0, & \text{else} \end{cases}$$

Based on the foregoing, we can define the seed cluster, which is used to denote the region influenced by a density function:

Definition 4 (Seed Cluster): $S^k = \{c_i \mid c_i = y_{sg} \cup \{y \in D \mid f(y_{sg}, y) > 0\}\}$.

Figure 2 shows the identification of seed grids and seed clusters, where these sets of dense grids in Figures 2(a)~Figure 2(d) are discovered with different parameters in CLIQUE. Clearly, a seed grid, e.g., grid A, B, C, or D, in Figure 2(a), has a local maximum density in the density distribution. In addition, a seed cluster wrt a seed grid y_{sg} covers a set of grids surrounding y_{sg} which are with the same distribution trend, indicating that grids within a seed cluster are highly condensed. Clearly, seed clusters can be considered as a set of most strictly defined clusters and the intra-cluster similarity can be entirely conserved in seed clusters.

Note that seed clusters inherently cannot contain grids with the density smaller than the density threshold in CLIQUE even though these grids may

satisfy the definition of density function. It is the natural limitation from the process of identifying dense grids. Nevertheless, as can be seen in Figure 2, four major seed clusters are all identified in various situations, showing the identification of seed clusters can robustly distinguish characteristics in groups of grids. On the other hand, clusters cannot be separated in Figure 2(c) and Figure 2(d) if we rashly connect dense grids into subspace clusters. The intra-cluster similarity is inevitably sacrificed.

The whole procedure to identify seed clusters in a subspace is outlined in Procedure 1. Specifically, the given set of dense grids should be sorted in order of decreasing grid density. Therefore, we can identify the seed grid from the root of the list and utilize a hill-climbing manner to search if a grid belongs to the generated seed grid. If a connected grid y_i is identified satisfying Definition 4, we set $y_i.cluster$ pointed to the corresponding seed cluster. The next grid in the sorted list is skipped if it has been identified belonging to a seed cluster. Otherwise, the procedure is iteratively executed until we have identified the cluster index for each grid. Finally, the set of seed clusters is returned.

Procedure: *Seed_Identify()*:

Input:

dense grids $D = \{y_1, y_2, \dots, y_m\}$

Output:

seed cluster $C = \{c_1, c_2, \dots, c_n\}$

1. $S_D := Sort(D)$; /*sort dense grids according to the density*/
2. for each dense grid $sy_i \in S_D$ do
3. if $sy_i.cluster = NULL$ then
4. let c_j be a new seed cluster;
5. $c_j.seed_grid = sy_i$;
6. $sy_i.cluster = c_j$;
7. *hill_climbing*($sy_i, c_j, sy_i.density$);
8. $C = C \cup c_j$;
9. end if
10. end for

Procedure: *hill_climbing()*:

Input:

Dense grid y_i ; Seed Cluster c_j ; Integer *count*;

1. if ($y_i.density \leq count = true$) then
2. $y_i.cluster = c_j$; /*identify that y_i belongs to seed cluster c_j */
3. for each dimension a_t of grid y_i do
4. $y_{left} = Left_Neighbor(y_i, a_t)$; /*return the left grid wrt the dimension a_t */
5. if ($y_{left} \neq NULL$) and ($y_{left}.cluster = NULL$)
6. *hill_climbing*($y_{left}, c_j, y_i.density$);
7. $y_{right} = Right_Neighbor(y_i, a_t)$; /*return the right grid wrt the dimension a_t */
8. if ($y_{right} \neq NULL$) and ($y_{right}.cluster = NULL$)
9. *hill_climbing*($y_{right}, c_j, y_i.density$);
10. end for
11. end if

Phase II of QASC: Merge Seed Clusters. In essence, the seed clusters conserve the intra-cluster similarity. The inter-cluster dissimilarity is not considered yet. As shown in Figure 2(a), it is expected that seed clusters A and B belong to the same cluster because they have the same trend of distribution and

are quite close to each other. Similarly, seed clusters C and D follow the same distribution. Note that the gap with the grid distance equal to one between seed clusters C and D may occur due to noise and the choice of the grid cutting-line. It is desired to have the clustering result with only two clusters in terms of both the intra-cluster similarity and the inter-cluster dissimilarity.

The second step of QASC is thus to deliberately merge seed clusters by a hierarchical clustering manner, where the distance between clusters is taken into consideration. Here we define the cluster grid distance first.

Definition 5 (Cluster Grid Distance): Let c_i and c_j be two clusters in S^k . The cluster grid distance between c_i and c_j is defined as

$$CDist(c_i, c_j) = \min \{Dist(y_i, y_j) \mid y_i \in c_i, y_j \in c_j\}.$$

The general criterion to merge two seed clusters is that they should be close to each other, i.e., they have the small $CDist(c_i, c_j)$. In addition, their seed grids should also be close to each other and the difference of the cluster sizes should be significantly large, meaning that they are likely to belong to the same distribution. As such, we build a global heap for merging clusters. The heap is sorted by the weight defined as:

$$weight(c_i, c_j) = \frac{size_ratio(c_i, c_j)}{MinSeedGridDist(c_i, c_j)} \times \frac{1}{CDist(c_i, c_j)},$$

where $size_ratio(c_i, c_j) = \max\{\frac{|c_i|}{|c_j|}, \frac{|c_j|}{|c_i|}\}$, and $|c_i|$ and $|c_j|$ are the number of points in clusters c_i and c_j , respectively. In addition, $MinSeedGridDist(c_i, c_j) = \min\{Dist(y_i, y_j)\}$, where y_i is a seed grid in c_i and y_j is a seed grid in c_j .

Importantly, clusters with a quite large $CDist(c_i, c_j)$ are not permitted to be merged even though $weight(c_i, c_j)$ is large. Note that it is reasonable to consider merging clusters with a small distance gap such as the example of seed clusters C and D in Figure 2(a). It is sufficient to avoid the influence from noise or the choice of the grid cutting-line if we permit a tolerant grid distance equal to one. As such, the prerequisite to insert the cluster pair into the heap is $CDist(c_i, c_j) \leq 2k$, where k is the number of dimensions in the subspace S^k .

The procedure in QASC to hierarchically merge clusters is outlined in Procedure 1, where the input is the set of seed clusters in S^k . Note that while two clusters c_i and c_j are merged, all information of cluster pairs in the heap related to c_i and c_j should be updated according to their new weight value.

Another criterion to determine if two clusters should be merged is shown in Line 11 in Procedure 1. Essentially, it is not desired to merge two clusters if they have similar cluster sizes because they are difficult to follow a single distribution trend. We set $\delta = 1.2$ in default to ensure the merged clusters are of variant sizes. Finally, the set of remaining clusters are returned when the heap is empty.

Procedure: *Seed_Merge()*:

Input:

Seed clusters $C = \{c_1, c_2, \dots, c_m\}$ in the subspace S^k

Output:

Subspace clusters in the subspace S^k

1. for each seed cluster $c_i \in c$ do
2. for each seed cluster $c_j \in c, c_j \neq c_i$ do
3. if $CDist(c_i, c_j) \leq 2 \times k$ then
4. $weight = \frac{size_ratio(c_i, c_j)}{MinSeedGridDist(c_i, c_j)} \times \frac{1}{CDist(c_i, c_j)}$;
5. *insertHeap*($c_i, c_j, weight$); /*insert the pair c_i, c_j in the heap sorted by the weight value*/
6. end if
7. end for
8. end for
9. while (*Heap* \neq *NULL*) do
10. $\{c_i, c_j\} = popHeapHead(Heap)$;
11. if $size_ratio(c_i, c_j) \geq \delta$ then
12. $c_i = c_i \cup c_j$;
13. remove c_j from *C*;
14. *QueuesUpdate*(c_i, c_j);
15. end if
16. end while

4 Experimental Studies

We assess the result of QASC in Windows XP professional platform with 1Gb memory and 1.7G P4-CPU. In this section, we call the methodology to rashly merge dense grids as the *naive* approach, which is used in CLIQUE and all grid-based subspace clustering algorithms. For fair comparison, we generate dense grids by the first step in CLIQUE for QASC and the naive approach. Note that our goal is to provide an effective approach for merging grids, and the current grid-based subspace algorithms all utilize the naive approach. The benefit from QASC for these variant algorithms is expected if we can gain good clustering quality for CLIQUE. All necessary codes are implemented by Java and compiled by Sun jdk1.5.

Note that various approaches to identify dense grids in subspaces introduce various parameters which would affect the clustering quality. We study the sensitivity of the QASC algorithm and the naive algorithm in various parameter setting of CLIQUE. For visualization reasons, the sensitivity analysis is studied in two dimensional spaces as the evaluation method used in traditional clustering algorithms. The result of the first study is shown in Figure 3, where a synthetic data with 6,547 points is used. Note that CLIQUE introduces two parameters, i.e., (1) the number of grids in each dimensions and (2) the density threshold, which are specified as "grid" and "minsup" in figures. Clearly, two clusters with similar diamond-like shapes are expected in the clustering results. However, the naive approach cannot capture the best result in this datasets wrt different parameter setting of CLIQUE. Figure 3(a) shows that the naive approach tends to merge clusters in high connectivity between dense grids, whereas Figures 3(b) and 3(c) show that many clusters are reported if dense grids distribute sparsely.

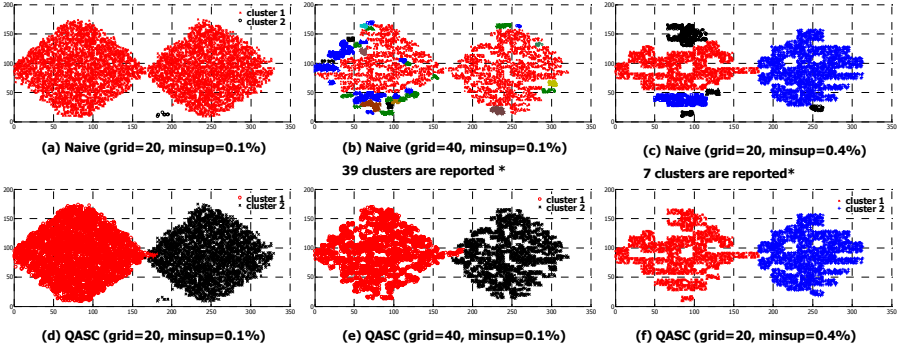


Fig. 3. The sensitivity studies on different parameter setting of CLIQUE

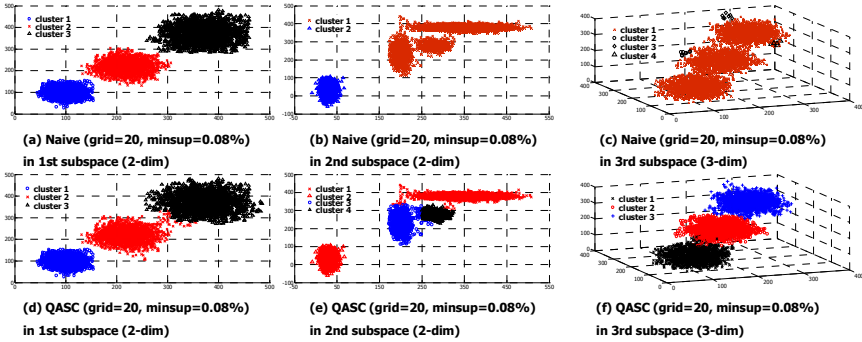


Fig. 4. Results of subspace clusters in different subspaces

On the other hand, Figures 3(d)~(f) show that QASC results in acceptable results with two expected clusters. Specifically, the two clusters are separated in Figure 3(d) because QASC does not merge two clusters with similar sizes. In addition, it is worth mentioning that QASC permits the combination of clusters when they are distributed with a gap equal to one. Therefore, QASC can report two acceptable clusters, as shown in Figure 3(f), to avoid the side-effect from the improper parameter setting of subspace clustering algorithms, indicating the robustness of QASC.

We study the sensitivity issue in another 7-dimensional synthetic data with 6,500 points. The data is generated by embedding clusters in two 2-dimensional spaces and a 3-dimensional space. The clustering results in these subspaces are shown in Figure 4. In this case, we set grid=20 and the density threshold equal to 0.08%, which is able to correctly retrieve three clusters in the first subspace for the naive approach. However, similar to the example illustrated in Figure 1(b), this parameter setting is difficult to make correct clustering result for other subspaces. In contrast, QASC can retrieve accurate subspace clusters in other subspaces since the data distribution is taken into consideration.

5 Conclusions

In this paper, we proposed an effective algorithm, QASC, to merge dense grids for generating high-quality subspace clusters. QASC is devised as a two-step method, where the first step generates seed clusters with high intra-cluster similarity and the second step deliberately merges seed clusters to construct subspace clusters with high inter-cluster dissimilarity. QASC is devised as a general approach to merge dense/interesting grids, and can be easily integrated into most of grid-based subspace clustering algorithms in place of the naive approach of rashly connecting dense grids as clusters. We complement our analytical and algorithmic results by a thorough empirical study, and show that QASC can retrieve high-quality subspace clusters in various subspaces, demonstrating its prominent advantages to be a practicable component for subspace clustering.

References

1. Beyer, K., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is Nearest Neighbor Meaningful? In: Proc. of ICDT Conference (1999)
2. Cheng, C.H., Fu, A.W., Zhang, Y.: Entropy-Based Subspace Clustering for Mining Numerical Data. ACM SIGKDD (1999)
3. Kailing, K., Kriegel, H.-P., Kroger, P.: Density-Connected Subspace Clustering for High-Dimensional Data. In: SDM (2004)
4. Kriegel, H.-P., Kroger, P., Renz, M., Wurst, S.: A generic framework for efficient subspace clustering of high-dimensional data. In: IEEE ICDM (2005)
5. Nagesh, H., Goil, S., Choudhary, A.: Adaptive grids for clustering massive data sets. In: Proc. of the SIAM Intern'l Conference on Data Mining (SDM) (2001)
6. Parsons, L., Haque, E., Liu, H.: Subspace Clustering for High Dimensional Data: A Review. ACM SIGKDD Explorations Newsletter (2004)
7. Sequeira, K., Zaki, M.: Schism: A new approach for interesting subspace mining. In: Proc. of the IEEE 4th Intern'l Conf. on Data Mining (ICDM) (2004)
8. Yip, K.Y., Cheung, D.W., Ng, M.K.: Harp: A practical projected clustering algorithm. IEEE Trans. Knowl. Data Eng. 16(11) (2004)

A Decremental Approach for Mining Frequent Itemsets from Uncertain Data^{*}

Chun-Kit Chui and Ben Kao

Department of Computer Science, The University of Hong Kong,
Pokfulam, Hong Kong
{ckchui, kao}@cs.hku.hk

Abstract. We study the problem of mining frequent itemsets from *uncertain data* under a *probabilistic model*. We consider transactions whose items are associated with *existential probabilities*. A *decremental pruning* (DP) technique, which exploits the statistical properties of items' existential probabilities, is proposed. Experimental results show that DP can achieve significant computational cost savings compared with existing approaches, such as U-Apriori and LGS-Trimming. Also, unlike LGS-Trimming, DP does not require a user-specified trimming threshold and its performance is relatively insensitive to the population of low-probability items in the dataset.

1 Introduction

Frequent itemset mining (FIM) is a core component in many data analysis tasks such as association analysis [1] and sequential-pattern mining [2]. Traditionally, FIM is applied to data that is certain and precise. As an example, a transaction in a market-basket dataset registers items that are purchased by a customer. Applying FIM on such a dataset allows one to identify items that are often purchased together. In this example, the presence/absence of an item in a transaction is known with certainty. Existing FIM algorithms, such as the well-known Apriori algorithm [1] and other variants, were designed for mining “certain” data.

Most of the previous studies on FIM assume a data model under which transactions capture doubtless facts about the items that are contained in each transaction. However, in many applications, the existence of an item in a transaction is best captured by a probability. As an example, consider experiments that test certain drug-resistant properties of pathogens. Results of such tests can be represented by a transactional dataset: each pathogen is represented by a transaction and the drugs it shows resistance to are listed as items in the transaction. Applying FIM on such a dataset allows us to discover multi-drug-resistant associations [3]. In practice, due to measurement and experimental errors, multiple measurements or experiments are conducted to obtain a higher confidence of the

^{*} This research is supported by Hong Kong Research Grants Council Grant HKU 7134/06E.

results. In such cases, the existence of an item or property in a transaction should be expressed in terms of a probability. For example, if *Streptococcus Pneumoniae* (a pathogen) shows resistance to penicillin (an antibiotics drug) 90 times out of 100 experiments, the probability that the property “penicillin-resistant” in *Streptococcus Pneumoniae* is 90%. We call this kind of probability *existential probability*. In this paper we study the problem of applying FIM on datasets under the *existential uncertain data model*, in which each item is associated with an existential probability that indicates the likelihood of its presence in a transaction. Table 1 shows an example of an existential uncertain dataset.

Table 1. An existential uncertain dataset with 2 transactions t_1, t_2 and 2 items a, b

Transaction \ Item	a	b
t_1	90%	80%
t_2	40%	70%

The problem of mining frequent itemsets under the existential uncertain data model was first studied in [4]. The Apriori algorithm was modified to mine uncertain data. The modified algorithm, called U-Apriori, was shown to be computationally inefficient. A data trimming framework (LGS-Trimming) was proposed to reduce the computational and I/O costs of U-Apriori. As a summary, given an existential uncertain dataset D , LGS-Trimming creates a trimmed dataset D^T by removing items with low existential probabilities in D . The trimming framework works under the assumption that a non-trivial portion of the items in the dataset are associated with low existential probabilities (e.g., a pathogen may be highly resistant to a few drugs but not so for most of the others). Based on this assumption, the size of D^T is significantly smaller than D and mining D^T instead of D has the following advantages:

- The I/O cost of scanning D^T is smaller.
- Since many low-probability items have been removed, transactions in D^T are much smaller. Hence, there are a lot fewer subsets contained in transactions leading to much faster subset testing of candidate itemsets and faster support counting.

However, there are disadvantages of the trimming framework. First, there is the overhead of creating D^T . Second, since D^T is incomplete information, the set of frequent itemsets mined from it is only a subset of the complete set. A patch-up phase (and thus some overhead) is therefore needed to recover those missed frequent itemsets. As a result, if there are relatively few low-probability items in D , then D^T and D will be of similar sizes. The savings obtained by LGS-Trimming may not compensate for the overhead incurred. The performance of LGS-Trimming is thus sensitive to the percentage (R) of items with low existential probabilities. Trimming can be counter-productive when R is very low. Third, a trimming threshold ρ_t (to determine “low” probability) is needed, which in some cases could be hard to set. A large ρ_t implies a greater reduction of the

size of D but a larger overhead in the patch-up phase to recover missed frequent itemsets. On the other hand, a small ρ_t would trim D by little extent resulting in little savings. The performance of Trimming is thus sensitive to ρ_t . In [4], it was assumed that the existential probabilities of items in a dataset follow a β -distribution. That is, most items' can be classified as very-high-probability ones or very-low-probability ones. There were few items with moderate existential probabilities. In that case, it is easy to determine ρ_t as there is a clearcut distinction between high and low existential probabilities. It would be harder to select an appropriate ρ_t if the distribution of existential probabilities is more uniform.

In this paper we propose an alternative method, called *Dynamic Pruning* (DP), for mining frequent itemsets from existential uncertain data. As we will discuss in later sections, DP exploits the statistical properties of existential probabilities to gradually reduce the set of candidate itemsets. This leads to more efficient support counting and thus significant CPU cost savings. Comparing with LGS-Trimming, DP has two desirable properties: (1) it does not require a user-specified trimming threshold; (2) its performance is relatively less sensitive to R , the fraction of small-probability items in the dataset. DP is thus more applicable to a larger range of applications. Moreover, we will show that DP and LGS-Trimming are complementary to each other. They can be combined to achieve an even better performance.

The rest of this paper is organized as follows. Section 2 describes the mining problem and revisits the brute force U-Apriori algorithm. Section 3 presents the DP approach. Section 4 presents some experimental results and discusses some observations. We conclude the study in Section 5.

2 Preliminaries

In the existential uncertain data model, a dataset D consists of d transactions t_1, \dots, t_d . A transaction t_i contains a number of items. Each item x in t_i is associated with a probability $P_{t_i}(x)$, which indicates the likelihood that item x is present in transaction t_i . A *probabilistic* model [5] can be applied to interpret an existential uncertain dataset. Basically, each probability $P_{t_i}(x)$ associated with an item x derives two possible worlds, say, W_1 and W_2 . In World W_1 , item x is present in transaction t_i ; In World W_2 , item x is not in t_i . Let $P(W_j)$ be the probability that World W_j being the true world, then we have $P(W_1) = P_{t_i}(x)$ and $P(W_2) = 1 - P_{t_i}(x)$. This idea can be extended to cover cases in which transaction t_i contains other items. For example, let y be another item in t_i with probability $P_{t_i}(y)$. Assume that the observations of item x and item y are independently done, then there are four possible worlds. In particular, the probability of the world in which t_i contains both items x and y is $P_{t_i}(x) \cdot P_{t_i}(y)$. We can further generalize the idea to datasets that contain more than one transaction. Figure 1 illustrates the 16 possible worlds derived from the dataset shown in Table 1.

¹ If an item has 0 existential probability, it does not appear in the transaction.

W_1		W_2		W_3		W_4		W_5		W_6		W_7		W_8			
	a	b		a	b		a	b		a	b		a	b		a	b
t_1	✓	✓	t_1	✓	✓	t_1	✓	✓	t_1	✓	✗	t_1	✗	✗	t_1	✗	✗
t_2	✓	✓	t_2	✓	✗	t_2	✗	✓	t_2	✓	✓	t_2	✗	✗	t_2	✓	✓

W_9		W_{10}		W_{11}		W_{12}		W_{13}		W_{14}		W_{15}		W_{16}			
	a	b		a	b		a	b		a	b		a	b		a	b
t_1	✗	✓	t_1	✗	✓	t_1	✓	✗	t_1	✗	✗	t_1	✗	✓	t_1	✓	✗
t_2	✗	✓	t_2	✗	✓	t_2	✗	✓	t_2	✓	✗	t_2	✗	✗	t_2	✗	✗

Fig. 1. 16 possible worlds derived from dataset with 2 transactions and 2 items

In traditional frequent itemset mining, the support count of an itemset X is defined as the number of transactions that contain X . For an uncertain dataset, such a support value is undefined since set containment is probabilistic. However, we note that each possible world derived from an uncertain dataset is certain, and therefore support counts are well-defined with respect to each world. For example, the support counts of itemset $\{a, b\}$ in Worlds W_1 and W_6 (Figure 1) are 2 and 1, respectively. In [4], the notion of $\text{freq}(X)$ was proposed as a frequency measure. Let W be the set of all possible worlds derivable from an uncertain dataset D . Given a world $W_j \in W$, let $P(W_j)$ be the probability of World W_j ; $S(X, W_j)$ be the support count of X with respect to W_j ; and $T_{i,j}$ be the i^{th} transaction in World W_j . Assuming that items' existential probabilities are determined through independent observations, then $P(W_j)$ and the expected support $S_e(X)$ of an itemset X are given by the following formulae²:

$$P(W_j) = \prod_{i=1}^{|D|} \left(\prod_{x \in T_{i,j}} P_{t_i}(x) \cdot \prod_{y \notin T_{i,j}} (1 - P_{t_i}(y)) \right), \text{ and} \quad (1)$$

$$S_e(X) = \sum_{j=1}^{|W|} P(W_j) \times S(X, W_j) = \sum_{i=1}^{|D|} \prod_{x \in X} P_{t_i}(x). \quad (2)$$

Problem Statement. Given an existential uncertain dataset D and a user-specified support threshold ρ_s , the problem of mining frequent itemsets is to return all itemsets X with expected support $S_e(X) \geq \rho_s \cdot |D|$.

U-Apriori, a modified version of the Apriori algorithm, was presented in [4] as a baseline algorithm to solve the problem. The difference between Apriori and U-Apriori lies in the way supports are counted. Given a candidate itemset X and a transaction t_i , Apriori tests whether $X \subseteq t_i$. If so, the support count of X is incremented by 1. Under U-Apriori, the support count of X is incremented by the value $\prod_{x \in X} P_{t_i}(x)$ instead (see Equation 2).

² Readers are referred to [4] for the details of the derivations.

3 Decremental Pruning

In this section we describe the *decremental pruning* (DP) technique, which exploits the statistical properties of the existential probabilities of items to achieve candidate reduction during the mining process. The basic idea is to estimate upper bounds of candidate itemsets' expected supports progressively after each dataset transaction is processed. If a candidate's upper bound falls below the support threshold ρ_s , the candidate is immediately pruned. To illustrate, let us consider a sample dataset shown in Table 2. Assume a support threshold $\rho_s = 0.5$, the minimum support count is $min_sup = 4 \times 0.5 = 2$. Consider the candidate itemset $\{a, b\}$. To obtain the expected support of $\{a, b\}$, denoted as $S_e(\{a, b\})$, U-Apriori scans the entire dataset once and obtains $S_e(\{a, b\}) = 1.54$, which is infrequent.

Table 2. An example of existentially uncertain dataset

Transaction \ Item	a	b	c	d
t_1	1	0.5	0.3	0.2
t_2	0.9	0.8	0.7	0.4
t_3	0.3	0	0.9	0.7
t_4	0.4	0.8	0.3	0.7

During the dataset scanning process, we observe that a candidate itemset X can be pruned before the entire dataset is scanned. The idea is to maintain a counter $\hat{S}_e(X, X')$ for some non-empty $X' \subset X$. The counter maintains an upper bound of the expected support count of X , i.e., $S_e(X)$. This upper bound's value is progressively updated as dataset transactions are processed. We use $\hat{S}_e(X, X', k)$ to denote the value of $\hat{S}_e(X, X')$ after transactions t_1, \dots, t_k have been processed.

Definition 1. Decremental Counter. For any non-empty $X' \subset X$, $k \geq 0$
 $\hat{S}_e(X, X', k) = \sum_{i=1}^k \prod_{x \in X} P_{t_i}(x) + \sum_{i=k+1}^{|D|} \prod_{x \in X'} P_{t_i}(x)$.

From Equation 2, we have

$$\begin{aligned}
 S_e(X) &= \sum_{i=1}^{|D|} \prod_{x \in X} P_{t_i}(x) \\
 &= \sum_{i=1}^k \prod_{x \in X} P_{t_i}(x) + \sum_{i=k+1}^{|D|} \prod_{x \in X'} P_{t_i}(x) \\
 &\leq \sum_{i=1}^k \prod_{x \in X} P_{t_i}(x) + \sum_{i=k+1}^{|D|} \left(\prod_{x \in X'} P_{t_i}(x) \cdot \prod_{x \in X-X'} 1 \right) \\
 &= \hat{S}_e(X, X', k).
 \end{aligned}$$

Hence, $\hat{S}_e(X, X', k)$ is an upper bound of $S_e(X)$. Essentially, we are assuming that the probabilities of all items $x \in X - X'$ are 1 in transactions $t_{k+1}, \dots, t_{|D|}$ in estimating the upper bound. Also, $\hat{S}_e(X, X', 0) = \sum_{i=1}^{|D|} \prod_{x \in X'} P_{t_i}(x) = S_e(X')$.

In our running example, suppose we have executed the first iteration of U-Apriori and have determined the expected supports of all 1-itemsets, in particular, we know $S_e(\{a\}) = 2.6$. At the beginning of the 2nd iteration, we have, for the candidate itemset $\{a, b\}$, $\hat{S}_e(\{a, b\}, \{a\}, 0) = S_e(\{a\}) = 2.6$. We then process the first transaction t_1 and find that $P_{t_1}(b)$ is 0.5 (instead of 1 as assumed when we calculated the upper bound), we know that we have overestimated $S_e(\{a, b\})$ by $P_{t_1}(a) \times (1 - P_{t_1}(b)) = 0.5$. Therefore, we refine the bound and get $\hat{S}_e(\{a, b\}, \{a\}, 1) = \hat{S}_e(\{a, b\}, \{a\}, 0) - 0.5 = 2.1$. Next, we process t_2 . By similar argument, we know that we have overestimated the support by $0.9 \times (1 - 0.8) = 0.18$. We thus update the bound to get $\hat{S}_e(\{a, b\}, \{a\}, 2) = \hat{S}_e(\{a, b\}, \{a\}, 1) - 0.18 = 1.92$. At this point, the bound has dropped below the support threshold. The candidate $\{a, b\}$ is thus infrequent and can be pruned.

Equation 3 summarizes the initialization and update of the decremental counter $\hat{S}_e(X, X', k)$:

$$\hat{S}_e(X, X', k) = \begin{cases} S_e(X') & \text{if } k = 0; \\ \hat{S}_e(X, X', k - 1) - S_e^{t_k}(X') \times \{1 - S_e^{t_k}(X - X')\} & \text{if } k > 0. \end{cases} \quad (3)$$

where $S_e^{t_k}(X') = \prod_{x \in X'} P_{t_k}(x)$ and $S_e^{t_k}(X - X') = \prod_{x \in X - X'} P_{t_k}(x)$.

From the example, we see that $\{a, b\}$ can be pruned before the entire dataset is scanned. This candidate reduction potentially saves a lot of computational cost. However, there are $2^{|X|} - 2$ non-empty proper subsets of a candidate itemset X . The number of decremental counters is thus huge. Maintaining a large number of decremental counters involves too much overhead, and the DP method could be counter-productive. We propose two methods for reducing the number of decremental counters while maintaining a good pruning effectiveness in the rest of this section.

Aggregate by Singletons (AS). The AS method reduces the number of decremental counters to the number of frequent singletons. First, only those decremental counters $\hat{S}_e(X, X')$ where X' is a frequent singleton are maintained. Second, given a frequent item x , the decremental counters $\hat{S}_e(X, \{x\})$ for any itemset X that contains x are replaced by a single counter $d_s(x)$. Let $d_s(x, k)$ be the value of $d_s(x)$ after the first k data transactions have been processed. Equation 4 shows the initialization and update of $d_s(x, k)$.

$$d_s(x, k) = \begin{cases} S_e(\{x\}) & \text{if } k = 0; \\ d_s(x, k - 1) - P_{t_k}(x) \times \{1 - \max_s(k)\} & \text{if } k > 0. \end{cases} \quad (4)$$

where $\max_s(k) = \max\{P_{t_k}(x') | x' \in t_k, x' \neq x\}$ returns the maximum existential probability among the items (except x) in transaction t_k .

One can prove by induction that $\hat{S}_e(X, \{x\}, k) \leq d_s(x, k)$ for any itemset X that contains item x . With the AS method, the aggregated counters can be organized in an array. During the mining process, if a counter's value $d_s(x, k)$ drops below the support requirement, we know that any candidate itemset X that contains x must not be frequent and hence can be pruned. Also, we can remove item x from the dataset starting from transaction t_{k+1} . Therefore, AS not only achieves candidate reduction, it also shrinks dataset transactions. The latter allows more efficient subset testing during support counting.

Common-Prefix Method (CP). The CP method aggregates the decremental counters of candidates with common prefix. Here, we assume that items follow a certain ordering Φ , and the set of items of an itemset is listed according to Φ . First, only decremental counters of the form $\hat{S}_e(X, X')$ where X' is a proper prefix of X (denoted by $X' \sqsubset X$) are maintained. Second, given an itemset X' , all counters $\hat{S}_e(X, X')$ such that $X' \sqsubset X$ are replaced by a single counter $d_p(X')$. Let $d_p(X', k)$ be the value of $d_p(X')$ after the first k data transactions have been processed. Equation 5 shows the initialization and update of $d_p(X', k)$.

$$d_p(X', k) = \begin{cases} S_e(X') & \text{if } k = 0; \\ d_p(X', k-1) - S_e^{t_k}(X') \times \{1 - \max_p(k)\} & \text{if } k > 0. \end{cases} \quad (5)$$

where $S_e^{t_k}(X') = \prod_{x \in X'} P_{t_k}(x)$ and $\max_p(k) = \max\{P_{t_k}(z) | z \text{ is after all the items in } X' \text{ according to the item ordering } \Phi\}$.

Again, by induction, we can prove that $\hat{S}_e(X, X', k) \leq d_p(X', k)$ for any $X' \sqsubset X$. Hence when $d_p(X', k)$ drops below the support requirement, we can conclude that any candidate itemset X such that $X' \sqsubset X$ must be infrequent and can thus be pruned. We remark that since most of the traditional frequent itemset mining algorithms apply a prefix-tree data structure to organize candidates [1][6][4], the way that CP aggregates the decremental counters facilitates its integration with the prefix-tree data structure.

Figure 2 shows the size-2 candidates of the dataset in Table 2 organized in a hash-tree data structure [1]. A hash-tree is essentially a prefix tree, where candidates with the same prefix are organized under the same sub-tree. A prefix is thus associated with a node in the tree. A prefix decremental counter $d_p(X')$ is stored in the parent node of the node that is associated with the prefix X' . For example, $d_p(b)$ is stored in the root node since the prefix b is at level 1 of the tree (the second child node shown in Figure 2). [1] presented a recursive strategy for searching candidates that are contained in each transaction using a hash-tree structure. We illustrate the steps of processing a transaction t_1 from our running example (see Table 2) and explain how the counter $d_p(b)$ is updated in Figure 2.

From the figure, we see that $d_p(b, 1) = 1.75$ after t_1 is processed. Since $d_p(b, 1)$ is an upper bound of the expected supports of $\{b, c\}$ and $\{b, d\}$, and since $d_p(b, 1)$ is smaller than the support requirement, we conclude that both $\{b, c\}$ and $\{b, d\}$ are infrequent and are thus pruned. With the hash-tree structure, we can virtually prune the candidates by setting the pointer $root.hash(b) = \text{null}$. Also, the counter $d_p(b)$ is removed from the root. As a result, the two candidates cannot

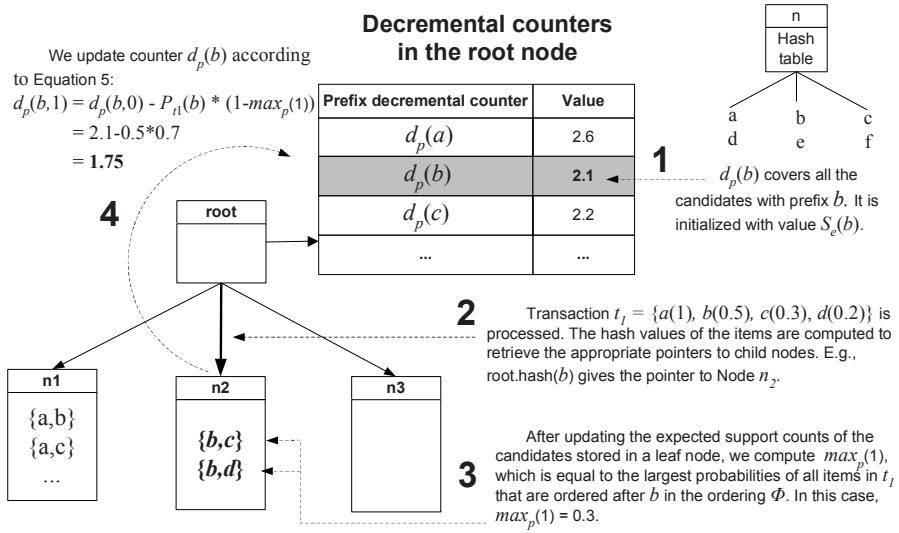


Fig. 2. A size-2 candidate hash tree with prefix decremental counters

be reached when subsequent transactions are processed. The computational cost of incrementing the expected support counts of the two candidates in subsequent transactions is saved.

Item ordering. According to Equation 5, the initial value of a counter $d_p(X')$ is given by $d_p(X', 0) = S_e(X')$, i.e., the expected support of the prefix X' . Since candidates are pruned if a prefix decremental counter drops below the support requirement, it makes sense to pick those prefixes X' such that their initial values are as small as possible. A heuristic would be to set the item ordering Φ in increasing order of items' supports. We adopt this strategy for the CP method.

4 Experimental Evaluation

We conducted experiments comparing the performance of the DP methods against U-Apriori and LGS-Trimming. The experiments were conducted on a 2.6GHz P4 machine with 512MB memory running Linux Kernel 2.6.10. The algorithms were implemented using C.

We use the two-step dataset generation procedure documented in [4]. In the first step, the generator uses the IBM synthetic generator [1] to generate a dataset that contains frequent itemsets. We set the average number of items per transaction (T_{high}) to 20, the average length of frequent itemsets (I) to 6, and the number of transactions (D) to 100K³. In the second step, the generator uses an

³ We have conducted our experiments using different values of T_{high} , I and D . Due to space limitation, we only report a representative result using $T_{high}20I6D100K$ in this paper.

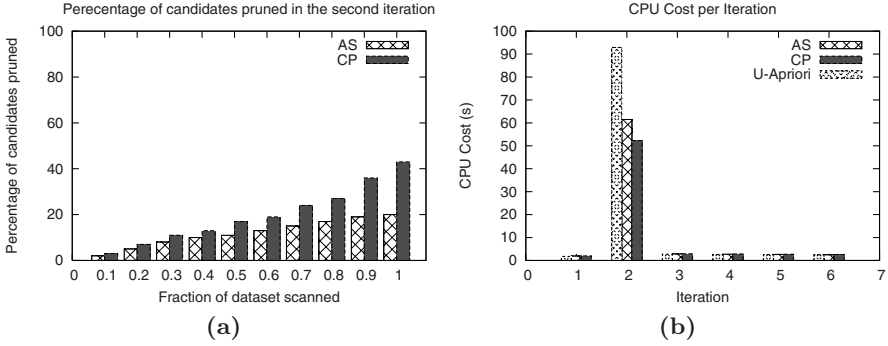


Fig. 3. a) Percentage of candidates pruned in the 2^{nd} iteration. b) CPU cost in each iteration.

uncertainty simulator to generate an existential probability for each item. The simulator first assigns each item in the dataset with a relatively high probability following a normal distribution with mean μ_{high} and standard deviation σ_{high} . To simulate items with low probabilities, the simulator inserts T_{low} items into each transaction. The probabilities of these items follow a normal distribution with mean μ_{low} and standard deviation σ_{low} . The average number of items per transaction, denoted by T , is equal to $T_{high} + T_{low}$. A parameter R is used to control the percentage of items with low probabilities in the dataset (i.e. $R = \frac{T_{low}}{T_{high} + T_{low}}$).

As an example, $T25/R20/I6/D100K/HB75/HD15/LB25/LD15$ represents an uncertain dataset with 25 items per transaction on average. Out of the 25 items, 20 are assigned with high probabilities and 5 are assigned with low probabilities. The high (low) probabilities are generated following a normal distribution with mean equal to 75% (25%) and standard deviation equal to 15% (15%). We call this dataset $\mathcal{D}_{T25/R20/I6/D100K/HB75/HD15/LB25/LD15}$.

4.1 Pruning Power of the Decremental Methods

In this section we investigate the pruning power of the decremental methods. The dataset we use is $\mathcal{D}_{T25/R20/I6/D100K/HB75/HD15/LB25/LD15}$ and we set $\rho_s = 0.1\%$ in the experiment. Figure 3a shows the percentage of candidates pruned by AS and CP in the second iteration after a certain fraction of the dataset transactions have been processed. For example, the figure shows that about 20% of the candidates are pruned by CP after 60% of the transactions are processed. From the figure, we observe that the pruning power of CP is higher than that of AS. In particular, CP prunes twice as many candidates as AS after the entire dataset is scanned.

Recall that the idea of AS and CP is to replace a group of decremental counters by either a singleton decremental counter (AS-counter) or a prefix decremental counter (CP-counter). We say that an AS- or CP-counter $d_{s/p}(X')$ “covers” a decremental counter $\hat{S}_e(X, X')$ if $\hat{S}_e(X, X')$ is replaced by $d_{s/p}(X')$. Essentially, an AS- or CP-counter serves as an upper bound of a group of decremental counters covered by it. In the 2^{nd} iteration, candidates are of size 2 and

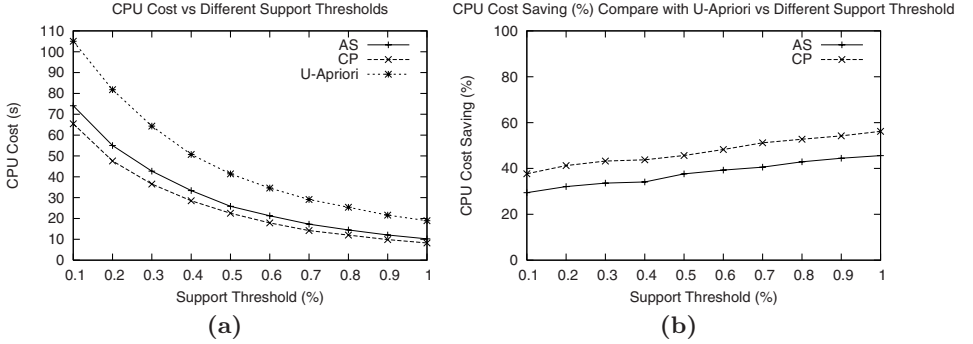


Fig. 4. CPU cost and saving with different ρ_s

therefore all proper prefixes contain only one item. We note that in general, a CP-counter, say $d_p(\{a\})$ covers fewer decremental counters than its AS counterpart, say $d_s(\{a\})$. This is because $d_p(\{a\})$ covers $\hat{S}_e(X, \{a\})$ only if $\{a\}$ is a prefix of X , while $d_s(\{a\})$ covers $\hat{S}_e(X, \{a\})$ only if $\{a\}$ is contained in X . Since prefix is a stronger requirement than containment, the set of counters covered by $d_p(\{a\})$ is always a subset of $d_s(\{a\})$. Therefore, each CP-counter “covers” fewer decremental counters than an AS-counter does. CP-counters are thus generally tighter upper bounds, leading to a more effective pruning.

Figure 3b shows the CPU cost in each iteration of the mining process. We see that in this experiment the costs of the 2^{nd} iteration dominates the others under all three algorithms. The pruning effectiveness of AS and CP in the 2^{nd} iteration (Figure 3a) thus reflects the CPU cost savings. For example, the 40% candidate reduction of CP translates into about 40s of CPU cost saving. Another observation is that although CP prunes twice as much as AS, the CPU cost saving of CP is not double of that of AS. This is because CP requires a more complex recursive strategy to maintain the prefix decremental counters, which is comparatively more costly.

4.2 Varying Minimum Support Threshold

Our next experiment compares the CPU costs of the DP methods against U-Apriori as the support threshold ρ_s varies from 0.1% to 1.0%. Figure 4a shows the CPU costs and Figure 4b shows the percentage of savings over U-Apriori. For example, when $\rho_s = 1\%$, CP saves about 59% of CPU time compared with U-Apriori. From the figures, we see that CP performs slightly better than AS over a wide range of ρ_s value. Also, the CPU costs of both CP and AS decrease as ρ_s increases. This is because a larger ρ_s implies fewer candidates and frequent itemsets, so the algorithms execute faster. Also, a larger ρ_s implies the minimum support requirement is larger. Hence, it is easier for the decremental counters to drop below the required value and more candidates can be pruned early.

4.3 Comparing with Data Trimming

Recall that LGS-Trimming consists of three steps: (1) remove low-probability items from dataset D to obtain a trimmed dataset D^T ; (2) mine D^T ; (3) patch up and recover missed frequent itemsets. LGS-Trimming and DP methods are orthogonal and can be combined. (DP can be applied to mining D^T and it also helps the patch-up step, which is essentially an additional iteration of candidate-generation and support-counting). In this section we compare U-Apriori, AS, CP, LGS-Trimming, and the combined method that integrates CP and LGS-Trimming. In particular, we study how the percentage of low-probability items (R) affects the algorithms' performance. In the experiment, we use $\rho_s = 0.1\%$. Figure 5a shows the CPU costs and Figure 5b shows the percentage of savings over U-Apriori. From Figure 5b, we see that the performance of LGS-Trimming is very sensitive to R . Trimming outperforms AS and CP when R is large (e.g., 50%). This is because when there are numerous low-probability items, the trimmed dataset D^T is very small, and mining D^T is very efficient. On the other hand, if R is small, Trimming is less efficient than DP methods, and it could even be counter-productive for very small R . This is because for small R , D^T is large, so not much savings can be achieved by mining a trimmed dataset to compensate for the patch-up overhead.

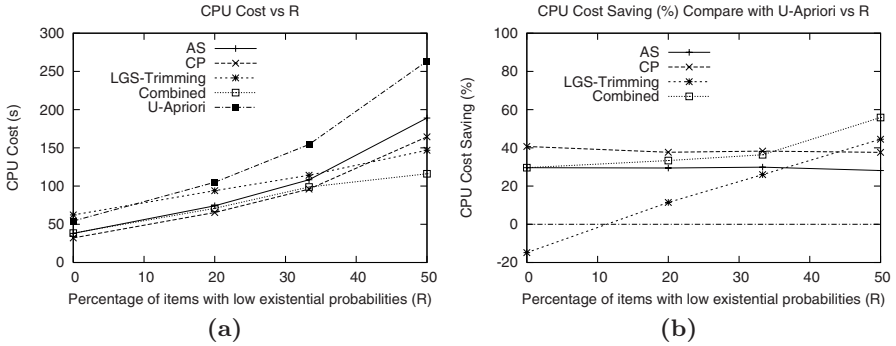


Fig. 5. CPU cost and saving with different R

In contrary, the performance of the DP methods are very stable over the range of R values. To understand this phenomenon let us consider Equation 4 for updating a AS-counter. The value of a AS-counter is determined by three terms: $S_e(x)$, $P_{t_k}(x)$ and $max_s(k)$. We note that varying R has small impact on the value of $S_e(x)$ because $S_e(x)$ is the expected support of item x , which is mainly determined by the high-probability entries of x in the dataset. Also, if transaction t_k contains a small-probability entry for x , then $P_{t_k}(x)$ is small and so the decrement to the value $d_s(x, k)$ would be insignificant. Hence, the population of small-probability items (i.e., R) has little effect in the decremental process. Finally, since $max_s(k)$ is determined by the maximum existential probability of the items (except x) in transaction t_k , low-probability items have little effect on

the value of $max_s(k)$. As a result, the performance of AS is not sensitive to the population of low-probability items. A similar conclusion can be drawn for CP by considering Equation 5.

From the figures, we also observe that the combined algorithm strikes a good balance and gives consistently good performance. It's performance is comparable to those of AS and CP when R is small, and it gives the best performance when R is large.

5 Conclusions

In this paper we proposed a decremental pruning (DP) approach for efficient mining of frequent itemsets from existential uncertain data. Experimental results showed that DP achieved significant candidate reduction and computational cost savings. Compared with LGS-Trimming, DP had the advantages of not requiring a trimming threshold and its performance was relatively stable over a wide range of low-probability-item population. In particular, it outperformed data trimming when the dataset contained few low-probability items. We argued that the Trimming approach and the DP approach were orthogonal to each other. We showed that the two approaches could be combined leading to a generally best overall performance.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of 20th ICDE, pp. 487–499. Morgan Kaufmann, San Francisco (1994)
2. Agrawal, R., Srikant, R.: Mining sequential patterns. In: Proc. of the 11th ICDE, pp. 3–14. IEEE Computer Society Press, Los Alamitos (1995)
3. Brossette, S.E., Sprague, A.P., Hardin, J.M., Jones, W.T., Moser, S.A.: Association rules and data mining in hospital infection control and public health surveillance. *Journal of the American Medical Informatics Association*, 373–381 (1998)
4. 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)
5. Zimányi, E., Pirotte, A.: Imperfect information in relational databases. In: *Uncertainty Management in Information Systems*, pp. 35–88 (1996)
6. Bayardo Jr., R.J.: Efficiently mining long patterns from databases. In: Proc. of SIGMOD 1998, pp. 85–93. ACM Press, New York (1998)

Multi-class Named Entity Recognition Via Bootstrapping with Dependency Tree-Based Patterns

Van B. Dang^{1,2} and Akiko Aizawa¹

¹ National Institute of Informatics,
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo, Japan 101-8430

² University of Natural Sciences,
227 Nguyen Van Cu St., Dist. 5, Hochiminh, Vietnam
dbvan@fit.hcmuns.edu.vn, aizawa@nii.ac.jp

Abstract. Named Entity Recognition (NER) has become a well-known problem with many important applications, such as Question Answering, Relation Extraction and Concept Retrieval. NER based on unsupervised learning via bootstrapping is gaining researchers' interest these days because it does not require manually annotating training data. Meanwhile, dependency tree-based patterns have proved to be effective in Relation Extraction. In this paper, we demonstrate that the use of dependency trees as extraction patterns, together with a bootstrapping framework, can improve the performance of the NER system and suggest a method for efficiently computing these tree patterns. Since unsupervised NER via bootstrapping uses the entities learned from each iteration as seeds for the next iterations, the quality of these seeds greatly affects the entire learning process. We introduce the technique of simultaneous bootstrapping of multiple classes, which can dramatically improve the quality of the seeds obtained at each iteration and hence increase the quality of the final learning results. Our experiments show beneficial results.

1 Introduction

Supervised learning for Named Entity Recognition (NER) has been studied thoroughly and has become the dominant technique [1, 2, 3, 4]. However, this approach requires hand-tagged training data, which is nontrivial to generate. Even more effort is needed to apply systems of this kind to a new domain of interest, since one needs to annotate a new training corpus by hand. Therefore, supervised learning for NER is used mainly in well-known domains like news and biomedical texts.

Unsupervised learning [5, 6, 7, 8, 9], by contrast, requires no manually annotated data at all. Therefore, this approach increases inter-domain portability. The common framework of this approach is known as “bootstrapping” [6, 8, 9], that is, from a list of some seed named entities, the system will discover many extraction patterns, a subset of which is selected to be the “good” set, which is then used to discover more seed entities, and so forth.

Unsupervised frameworks for NER use extraction patterns to extract entities, and many types of patterns have been proposed. Yangarber [6] used string-level representation of extraction patterns, that is, used only the surrounding text in a context window. Nevertheless, string-level patterns usually have limitations due to lack of syntactic information. Incorporating the parts of speech tends to help; the predicate–argument model [5] is of this type. A further improvement on the predicate–argument model [5] is the dependency sub-tree model proposed by Sudo et al. [10]. Since it is designed for Relation Extraction, it aims to extract relationships between entities recognized by a Named Entity (NE) recognizer. Thus, one needs an NE recognizer for this type of pattern. For example, with the sentence, “A smiling Palestinian suicide bomber triggered a massive explosion in downtown Jerusalem,”¹ firstly, the NE recognizer understands that “A smiling Palestinian suicide bomber” is a <PERSON> and “downtown Jerusalem” is a <LOCATION>. The sentence is then generalized with the NE tag in place of the actual text: “<PERSON> triggered a massive explosion in <LOCATION>.” The dependency tree-based pattern obtained from this generalized sentence captures the relation between those two entities. In this paper, we show that sub-tree patterns can be effectively integrated into the framework of bootstrapping NER, rather than using an NE tagger for the work. Since each sentence can produce a large number of sub-trees, we also propose an efficient method to compute them.

Since bootstrapping uses the entities learned from each iteration as seeds for the next ones, any false seed will mislead the next iteration of learning and might lead to even more false seeds, which degrades the performance of the entire learning process. Therefore, we introduce the technique of simultaneous bootstrapping of multiple classes, which can dramatically improve the quality of the seeds obtained at each iteration and hence increase the quality of the final learning results.

The main contributions of this paper are: (i) we apply sub-tree models to the NER task with an efficient computational method with beneficial results; (ii) we show the advantage of simultaneous bootstrapping of multiple classes to improve the quality of learning. In the following sections we focus on these points. Section 2 describes the bootstrapping framework with dependency tree-based patterns and introduces a method to efficiently compute the large number of generated patterns. Section 3 describes the advantages of bootstrapping from multiple classes. Experimental results are presented in Section 4. Section 5 contains an overview of some related works. Finally, Section 6 gives concluding remarks and describes future paths for our research.

2 Bootstrapping NER with Tree-Based Patterns

2.1 Pattern Acquisition

From the list of seed entities in each category, our system first retrieves all sentences that contain any of them. Occurrences of these seeds are then replaced

¹ This example is taken from [10].

by a generalized concept $\langle C \rangle$ denoting their category (see Table 1). In our experiment, we use four categories: APPROACH (“ $\langle \text{APPR} \rangle$ ”) such as “Maximum Entropy Models”, TASK (“ $\langle \text{TASK} \rangle$ ”) such as “Named Entity Recognition”, TOOL (“ $\langle \text{TOOL} \rangle$ ”) such as “SVMLight”, and DATASET (“ $\langle \text{COL} \rangle$ ”) such as “Wall Street Journal”. We use the Stanford Parser [16] to parse these

Table 1. The instance of the seed in the source sentence is replaced by its generalized concept to form a generalized sentence

Seed Entity	Source Sentence	Generalized Sentence
SVMs	Kernel functions allow SVMs to combine the input features at relatively low computational cost.	Kernel functions allow $\langle \text{APPR} \rangle$ to combine the input features at relatively low computational cost.

generalized sentences to obtain dependency trees. Then we apply a rightmost expansion-based algorithm for sub-tree discovery [14] to generate all sub-trees of them. Each of these sub-trees is an extraction pattern, and these patterns form the set of potential patterns Ω . Fig. 1(a) shows the dependency tree obtained from the generalized sentence shown in Table 1 and Fig. 1(b), (c) show some examples of patterns generated from this tree.

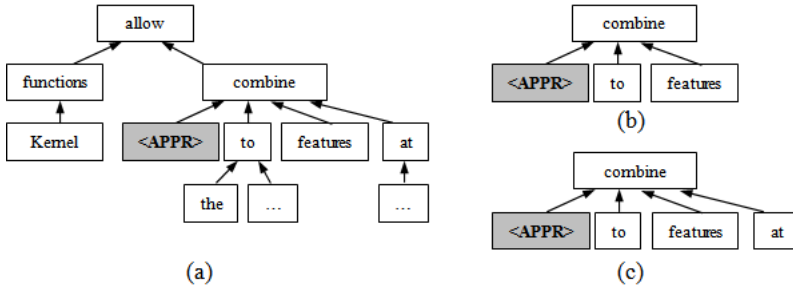


Fig. 1. Dependency trees and sub-trees obtained from the generalized sentence “Kernel functions allow $\langle \text{APPR} \rangle$ to combine the input features at relatively low computational cost.” (a) Dependency tree of the sentence. (b), (c) Some of the acquired patterns.

2.2 Pattern Matching

A tree-based pattern is said to match a target sentence if, (i) the pattern excluding the generalized node is a sub-tree of the dependency tree obtained from that sentence, and (ii) the node on the target tree corresponding to the generalized node is a noun and is called the target node. The noun group ($[\text{Adj}^* \text{Noun}^+]$) containing this target node is considered to be the extraction target. Fig. 2 shows a matching between one tree-based pattern learned from the source sentence

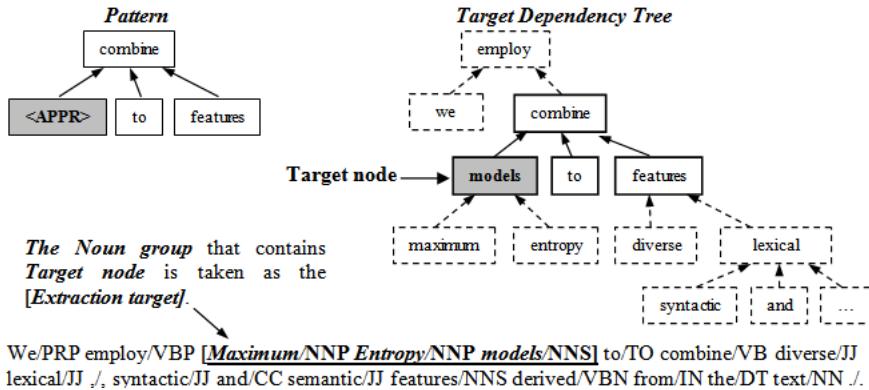


Fig. 2. Matching between a pattern and a target sentence. Since “Maximum Entropy models” is the noun group containing the target node “models”, it is considered an extracted entity.

... ” (as shown in Section 2.1 above) against the target sentence “... ”. We can see from the Fig. 2 that the pattern (on the left) is a sub-tree of the dependency tree of the target sentence (on the right), and “...” is the target node. Thus, the noun group containing “...” —the phrase “...” —is extracted as the entity.

After discovering a list of extraction patterns Ω for each category, our system starts to match all of them against all sentences in the entire corpus. If a pattern matches a sentence, the extraction target will be extracted as an entity (as “...” in Fig. 2), which is then checked and labeled as follows.

- ... : An entity of category A is labeled as “positive” when it has been in the list of seed entities for category A.
- ... : An entity of category A is labeled as “negative” when it has been in the list of seed entities for any category but A.
- ... : An entity of category A is labeled as “unknown” when it has not been in any seed entity list for any category. This is the pool where new seed entities come from.

The limitation of dependency sub-trees models [10] is the very large number of patterns that need to be computed [12]. In our work, therefore, we take only patterns of maximum length five nodes, and their root node has to be either a noun or a verb. In each iteration, we observed the discovery of on average 1000 new potential patterns for each category. We needed to match all of these 1000 patterns against 2760 sentences in the entire corpus in each iteration to see whether they could extract any entities, which could then be used to check the accuracy of the potential patterns (please refer to Section 2.3). Exhaustively matching them against thousands of sentences is very time consuming, however there are actually only a small number of sentences that match each pattern, and the unmatched ones can be rejected with only a small amount of computation.

IR-style Inverted File. We construct an IR-style inverted file for all sentences in the corpus. For each pattern to be matched, we only perform matching on sentences that contain all keywords of the pattern.

Pattern Hierarchical Structure. Since patterns are actually trees, they have a hierarchical structure. This means that if a pattern does not match a target sentence, neither does its child. Therefore, each pattern only needs to be compared with those sentences which its parent matches.

2.3 Pattern Ranking

To score the patterns in Ω , we use the scoring strategy proposed in [6]. The accuracy and confidence of the pattern p are defined as follows:

$$acc(p) = \frac{pos(p)}{pos(p) + neg(p)} \quad (1)$$

$$conf(p) = \frac{pos(p)}{pos(p) + neg(p) + unkn(p)} \quad (2)$$

where $pos(p)$ is the number of unique positive entities that p extracts, $neg(p)$ is the number of unique negative entities p extracts and $unkn(p)$ is the number of unique unknown entities p extracts. All patterns with an accuracy above a predefined threshold θ will be selected and ranked by a measure given by:

$$rank(p) = conf(p) \times \log_2 pos(p) \quad (3)$$

The top n patterns are then added to the list of accepted patterns for each category. Unknown entities extracted by all accepted patterns for a category will be considered candidate entities to be added to the list of seed entities for that category. Our next step is to select good entities from these as new seeds.

2.4 Entity Ranking

We use the entity scoring strategy proposed in [6]. A score of an entity e is then defined as:

$$score(e) = 1 - \prod_{p \in SupportPat} (1 - conf(p)) \quad (4)$$

The top m unknown entities with the highest scores extracted by accepted patterns for each category are then added to the list of seed entities for that category, and the process from 2.1 to 2.4 is iterated until no new entities are found.

3 The Advantage of Bootstrapping from Multiple Classes

The quality of seeds obtained in each iteration greatly affects the performance of the entire learning process. For example, if the learning process for finding extraction patterns for APPROACH (e.g., SVMs, Maximum Entropy Models)

mistakes entities denoting TASK (e.g., word sense disambiguation) as new seeds in iteration k , it might then wrongly accept patterns that actually extract TASK rather than APPROACH, and hence, it might mistake more TASK entities as seeds. As a result, the final set of learned patterns for APPROACH will extract entities denoting both APPROACH and TASK, which is undesirable.

By bootstrapping from multiple classes simultaneously, we have the information about seeds from more than one class. We can develop methods to exploit information about the seeds of these competing categories in order to guide the learning process to avoid mistakenly generating the wrong seeds. Our experiments show that a simple list of words created automatically—the Exception Lists—from 10 starting seeds can improve the quality of seeds obtained in each iteration and dramatically improve the final learning results.

3.1 Exception List Construction

Our bootstrapping system starts with 10 seeds for each category. The so-called Exception List is constructed also from only these 10 starting seeds, as described in Table 2 below.

Table 2. Method for constructing the Exception List

For each category i :
- Retrieve all sentences S containing any instance of the 10 seeds
- Record all words that appear right after instances of the seeds
- Measure the frequency with which they co-occur with seeds
- Discard all words with frequency less than two. The resulting words form the list L_i
For each category i :
- All words in L_i that do not appear in any other lists L_j , $j \neq i$, form the Exception List for category i

3.2 Exception List Usage

Generally, the new entities discovered in iteration k will be used as seeds for iteration $k + 1$. In the stage of Pattern Acquisition for iteration $k + 1$, for each category, we retrieve sentences containing any of its seeds to learn patterns. However, the seed for one category may actually be a seed for another due to a learning error. To prevent this from happening, for each retrieved sentence, we check whether the word appearing right after the seed instance appears in the Exception Lists of any other classes. If it does appear, this instance is not likely to be a good seed. For instance, suppose the learning process mistakes “ r_{i-1} ” as a seed for the class APPROACH. It should then know that

“ r_{i-1} ”

is not a good sentence from which to learn patterns for this class if “ r_{i-1} ” appears in the Exception List of TASK, meaning that “ r_{i-1} ” is more likely to

come after entities of TASK. It should be noted that the simple Exception Lists constructed as described above are not used to conclude that a new entity is a good seed. Instead, they are used to tell whether that entity is less likely to be a good seed for a particular class. In the example above, we do not conclude that “Maximum Entropy Models” is a good seed for TASK; instead, we say that it is less likely to be one for APPROACH.

Our experiments have shown that this simple method of exploiting information from competing categories dramatically improves the performance of the NER system. We believe that the system of bootstrapping from multiple classes has great potential, and that more powerful methods, such as statistical tools, can give even better results.

4 Experiments

4.1 Data Preparation

We conducted experiments in the domain of Computer Science papers, extracting Computer Science-specific entities. This choice was made because the aim of an unsupervised approach is to eliminate time-consuming manual effort so that the approach can be applied to domains where no tagged data is available, and Computer Science text is such a domain. Moreover, entity recognition in Computer Science texts can provide useful information for researchers. For instance, if we can extract entities denoting approaches and tasks, we can then tell which approaches have been applied to which tasks. This obviously facilitates the process of literature review.

As our first attempt, we aimed to extract entities of four classes: APPROACH, such as “Maximum Entropy Models”, TASK, such as “Named Entity Recognition”, TOOL, such as “SVMlight”, and DATASET, such as “Wall Street Journal”. For each of the four classes, we manually constructed a list of common entities and submitted them to the Yahoo! search engine through the supported search API [17]. We took the top 20 returned pdf documents for each class. We then extracted sentences containing any instances of seeds and manually tagged these instances for evaluation. This process resulted in a collection of 2760 sentences with the following statistics.

- APPROACH: 1000 sentences with 1456 instances.
- TASK: 1000 sentences with 1352 instances.
- TOOL: 380 sentences with 436 instances.
- DATASET: 380 sentences with 480 instances.

We used the Stanford Parser [16] to parse sentences to obtain dependency trees. We selected $m = n = 5$ and $\theta=0.8$ in our experiments.

4.2 Experimental Results

We evaluated our system by comparing it to the one described in [6] since it was the closest work to ours. In [6], the author used a bootstrapping framework

Table 3. String-based patterns learned from the source sentence “We employ <APPR>to combine diverse lexical, syntactic and semantic features derived from the text”

String-based Extraction Pattern			
<APPR>	*	combine	diverse
<APPR>	to	*	diverse
<APPR>	to	combine	*
...

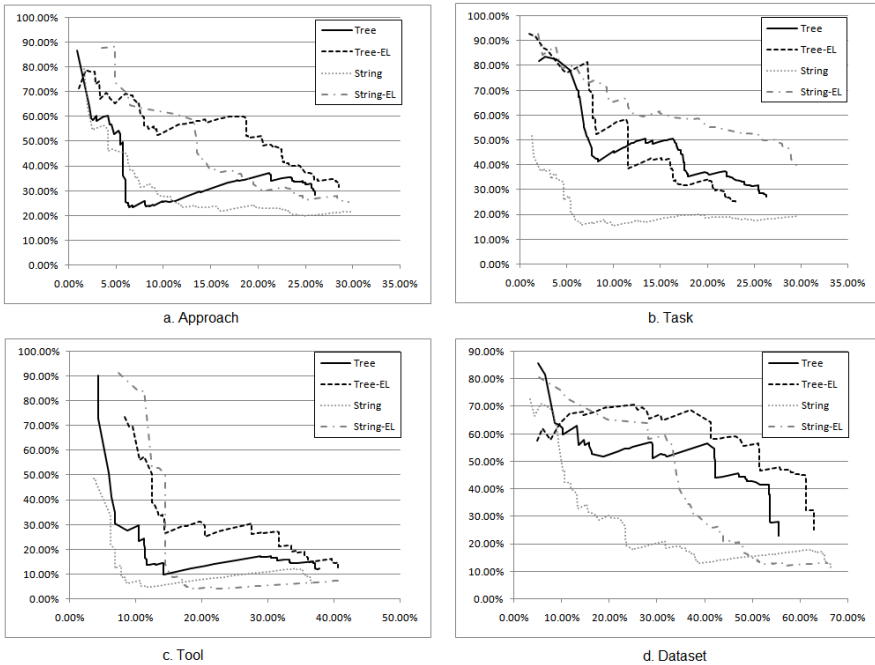


Fig. 3. Precision/Recall curves for the four categories APPROACH, TASK, TOOL and DATASET. Settings with “-EL” are those with Exception Lists.

with string-based patterns that were generated from a context window of width w around the generalized concept.

Fig. 3 shows the learning curves for all four categories. For the APPROACH, TOOL and DATASET categories, we can see that tree-based patterns outperform string-based patterns. Our analysis shows that string-based patterns can only work with sentences with minimal variation such as “ $\langle \text{APPR} \rangle$ to combine diverse lexical, syntactic and semantic features derived from the text” and “ $\langle \text{APPR} \rangle$ to combine diverse lexical, syntactic and semantic features derived from the text”. Otherwise, they fail to capture the salient contexts from the sentence. For instance, from the source sentence “ $\langle \text{APPR} \rangle$ to combine diverse lexical, syntactic and semantic features derived from the text”, some generated string-based patterns are showed in Table 3

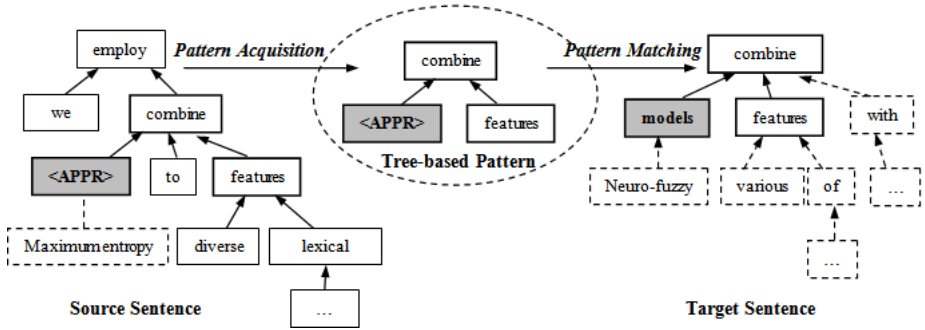


Fig. 4. Tree-based patterns can capture the salient contexts from source sentences and match them to target sentences, even though the surface texts may differ greatly

Table 4. Sentences in which our system extracts the correct entity, but where the extraction is judged to be incorrect since the extracted entity is not a name of interest

Figure 5 compares the runtime of *our algorithm* only with bisecting k-means and HFTC.

We ran the *algorithm given in Section 4.1* on the Penn Treebank.

We applied *our approach* to translation from German to English in the Europarl corpus.

Our tests with the Penn Treebank showed that *our integrated approach* achieves 92.3 % in precision and 93.2 % in recall.

below. None of them matches the target sentence “*... ..*” since it fails to capture the important context “*<... > ...*”, in which the key words are noncontiguous. Tree-based patterns, on the contrary, can deal with this very well. Fig. 4 illustrates the effectiveness of tree-based patterns. They can capture the crucial context shared between the source and target sentences, which can be very different to the surface texts.

It should be noted that the practical precision of our system is underestimated in the evaluation conducted here. With sentences such as “*... ..*”, the tree-based patterns extract “*... ..*” as an entity of APPROACH. Since we aim to extract the name of the approach, we judge this as a wrong extraction. However, “*... ..*” is actually a co-reference of a name which is mentioned somewhere. Thus, we understate the precision of our system (with both string-based and tree-based patterns). Table 4 shows examples of such sentences. The system even recognizes approaches that do not have a name, for instance, “*... ..*”, even though “*... ..*” (which is not always named) is a combination of many modules or techniques. We are not tackling this problem now: since we want to study the effectiveness of tree-based patterns and a bootstrapping

framework for NER, we have not applied co-reference resolution to our work, though we believe it would improve precision.

Fig. 3 also shows the effectiveness of the Exception List. Either with string-based or tree-based patterns, systems with an Exception List outperform systems without one most of the time, which means the simple Exception List can help prevent the system from generating the wrong entities as new seeds during learning. This indicates the potential of simultaneously bootstrapping from multiple classes. We believe that employing more complicated methods can further improve the quality of learning.

5 Related Work

Riloff [11] employed a weakly supervised method to the problem of NER. The author only requires the training corpus to be labeled as relevant and irrelevant rather than fully tagged. From a set of handcrafted rule templates, their system learns extraction patterns and selects those that occur most frequently in the relevant corpus. The set of patterns learned is then filtered manually. This approach greatly reduces human intervention, but human labor is still required to judge the training data as relevant or not and to compose rule templates.

To the best of our knowledge, the first work using bootstrapping for Information Extraction is DIPRE [8]. From a handful of examples of (book title, author) relations, their system searches the web for their instances, and extracts patterns that are then used to extract new instances of (book title, author). Their system only uses simple token-based patterns and simple methods to select good patterns—just the length of the pattern and the number of times it occurs. Snowball [9] improved DIPRE by only extracting relationships between entities recognized by a named entity tagger. However, their techniques were proposed mainly for Relation Extraction.

The prior work that is closest to ours is the one proposed in [6], which describes the unsupervised learning of disease names and locations via bootstrapping. Since their system uses only string-based patterns, it has limitations because of the variation of text. They also took advantage of competing categories to select more distinctive patterns, but they did not address the problem of how picking bad entities as seeds can mislead the entire learning process.

Sudo et al. [10] proposed dependency tree-based patterns for Relation Extraction, and they have been used very successfully in an On-Demand Information Extraction system [13]. Its powerful representation ability for Relation Extraction has also been confirmed by Stevenson et al. [12]. However, since it is designed for Relation Extraction, it requires a named entity tagger to specify the boundary of entities for which the relationship is to be extracted. Therefore, we modify it for NER.

Etzioni et al. [15] proposed an unsupervised method to extract named entities from the web. This interesting scheme uses web statistics to improve the accuracy of extraction. However, since their system is targeted more at extracting entities

than learning patterns, it is not related to our work. Nevertheless, the idea is inspiring, and we can incorporate it into the present framework in the future.

6 Conclusions and Future Work

Dependency tree-based extraction patterns have powerful representation abilities [10, 12, 13]. They were originally used to identify relationships between entities extracted by a Named Entity recognizer. In this paper, we have adapted them to the task of NER—rather than using a Named Entity Recognizer for this task—via a bootstrapping framework, and shown that this is also very effective. We also proposed an efficient method for handling the large number of tree-based patterns. Finally, we introduced a novel scheme using bootstrapping from multiple classes to improve the quality of the seeds obtained in each iteration, improving the final learning results.

Our system can be improved in many aspects. We have implemented a very simple technique for taking advantage of multi-class bootstrapping—only a list of words that co-occur with entities of interest more than twice. We believe that by employing statistical methods such as co-occurrence statistics, we can further improve the results. Moreover, since we take the noun group containing the extraction target given by patterns as an entity, some seeds obtained during learning are not “clean”, for example, “traditional speech recognition” instead of “speech recognition”. The system will miss sentences in which “speech recognition” occurs on its own, which are even more frequent. We have to implement techniques to remove these “noisy” words to improve the learning quality.

In this paper, we only work with the four fixed classes in which we are interested. We believe that the selection of classes for learning will affect the final learning results, and we will investigate this problem in the future. We also believe the integration of co-reference resolution can help the system extract more precise entities, rather than only their co-references.

References

- [1] Bikel, D.M., Miller, S., Schwartz, R., Weischedel, R.M.: Nymble: A high-performance learning name-finder. In: Proceedings of the 5th Conference on Applied Natural Language Processing, pp. 194–201 (1997)
- [2] Borthwick, A., Sterling, J., Agichtein, E., Grishman, R.: Exploiting Diverse Knowledge Sources via Maximum Entropy in Named Entity Recognition. In: Proceedings of the 6th Workshop on Very Large Corpora, pp. 152–160 (1998)
- [3] Collins, M.: Ranking Algorithms for Named-Entity Extraction: Boosting and the Voted Perceptron. In: Proceedings of the Annual Meeting of the Association for Computation Linguistics, pp. 489–496 (2002)
- [4] McCallum, A., Li, W.: Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web-Enhanced Lexicons. In: The 7th Conference on Natural Language Learning (CoNLL), pp. 188–191 (2003)
- [5] Yangarber, R., Grishman, R., Tapanainen, P., Huttunen, S.: Unsupervised Discovery of Scenario-Level Patterns for Information Extraction. In: Proceedings of Conference on Applied Natural Language Processing, pp. 282–289 (2000)

- [6] Yangarber, R., Lin, W., Grishman, R.: Unsupervised learning of generalized names. In: Proceedings of the 19th International Conference on Computational Linguistics, pp. 1–7 (2002)
- [7] Collins, M., Singer, Y.: Unsupervised Models for Named Entity Classification. In: Proceedings of Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora (1999)
- [8] Brin, S.: Extracting Patterns and Relations from the World Wide Web. In: Proceedings of the International Workshop on the Web and Databases, pp. 172–183 (1998)
- [9] Agichtein, E., Gravano, L.: Snowball: Extracting Relations from Large Plain-Text Collections. In: The 5th ACM International Conference on Digital Libraries, pp. 85–94 (2000)
- [10] Sudo, K., Sekine, S., Grishman, R.: An Improved Extraction Pattern Representation Model for Automatic IE Pattern Acquisition. In: Proceedings of the 41st Annual Meeting of Association of Computational Linguistics, pp. 224–231 (2003)
- [11] Riloff, E.: Automatically Generating Extraction Patterns from Untagged Text. In: Proceedings of the 13th National Conference on Artificial Intelligence, pp. 1044–1049 (1996)
- [12] Stevenson, M., Greenwood, M.A.: Comparing Information Extraction Pattern Models. In: Proceedings of the Workshop on Information Extraction Beyond The Document, pp. 12–19 (2006)
- [13] Sekine, S.: On-Demand Information Extraction. In: Proceedings of the International Conference on Computational Linguistics and the Association for Computational Linguistics, pp. 17–21 (2006)
- [14] Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., Arikawa, S.: Efficient Substructure Discovery from Large Semi-structured Data. IEICE Transactions on Information and Systems E87-D(12), 2754–2763 (2004)
- [15] Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A., Shaked, T., Soderland, S., Weld, D., Yates, A.: Unsupervised named-entity extraction from the web: An experimental study. *Artificial Intelligence* 165(1), 91–134 (2005)
- [16] <http://nlp.stanford.edu/software/lex-parser.shtml>
- [17] <http://developer.yahoo.com/download/>

Towards Region Discovery in Spatial Datasets

Wei Ding^{1,*}, Rachsuda Jiamthapthaksin¹, Rachana Parmar¹, Dan Jiang¹,
Tomasz F. Stepinski², and Christoph F. Eick¹

¹ University of Houston, Houston TX 77204-3010, USA
{wding,rachsuda,rparmar,djiang,ceick}@uh.edu

² Lunar and Planetary Institute, Houston, TX 77058, USA
tstepinski@lpi.usra.edu

Abstract. This paper presents a novel region discovery framework geared towards finding scientifically interesting places in spatial datasets. We view region discovery as a clustering problem in which an externally given fitness function has to be maximized. The framework adapts four representative clustering algorithms, exemplifying prototype-based, grid-based, density-based, and agglomerative clustering algorithms, and then we systematically evaluated the four algorithms in a real-world case study. The task is to find feature-based hotspots where extreme densities of deep ice and shallow ice co-locate on Mars. The results reveal that the density-based algorithm outperforms other algorithms inasmuch as it discovers more regions with higher interestingness, the grid-based algorithm can provide acceptable solutions quickly, while the agglomerative clustering algorithm performs best to identify larger regions of arbitrary shape. Moreover, the results indicate that there are only a few regions on Mars where shallow and deep ground ice co-locate, suggesting that they have been deposited at different geological times.

Keywords: Region Discovery, Clustering, Hotspot Discovery, Spatial Data Mining.

1 Introduction

The goal of spatial data mining [1,2,3] is to automate the extraction of interesting and useful patterns that are not explicitly represented in spatial datasets. Of particular interests to scientists are the techniques capable of finding scientifically meaningful regions as they have many immediate applications in geoscience, medical science, and social science; e.g., detection of earthquake hotspots, disease zones, and criminal locations. An ultimate goal for region discovery is to provide search-engine-style capabilities to scientists in a highly automated fashion. Developing such a system faces the following challenges. First, the system must be able to find regions of arbitrary shape at different levels of resolution. Second, the system needs to provide suitable, plug-in measures of interestingness to instruct discovery algorithms what they should seek for. Third, the identified regions should be properly ranked by relevance.

* Also, Computer Science Department, University of Houston-Clear Lake.

Fourth, the system must be able to accommodate discrepancies in various formats of spatial datasets. In particular, the discrepancy between continuous and discrete datasets poses a challenge, because existing data mining techniques are not designed to operate on a mixture of continuous and discrete datasets. Fifth, it is desirable for the framework to provide pruning and other sophisticated search strategies as the goal is to seek for interesting, highly ranked regions.

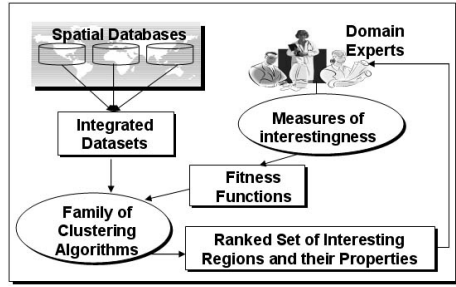


Fig. 1. Region discovery framework

This paper presents a novel region discovery framework (see Fig. 1) geared towards finding scientifically interesting places in spatial datasets. We view region discovery as a clustering problem in which an externally given fitness function has to be maximized. The framework adapts four representative clustering algorithms, exemplifying prototype-based, grid-based, density-based, and agglomerative clustering algorithms for the task of region discovery. The fitness function combines contributions of interestingness from constituent clusters and can be customized by domain experts. The framework allows for plug-in fitness functions to support a variety of region discovery applications correspondent to different domain interests.

Relevant Work. Many studies have been conducted in region discovery. These most relevant to our work are region-oriented clustering techniques and hotspot discovery. In our previous work, we have discussed a region discovery method that was restricted to one categorical attribute [4,5]. The integrated framework introduced in this paper is generalized to be applicable to both continuous and discrete datasets. The framework allows for various plug-in fitness functions and extends our work to the field of feature-based hotspot discovery (see Section 2). [1] introduces a “region oriented” clustering algorithm to select regions to satisfy certain condition such as density. This approach uses statistical information instead of a fitness function to evaluate a cluster.

Hotspots are object clusters with respect to spatial coordinates. Detection of hotspots using variable resolution approach [6] was investigated in order to minimize the effects of spatial superposition. In [7] a region growing method for hotspot discovery was described, which selects seed points first and then grows clusters from these seed points by adding neighbor points as long as a density threshold condition is satisfied. Definition of hotspots was extended in [8] using circular zones for multiple variables.

Contributions. This paper presents a highly generic framework for region discovery in spatial datasets. We customize our discovery framework to accommodate raster, continuous, and categorical datasets. This involves finding a suitable object structure, suitable preprocessing techniques, a family of reward-based fitness functions for various measures of interestingness, and a collection of

clustering algorithms. We systematically evaluate a wide range of representative clustering algorithms to determine when and which type of clustering techniques are more suitable for region discovery. We apply our framework to a real-world case study concerning ground ice on Mars and successfully find scientifically interesting places.

2 Methodology

Region Discovery Framework. Our region discovery method employs a reward-based evaluation scheme that evaluates the quality of the discovered regions. Given a set of regions $R = \{r_1, \dots, r_k\}$ identified from a spatial dataset $O = \{o_1, \dots, o_n\}$, the fitness of R is defined as the sum of the rewards obtained from each region r_j ($j = 1 \dots k$):

$$q(R) = \sum_{j=1}^k (i(r_j) \times size(r_j)^\beta) \quad (1)$$

where $i(r_j)$ is the interestingness measure of region r_j – a quantity based on domain interest to reflect the degree to which the region is “newsworthy”. The framework seeks for a set of regions R such that the sum of rewards over all of its constituent regions is maximized. $size(r_j)^\beta$ ($\beta > 1$) in $q(R)$ increases the value of the fitness nonlinearly with respect to the number of objects in O belonging to the region r_j . A region reward is proportional to its interestingness, but given two regions with the same value of interestingness, a larger region receives a higher reward to reflect a preference given to larger regions.

We employ clustering algorithms for region discovery. A region is a contiguous subspace that contains a set of spatial objects: for each pair of objects belonging to the same region, there always exists a path within this region that connects them. We search for regions r_1, \dots, r_k such that:

1. $r_i \cap r_j = \emptyset, i \neq j$. The regions are disjoint.
2. $R = \{r_1, \dots, r_k\}$ maximizes $q(R)$.
3. $r_1 \cup \dots \cup r_k \subseteq O$. The generated regions are not required to be exhaustive with respect to the global dataset O .
4. r_1, \dots, r_k are ranked based on their reward values. Regions that receive no reward are discarded as outliers.

Preprocessing. Preprocessing techniques are introduced to facilitate the application of the framework to heterogeneous datasets. Given a collection of raster, categorical, and continuous datasets with a common spatial extent, the raster datasets are represented as ($\langle \text{pixel} \rangle, \langle \text{continuous variables} \rangle$), the categorical dataset as ($\langle \text{point} \rangle, \langle \text{category variables} \rangle$)¹, and the continuous datasets as ($\langle \text{point} \rangle, \langle \text{continuous variables} \rangle$). Fig. 2 depicts our preprocessing procedure:

¹ To deal with multiple categorical datasets a single dataset can be constructed by taking the union of multiple categorical datasets.

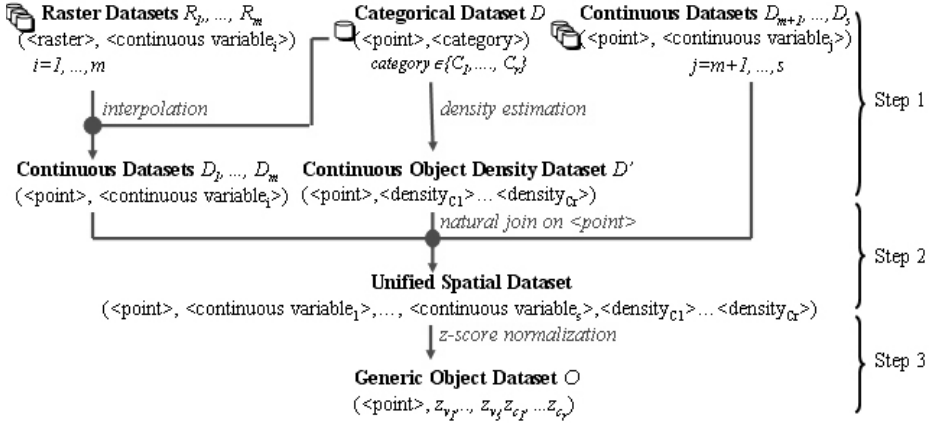


Fig. 2. Preprocessing for heterogeneous spatial datasets

- Step 1. Dataset Integration.** Categorical datasets are converted into a continuous density dataset ($\langle \text{point} \rangle, \langle \text{density variables} \rangle$), where a density variable describes the density of a class for a given point. Classical density estimation techniques [9], such as Gaussian kernel functions, can be used for such transformation. Raster datasets are mapped into point datasets using interpolation functions that compute point values based on the raster values.
- Step 2. Dataset Unification.** A single unified spatial dataset is created by taking a natural join on the spatial attributes of each dataset. Notice that the datasets have to be made “join compatible” in Step 1. This can be accomplished by using the same set of points in each individual dataset.
- Step 3. Dataset Normalization.** Finally, continuous variables are normalized into z-scores to produce a generic dataset $O = (\langle \text{point} \rangle, \langle \text{z-scores} \rangle)$, where z-score is the number of standard deviations that a given value is above or below the mean.

Measure of Interestingness. The fitness function $q(R)$ (Eqn. (1)) allows a function of interestingness to be defined based on different domain interests. In our previous work, we have defined fitness functions to search risk zones of earthquakes [4] and volcanoes [5] with respect to a single categorical attribute. In this paper, we define $\mathcal{R}_1, \dots, \mathcal{R}_q$ as localized regions where continuous non-spatial features of objects attain τ_1, \dots, τ_q the values from the wings of their respective distributions. Hence our feature-based hotspots are places where multiple, potentially globally uncorrelated attributes happen to attain extreme values. We then introduce a new interestingness function i on the top of the generic dataset O : given set of continuous features $A = \{A_1, \dots, A_q\}$ the interestingness of an object $o \in O$ is measured as follows:

$$i(A, o) = \prod_{j=1}^q z_{A_j}(o) \quad (2)$$

where $z_{A_j}(o)$ is the z-score of the continuous feature A_j . Objects with $|i(A, o)| \gg 0$ are clustered in feature-based hotspots where the features in A happen to attain extreme values—measured as products of z-scores.

We then extend the definition of interestingness to regions: the interestingness of a region r is the absolute value of the average interestingness of the objects belonging to it:

$$i(A, r) = \begin{cases} \left(\frac{|\sum_{o \in r} i(A, o)|}{\text{size}(r)} - z_{\text{th}} \right) & \text{if } \frac{|\sum_{o \in r} i(A, o)|}{\text{size}(r)} > z_{\text{th}} \\ 0 & \text{otherwise.} \end{cases} \quad (3)$$

In Eqn. 3 the interestingness threshold z_{th} is introduced to weed out regions with $i(r)$ close to 0, which prevents clustering solutions from containing only large clusters of low interestingness.

Clustering Algorithms. Our regional discovery framework relies on reward-based fitness functions. Consequently, clustering algorithms embedded in the framework, have to allow for plug-in fitness functions. However, the use of fitness function is quite uncommon in clustering, although a few exceptions exist, e.g., CHAMELEON [10]. Furthermore, region discovery is different from traditional clustering as it gears to find interesting places with respect to a given measure of interestingness. Consequently, existing clustering techniques need to be modified extensively for the task of region discovery. The proposed region discovery framework adapts a family of prototype-based, agglomerative, density-based, and grid-based clustering approaches. We give a brief survey of these algorithms in this section.

Prototype-based Clustering Algorithms. Prototype-based clustering algorithms first seek for a set of representatives; clusters are then created by assigning objects in the dataset to the closest representatives. We introduce a modification of the PAM algorithm [11] which we call SPAM (Supervised PAM). SPAM starts its search with a random set of k representatives, and then greedily replaces representatives with non-representatives as long as $q(R)$ improves. SPAM requires the number of clusters, k , as an input parameter. Fig. 3a illustrates the application of SPAM to a supervised clustering task in which purity of clusters with respect to the instances of two classes has to be maximized. SPAM correctly separates cluster A from cluster B because the fitness value would be decreased if the two clusters were merged, while the traditional PAM algorithm will merge the two clusters because they are in close proximity.

Agglomerative Algorithms. Due to the fact that prototype-based algorithms construct clusters using nearest neighbor queries, the shape of clusters identified are limited to convex polygons (Voronoi cells). Interesting regions, and in particular, hotspots, may not be restricted to convex shapes. Agglomerative clustering algorithms are capable of yielding solutions with clusters of arbitrary shape by constructing unions of small convex polygons. We adapt the MOSAIC algorithm [5] that takes a set of small convex clusters as its input and greedily merges neighboring clusters as long as $q(R)$ improves. In our experiments

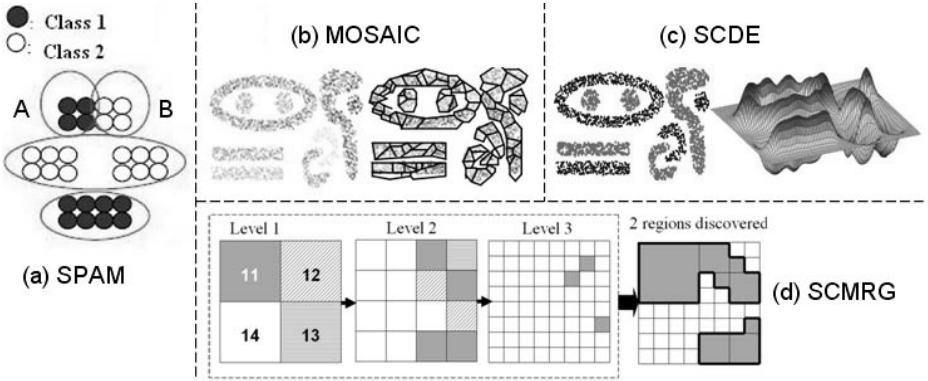


Fig. 3. Clustering algorithms

the inputs are generated by the SPAM algorithm. Gabriel graphs [12] are used to determine which clusters are neighbors. The number of clusters, k , is then implicitly determined by the clustering algorithm itself. Fig. 3b illustrates that MOSAIC identifies 9 clusters (4 of them are in non-convex shape) from the 95 small convex clusters generated by SPAM.

Density-Based Algorithms. Density-based algorithms construct clusters from an overall \mathcal{O} . We adapt the SCDE (Supervised Clustering Using Density Estimation) algorithm [13] to search feature-based hotspots. Each object o in \mathcal{O} is assigned a value of $i(A, o)$ (see Eqn. 2). The influence function of object o , $f_{Gauss}(p, o)$, is defined as the product of $i(A, o)$ and a Gaussian kernel:

$$f_{Gauss}(p, o) = i(A, o) \times e^{-\frac{d(p, o)^2}{2\sigma^2}}. \quad (4)$$

The parameter σ determines how quickly the influence of o on p decreases as the distance between o and p increases. The density function, $\Psi(p)$ at point p is then computed as:

$$\Psi(p) = \sum_{o \in \mathcal{O}} f_{Gauss}(p, o). \quad (5)$$

Unlike traditional density estimation techniques, which only consider the spatial distance between data points, our density estimation approach additionally considers the influence of the interestingness $i(A, o)$. SCDE uses a hill climbing approach to compute local maxima and local minima of the density function Ψ . These locales act as cluster attractors; clusters are formed by associating objects in \mathcal{O} with the attractors. The number of clusters, k , is implicitly determined by the parameter σ . Fig. 3c illustrates an example in which SCDE identifies 9 regions that are associated with maxima (in red) and minima (in blue) of the depicted density function on the right.

Grid-based Algorithms. SCMRG (Supervised Clustering using Multi-Resolution Grids) [4] is a hierarchical, grid-based method that utilizes a divisive, top-down search. The spatial space of the dataset is partitioned into grid

cells. Each grid cell at a higher level is partitioned further into smaller cells at the lower level, and this process continues as long as the sum of the rewards of the lower level cells $q(R)$ is not decreased. The regions returned by SCMRG are combination of grid cells obtained at different level of resolution. The number of clusters, k , is calculated by the algorithm itself. Fig. 3d illustrates that SCMRG drills down 3 levels and identifies 2 clusters (the rest of cells are discarded as outliers due to low interestingness).

3 A Real-World Case Study: Ground Ice on Mars

Dataset Description and Preprocessing. We systematically evaluate our region discovery framework on spatial distribution of ground ice on Mars. Mars is at the center of the solar system exploration efforts. Finding scientifically interesting places where shallow and deep ice abundances coincide provides important insight into the history of water on Mars. z_{di} , located in the shallow subsurface of Mars, within an upper 1 meter, is obtained remotely from orbit by the gamma-ray spectrometer [14] (see Fig. 4a, shallow ice in $5^\circ \times 5^\circ$ resolution). A spatial distribution of z_{si} , up to the depth of a few kilometers, can be inferred from spatial distribution of rampart craters [15] (see Fig. 4b, distribution of 7559 rampart craters restricted to the spatial extent defined by the shallow ice raster). Rampart craters, which constitute about 20% of all the 35927 craters on Mars, are surrounded by ejecta that have patterns like splashes and are thought to form in locations once rich in subsurface ice. Locally-defined relative abundance of rampart craters can be considered a proxy for the abundance of deep ice.

Using the preprocessing procedure outlined in Section 2 we construct a generic dataset $(\langle \text{longitude, latitude} \rangle, z_{di}, z_{si})$ where $\langle \text{longitude, latitude} \rangle$ is the coordinate of each rampart crater, z_{di} denotes the z-score of deep ice and z_{si} denotes the z-score of shallow ice. The values of these two features at location p are computed using a $5^\circ \times 5^\circ$ moving window wrapped around p . The shallow ice feature is an average of shallow-ice abundances as measured at locations of objects within the window, and the deep-ice feature is a ratio of rampart to all the craters located within the window.

Region Discovery Results. SPAM, MOSAIC, SCDE, and SCMRG clustering algorithms are used to find feature-based hotspots where extreme values of deep ice and shallow ice co-locate on Mars. The algorithms have been developed in our open source project *Cougar*² [16]. In the experiments, the clustering algorithms maximize the following fitness function $q(R)$ — see also Eqn 6

$$q(R) = \sum_{r \in R} (i(\{z_{di}, z_{si}\}, r) \times \text{size}(r)^\beta) \quad (6)$$

For the purpose of simplification, we will use z for $i(\{z_{di}, z_{si}\}, r)$ in the rest of the paper. In the experiments, the interestingness threshold is set to be $z_{th} = 0.15$

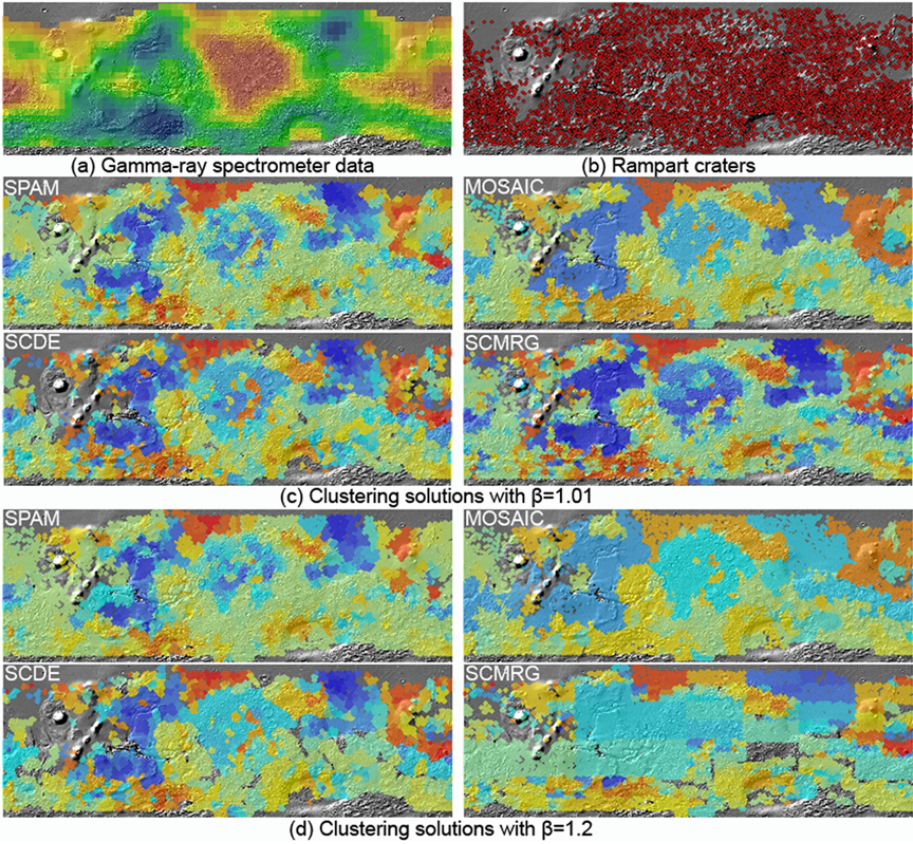


Fig. 4. Grayscale background depicts elevation of the Martian surface between longitude of -180° to 180° and latitude -60° to 60° . Legend indicates z value for each cluster. Objects not belonging to regions identified are not shown (better viewed in color).

and two different β values are used: $\beta = 1.01$ is used for finding stronger hotspots characterized by higher values of z even if the sizes are small, and $\beta = 1.2$ for identifying larger but likely weaker hotspots. Table 1 summarizes the experimental results. Fig. 4c shows the correspondent clustering results using $\beta = 1.01$. And Fig. 4d demonstrates that larger (but weaker) hotspots are identified for $\beta = 1.2$. Objects (craters) are color-coded according to the z values of clusters to which they belong. The hotspots are in the locations where objects are coded by either deep red or deep blue colors. In the red-coded hotspots the two variables have values from the same-side wings of their distributions (high-high or low-low). In the blue-coded hotspots the two variables have values from the opposite-side wings of their distributions (high-low or low-high).

Which clustering algorithm produces the best region discovery results? In the rest of section, we evaluate the four clustering algorithms with respect to

Table 1. Parameters of clustering algorithms and statistical analysis

	SPAM	SCMRG	SCDE	MOSAIC
$\beta = 1.01/\beta = 1.2$				
Parameters	$k = 2000/k = 807$	None	$\sigma = 0.1/\sigma = 1.2$	None
$q(R)$	13502/24265	14129 / 34614	14709/39935	14047/59006
# of clusters	2000/807	1597/644	1155/613	258/152
Statistics of Number of Objects Per Region				
Max	93/162	523/2685	1258/3806	4155/5542
Mean	18/45	15/45	25/49	139/236
Std	10/25	31/201	80/193	399/717
Skewness	1.38/1.06	9.52/10.16	9.1/13.44	6.0/5.24
Statistics of Rewards Per Region				
Max	197/705	743/6380	671/9488	3126/16461
Mean	10/46	9/54	12/65	94/694
Std	15/66	35/326	38/415	373/2661
Skewness	5.11/4.02	13.8/13.95	10.1/19.59	6.24/4.69
Statistics of \sqrt{z} Per Region				
Max	2.7/2.45	2.85/2.31	2.95/2.94	1.24/1.01
Mean	0.6/0.57	0.74/0.68	0.95/0.97	0.44/0.40
Std	0.38/0.36	0.31/0.26	0.47/0.47	0.24/0.22
Skewness	1.14/1.34	1.58/1.88	1.28/1.31	0.73/0.40

statistical measures, algorithmic consideration, shape analysis, and scientific contributions.

Statistical Measures. Table 1 is divided into four sections. The first section reports on the overall properties of clustering solutions: the parameters used by the clustering algorithms, the total reward and the number of regions discovered. The remaining three sections report on statistics of three different properties: region size, its reward on the population of the constituent regions, and \sqrt{z} , the square root of the interestingness of regions. The SPAM algorithm requires an input parameter k , which is chosen to be a value that is of the same order of magnitude as the values of k yielded by the SCMRG and SCDE algorithms. Due to its agglomerative character the MOSAIC algorithm always produces a significantly smaller number of clusters regardless of the size of its input provided by the SPAM clustering solution. Thus the MOSAIC is separated from the other solutions in the table.

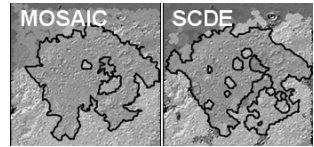
To seek for feature-based hotspots of shallow ice and deep ice, the solution that receives high value of $q(R)$ and provides more clusters with the highest values of \sqrt{z} is the most suitable. This is the solution having a large value of skewness for the reward and \sqrt{z} properties. Skewness measures the asymmetry of the probability distribution, as the large value of skewness indicates existence of hotspots (more extreme values). In addition a suitable solution has larger values of the mean and the standard deviation for the reward and \sqrt{z} properties, as they indicate existence of stronger hotspots. The analysis of Table 1 indicates that SCDE and SCMRG algorithms are more suitable to discovery hotspots with higher

values in z . Furthermore, we are interested in evaluating the search capability, how the top n regions are selected by the four algorithms. Fig. 5a illustrates the average region size with respect to the top 99th, 97th, 94th, 90th, 80th, 60th percentile for the value of interestingness z . Fig. 5b depicts the average value of interestingness per cluster with respect to the top 10 largest regions. We observe that SCDE can pinpoint stronger hotspots in smaller size (e.g., $size = 4$ and $z = 5.95$), while MOSAIC is the better algorithm for larger hotspots with relatively higher value of interestingness (e.g., $size = 2096$ and $z = 1.38$).

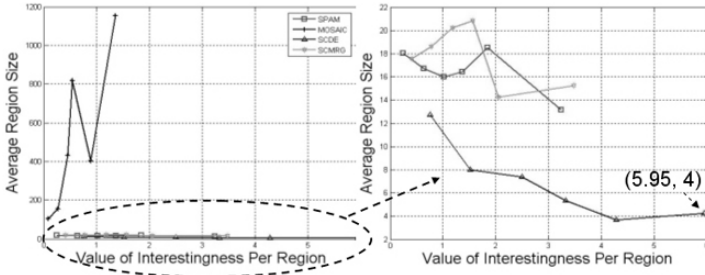
Algorithmic Considerations. As determined by the nature of the algorithm, SCDE and SCMRG algorithms support the notion of outliers – both algorithms evaluate and prune low-interest regions (outliers) dynamically during the search procedure. Outliers create an overhead for MOSAIC and SPAM because both algorithms are forced to create clusters to separate non-reward regions (outliers) from reward regions. Assigning outliers to a reward region in proximity is not an alternative because this would lead to a significant drop in the interestingness value and therefore to a significant drop in total rewards.

The computer used in our experiments is Intel(R) Xeon, CPU 3.2GHz, 1GB of RAM. In the experiments of $\beta = 1.01$ the SCDE algorithm takes $\sim 500s$ to complete, whereas the SCMRG takes $\sim 3.5s$, the SPAM takes $\sim 50000s$, and the MOSAIC took $\sim 155000s$. Thus, the SCMRG algorithm is significantly faster than the other clustering algorithms and, on this basis, it could be a suitable candidate to searching for hotspots in a very large dataset with limited time.

Shape Analysis. As depicted in Fig. 4, in contrast to SPAM whose shapes are limited to convex polygons, and SCMRG whose shapes are limited to unions of grid-cells, MOSAIC and SCDE can find arbitrary-shaped clusters. The SCMRG algorithm only produces good solutions for small values of β , as larger values of β lead to the formation of large, boxy segments that are not effective in isolating the hotspots. In addition, the figure on the right depicts the area of Acidalia Plantia on Mars (centered at $\sim -15^\circ$ longitude, -40° latitude). MOSAIC and SCDE have done a good job in finding non-convex shape clusters. Moreover, notice that both algorithms can discover interesting regions inside other regions – red-coded regions (high-high or low-low) are successfully identified inside the blue-coded regions (low-high or high-low). It thus makes the hotspots even “hotter” when excluding inside regions from an outside region.



Scientific Contributions. Although the global correlation between the shallow ice and deep ice variables is only -0.14434 — suggesting the absence of a global linear relationship — our region discovery framework has found a number of local regions where extreme values of both variables co-locate. Our results indicate that there are several regions on Mars that show a strong anti-collocation between shallow and deep ice (in blue), but there are only few regions on Mars where shallow and deep ground ice co-locate (in red). This suggests that shallow ice and deep ice have been deposited at different geological times on Mars. These



(a) 99th to 60th percentile sorted by the value of z

	SPAM	SCMRG	SCDE	MOSAIC				
1	0.56	93	1.34	523	0.50	1258	0.57	4155
2	0.30	77	0.94	422	0.22	1145	0.00	2139
3	2.52	73	1.66	407	0.58	903	1.38	2096
4	0.18	72	0.70	367	0.53	784	0.00	2056
5	0.54	71	0.60	335	0.71	656	0.00	1491
6	0.66	70	0.98	315	0.21	571	0.47	1452
7	1.00	61	0.39	313	0.05	563	0.31	1174
8	1.81	59	0.62	282	0.22	463	0.45	1172
9	0.81	58	0.15	277	0.05	447	1.36	1143
10	0.04	57	0.30	262	0.16	435	0.64	1093

outliers

(b) z values of the top 10 regions sorted by region size

Fig. 5. Search capability evaluation

places need to be further studied by the domain experts to find what particular set of geological circumstances led to their existence.

4 Conclusion

This paper presents a novel region discovery framework for identifying the feature-based hotspots in spatial datasets. We have evaluated the framework with a real-world case study of spatial distribution of ground ice on Mar. Empirical statistical evaluation was developed to compare the different clustering solutions for their effectiveness in locating hotspots. The results reveal that the density-based SCDE algorithm outperforms other algorithms inasmuch as it discovers more regions with higher interestingness, the grid-based SCMRG algorithm can provide acceptable solutions within limited time, while the agglomerative MOSAIC clustering algorithm performs best on larger hotspots of arbitrary shape. Furthermore, our region discovery algorithms have identified several interesting places on Mars that will be further studied in the application domain.

Acknowledgments

The work is supported in part by the National Science Foundation under Grant IIS-0430208. A portion of this research was conducted at the Lunar and Planetary Institute, which is operated by the USRA under contract CAN-NCC5-679 with NASA.

References

1. Wang, W., Yang, J., Muntz, R.R.: STING: A statistical information grid approach to spatial data mining. In: 23rd Intl. Conf. on Very Large Data Bases (1997)
2. Koperski, K., Han, J.: Discovery of spatial association rules in geographic information databases. In: Egenhofer, M.J., Herring, J.R. (eds.) *Procs. of the 4th Intl. Symp. Advances in Spatial Databases*, vol. 951, 6–9, pp. 47–66 (1995)
3. Shekhar, S., Huang, Y.: Discovering spatial co-location patterns: A summary of results. In: Jensen, C.S., Schneider, M., Seeger, B., Tsotras, V.J. (eds.) *SSTD 2001. LNCS*, vol. 2121, Springer, Heidelberg (2001)
4. Eick, C.F., Vaezian, B., Jiang, D., Wang, J.: Discovering of interesting regions in spatial data sets using supervised clustering. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) *PKDD 2006. LNCS (LNAI)*, vol. 4213, Springer, Heidelberg (2006)
5. Choo, J., Jiamthaphaksin, R., Sheng Chen, C., Celepcikay, O.U., Giusti, C., Eick, C.F.: MOSAIC: A proximity graph approach for agglomerative clustering. In: *The 9th Intl. Conf. on Data Warehousing and Knowledge Discovery* (2007)
6. Brimicombe, A.J.: Cluster detection in point event data having tendency towards spatially repetitive events. In: *The 8th Intl. Conf. on GeoComputation* (2005)
7. Tay, S.C., Hsu, W., Lim, K.H.: Spatial data mining: Clustering of hot spots and pattern recognition. In: *The Intl. Geoscience & Remote Sensing Symposium* (2003)
8. Kulldorff, M.: Prospective time periodic geographical disease surveillance using a scan statistic. *Journal Of The Royal Statistical Society Series A* 164, 61–72 (2001)
9. Silverman, B.: *Density Estimation for Statistics and Data Analysis*. Chapman and Hall, Boca Raton (1986)
10. Karypis, G., Han, E.H.S., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. *IEEE Computer* 32(8), 68–75 (1999)
11. Kaufman, L., Rousseeuw, P.J.: *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Chichester (1990)
12. Gabriel, K.R., Sokal, R.R.: A new statistical approach to geographic variation analysis. *Systematic Zoology* 18, 259–278 (1969)
13. Jiang, D., Eick, C.F., Chen, C.: On supervised density estimation techniques and their application to clustering. In: *Procs. of the 15th ACM Intl. Symposium on Advances in Geographic Information Systems* (2007)
14. Feldman, W.C.: Global distribution of near-surface hydrogen on mars. *J. Geophys. Res.* 109, E09006 (2004)
15. Barlow, N.G.: Crater size-distribution and a revised martian relative chronology. *Icarus* 75(20), 285–305 (1988)
16. Data Mining and Machine Learning Group, University of Houston: CougarSquared Data Mining and Machine Learning Framework (2007), <https://cougarsquared.dev.java.net/>

Accurate and Efficient Retrieval of Multimedia Time Series Data Under Uniform Scaling and Time Warping

Waiyawuth Euachongprasit and Chotirat Ann Ratanamahatana

Chulalongkorn University
Department of Computer Engineering
Phayathai Rd., Pathumwan, Bangkok 10330 Thailand
{g50wch, ann}@cp.eng.chula.ac.th

Abstract. In this digital age, great interest has been shifted toward multimedia data manipulations. This includes videos, images, and audios, where typical manipulations require fairly large storage and are computationally intensive. Recent research has demonstrated the utilities of time series representation in various data mining tasks, allowing considerable reduction in time and space complexity. Specifically, the utilities of Uniform Scaling (US) and Dynamic Time Warping (DTW) have been shown to be necessary in several human-related domains, where uniform stretching or shrinking, as well as some local variation are typical. Classic examples include a query-by-humming system and motion capture data. However, all the past work has neglected the importance of data normalization before distance calculations, and therefore does not guarantee accurate retrievals. In this work, we discuss this concern and present a technique that accurately and efficiently searches under the US with DTW for normalized time series data, where no-false-dismissals are guaranteed.

Keywords: Data Mining, Content-based Multimedia Retrieval, Time Series, Uniform Scaling, Dynamic Time Warping.

1 Introduction

At present, multimedia data have evolved into our lives, where we increasingly have higher expectations in exploiting these data at hands. Typical manipulation usually requires fairly large amount of storage and is computationally intensive. Recently, it has been demonstrated that time series representation could be more efficient and effective in several domains, including science, bioinformatics, economics, and especially in multimedia [1]. For example, in a query-by-humming system, we can just extract a sequence of pitch from a sung query [2-6] to retrieve an intended song from the database. In motion retrieval, we can extract a sequence of motion in each video frame from a centroid of the object of interest in X, Y, and Z axes [7-9]. Similarly, in a content-based image search, image's shape can also be transformed into time series data for an efficient retrieval [10].

For the past decade, the most widely used distance measure in time series data has been Euclidean distance. It has been used for data retrieval, classification, clustering,

etc. However, Euclidean distance appears to be unsuitable for many applications because of its high sensitivity to variability in the time axis, and the superiority in accuracy of Dynamic Time Warping (DTW) over the Euclidean distance, which have been noted in various domains [7-9, 11, 12]. DTW, nonetheless, can handle only *local* variations in the data. Thus, it appears to be inappropriate for many applications, especially for multimedia or human-related domains where uniform stretching and shrinking are very typical [8, 12]; for example, in a query-by-humming system, most users tend to sing slower or faster than the original song. Similar problems also arise in other applications, such as motion capture and image retrievals [7, 9]. Recently, the use of Uniform Scaling (US) together with DTW has been introduced to mitigate this problem since US allows global scaling of time series before DTW distance calculation. However, this combination comes at a cost and cannot scale well with large databases. Fortunately, we have a lower-bounding function of this distance measure [12], which can efficiently prune most of the dissimilar candidate sequences to achieve significant speedup over these calculations.

The Importance of Normalization

Even though we now have a relatively efficient technique to speed up US with DTW calculations, in almost all applications, data pre-processing is still mandatory for accurate and meaningful similarity measurements. In addition, z-normalization and mean normalization are typically used when our primary interest is the time series' shapes.

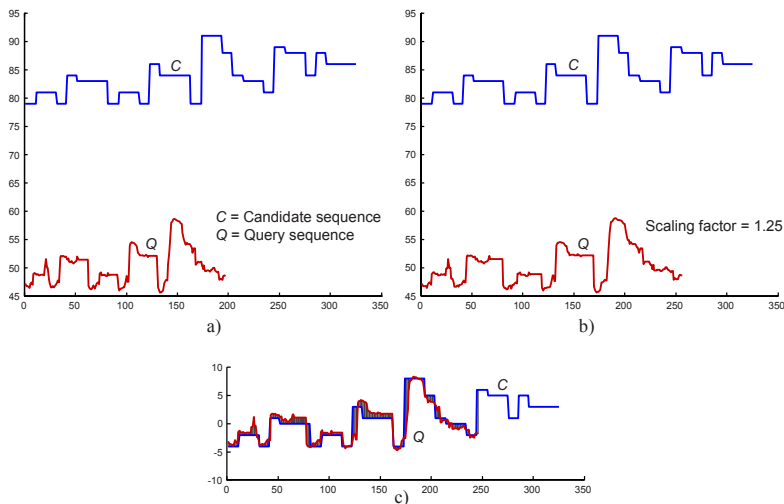


Fig. 1. a) A raw pitch contour extracted from a sung query represents a query sequence Q , and a MIDI pitch contour of “Happy Birthday” song represents a candidate sequence C . b) A re-scaled query sequence Q with scaling factor = 1.25. c) Both sequences after mean normalization at the query’s length. The shaded region shows their Euclidean distance.

In query by humming, for instance, we want to search the song database for the one whose segment is most similar to the sung query. However, most users may not sing queries in the same key or the same pitch level as the songs stored in the database. Thus, normalization of both the query and candidate sequences is needed to remove any existing offsets [3, 4, 13] before any distance calculations. An example of this problem is illustrated in Fig.1, where Q is the extracted sung query segment of the popular “Happy Birthday” song, and C is one of the candidate sequences stored in the database. We can quickly see that the shape of Q and a prefix of C are quite similar. Nevertheless, if we measure the similarity of these sequences directly using either Euclidean distance or DTW distance without any normalization nor rescaling, the distance will become excessive since both sequences may not be in the same pitch level nor in the right scale. Therefore, normalization and rescaling of both sequences before distance calculation are crucial steps to achieve an accurate retrieval (see Fig. 1 b) and c)).

In addition, the current lower-bounding method to prune off a large number of candidate sequences is developed without realization of the importance nor effects of data normalization. Hence, we propose a lower-bounding function to deal with this normalization problem efficiently and to calculate a distance under the US with DTW, where no false dismissals are guaranteed.

The rest of this paper is organized as follows. In section 2, we describe related research work and our motivation behind solving this normalization problem. Section 3 covers necessary background. In section 4, we describe our proposed method with a proof of no false dismissals. Section 5 verifies correctness of our method with a set of experiments to confirm the large pruning power in massive databases. Finally, section 6 gives some conclusions and offers possible future work.

2 Related Work

For the past decade, DTW has attracted many researchers because of its superiority in accuracy over the ubiquitous Euclidean distance, which has been widely known in a variety of domains and applications [1-3, 5, 7-12, 14, 15]. However, lack of ability to globally stretch or shrink a time series of DTW in dealing with tempo variations has been known in music retrieval community [6, 16]. A straightforward solution is to generate every possible scaled version of the query or the candidates to be used in the measurement; it is, however, impractical for large databases. Thus, some researchers have proposed the methods to address and resolve this concern efficiently. Keogh has proposed a lower-bounding function for the US that can speed up the calculation by two to three orders of magnitude [8]. In 2005, Fu et al. have extended Keogh’s method providing a solution for a lower-bounding distance calculation under US with DTW [12]. At this point, although there have been relatively efficient solutions to deal with both US and DTW, practically none of the researchers has realized the importance and effects of normalization under US, and this is a primary cause of flaws in their methods due to the inapplicable distance calculation. Regardless of the dire need in normalization as mentioned earlier, it has been neglected and in turn has blocked up both US and DTW to achieve high accuracy and efficient retrieval.

Our contribution is to propose an efficient lower-bounding function for US with DTW distance calculation under normalization requirement, which can prune a significant number of unqualified candidate sequences. Nonetheless, we would like to reemphasize that normalization is a crucial step to achieve a meaningful distance

calculation, especially in multimedia applications, as well as for efficient retrieval of the time series data.

3 Background

We begin with a formal problem definition as well as reviews of necessary background.

Problem definition. Suppose we have a query sequence Q of length m , where $Q = q_1, q_2, q_3, \dots, q_m$. It is scalable between lengths $sfmin*m$ and $sfmax*m$, where $sfmin$ and $sfmax$ are minimum and maximum scaling factors respectively, i.e., we can shrink or stretch a query sequence from lengths $sfmin*m$ to $sfmax*m$, where $sfmax \geq 1$ and $0 < sfmin \leq 1$. In addition, each candidate sequence C of length n , $C = c_1, c_2, c_3, \dots, c_n$, is stored in a database D . For simplicity, here, we define $n \geq sfmax*m$. Finally, we want to find the most similar-shaped candidate sequence C in the database D to the query sequence Q , which is also scalable in arbitrary lengths between $sfmin*m$ and $sfmax*m$.

Definition 1. Squared Euclidean distance: We define a squared Euclidean distance measure in eq.(1), which calculates distance between two sequences of equal length m (query's length). Note that the square root from the original Euclidean distance has been removed for an optimization purpose since the rankings of the results from both of these approaches are identical [8].

$$D(Q, C) \equiv \sum_{i=1}^m (q_i - c_i)^2 \quad (1)$$

Definition 2. Uniform Scaling: Uniform Scaling is a technique that uniformly stretches or shrinks a time series. In this approach, if we want to stretch a prefix of a candidate sequence C of length l to length m , we can use the Uniform Scaling function in eq.(2); shrinking of a candidate is done similarly to a stretching process.

We can formally define the Uniform Scaling function as follows.

$$c_j = c_{\lfloor j * l / m \rfloor} \quad \text{where } 1 \leq j \leq m \quad (2)$$

For US distance calculation, prefixes of a candidate sequence C of length l , where $\lfloor sfmin * m \rfloor \leq l \leq \min(\lfloor sfmax * m \rfloor, n)$, are rescaled to length m (query's length). Then we use a squared Euclidean distance function to calculate distance between a query sequence and all rescaled prefix sequences in order to find a minimum distance value ranging from $sfmin$ to $sfmax$.

The formal definition of a Uniform Scaling distance function (US) is provided in eq.(3) and eq.(4), where $RP(C, m, l)$ is a Rescaled Prefix function that returns a prefix of a candidate sequence of length l rescaled to length m .

$$RP(C, m, l)_i = c_{\lfloor i * l / m \rfloor} \quad \text{where } 1 \leq i \leq m \text{ and } 1 \leq l \leq m \quad (3)$$

$$US(Q, C, sfmin, sfmax) = \min_{l=\lfloor sfmin * m \rfloor}^{\min(\lfloor sfmax * m \rfloor, n)} D(RP(C, m, l), Q) \quad (4)$$

Definition 3. *Lower bounding of Uniform Scaling* [8, 12]: Lower bounding of Uniform Scaling is a distance approximation function, which can quickly compute a lower-bounding distance between a query and a candidate sequences; however, this lower bound value must not exceed the true distance value in order to be a valid lower-bounding function. To illustrate the idea, two new sequences are created, an upper envelope sequence UY and a lower envelope sequence LY , which enclose a candidate sequence. This envelope represents all scaled candidate sequences for a lower-bounding distance calculation.

UY and LY are formally defined in eq.(5), which was proposed in [8]. Note that a lower-bounding distance can simply be a squared Euclidean distance between a query sequence and the candidate's envelope, as defined in eq.(6).

$$UY_i = \max(c_{[i*sfmin]}, \dots, c_{[i*sfmax]}) \quad (5)$$

$$LY_i = \min(c_{[i*sfmin]}, \dots, c_{[i*sfmax]})$$

$$LBY(Q, C) = \sum_{i=1}^m \begin{cases} (q_i - UY_i)^2 & \text{if } q_i > UY_i \\ (LY_i - q_i)^2 & \text{if } q_i < LY_i \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

Interest readers may consult [12] for further details about US and DTW.

4 Our Proposed Method

Because the existing lower-bounding functions are not designed for distance calculation under normalization requirement, they are flawed and do not give correct and meaningful calculation. Consequently, this paper is highly motivated to fix this flaw by proposing US with DTW function under *normalization* condition as well as their efficient lower-bounding functions. Furthermore, our proposed lower-bounding function is able to prune a large number of candidate sequences without undergoing such costly distance calculation, primarily to speed up the computation with no false dismissals.

The failure in lower-bounding distance calculation under z-normalization condition of the previous work is illustrated in Fig. 2 b) and c). The shown query in Fig. 2 a) is a rescaled version of the candidate's prefix (scaling factor = 1.2). Then we normalize both sequences by using z-normalization as presented in Fig. 2 b) to e). However, in Fig. 2 b) and c) being the previously proposed lower-bounding functions [8, 12], it is apparent that their results (lower-bounding distance) are not zero, i.e., the normalized query is not fully contained within the lower-bounding envelopes, as illustrated by the shaded regions. This phenomenon definitely violates the lower-bounding rule because the lower-bounding distance must not exceed the true distance; in this case, the true distance should in fact be zero. Therefore, these existing lower-bounding functions could cause some false dismissals in normalization scheme. Actually, this phenomenon is not surprising since both of the previously proposed lower-bounding functions are not developed for the normalization problem. Figs. 2 d) and e) are our proposed lower-bounding function that satisfy all the lower bounding conditions.

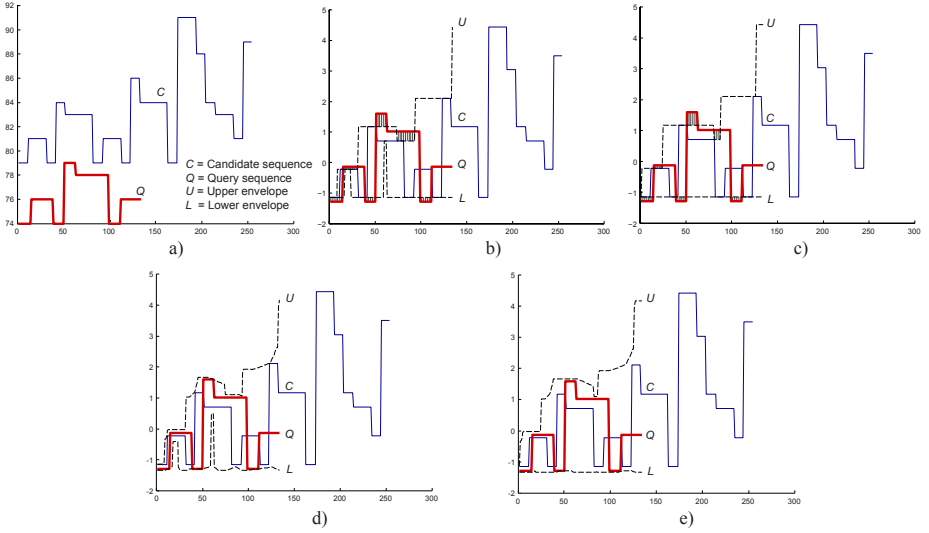


Fig. 2. a) Raw pitch contours extracted from a “Happy Birthday” song, where C represents the candidate, and Q is the query sung in a slower tempo (scaling factor = 1.2). Assume that a query sequence is no longer than the candidate sequence. b)–e) the query and the candidate sequences are z-normalized within the query’s length, enclosed by different lower-bounding envelopes with the scaling factor of range $[0.7, 1.3]$. b) A lower-bounding function of US that do not satisfy the lower-bounding condition [8]. c) A lower-bounding function of US with DTW, where a size of global constraint is 5%, which also does not satisfy the lower-bounding condition [12]. d) Our proposed lower-bounding function of US. e) Our proposed lower-bounding function of US with DTW, where a size of global constraint is 5%.

In this section, we begin with solutions for US and US with DTW distance measure, which satisfy normalization requirement, followed by the proof of no false dismissals.

Definition 4. Uniform Scaling with Normalization: The formal definition of a US with z-normalization is shown in eq.(7) and eq.(8), where Q' is a z-normalized query, and $\overline{c_{1..l}}$ and $SD(c_{1..l})$ are mean and standard deviation of a candidate’s prefix of length l , respectively. Although different scalings of the same sequence through interpolation may yield different mean and standard deviation values, our observation discovers no statistically significant difference of mean and standard deviation between normalization before rescaling the sequences and rescaling the sequences before normalization.

$$RP_{\text{norm}}(C, m, l)_i = \frac{\overline{c_{\lfloor j^*l/m \rfloor}} - \overline{c_{1..l}}}{SD(c_{1..l})} \quad \text{where } 1 \leq i \leq m \text{ and } 1 \leq j \leq m \quad (7)$$

$$US_{\text{norm}}(Q', C, sfmin, sfmax) = \min_{l=\lfloor sfmin * m \rfloor}^{\min(\lfloor sfmax * m \rfloor, n)} D(RP_{\text{norm}}(C, m, l), Q') \quad (8)$$

Definition 5. Lower bound of Uniform Scaling with Normalization: We develop a bounding envelope as expressed in eq.(9) and eq.(10), where UZ'_i and LZ'_i are upper and lower envelope sequences respectively. The corresponding distance calculation function is shown in eq.(11).

$$UZ'_i = \max\left(\frac{C_{sfmin*i} - C_{1..sfmin*m}}{SD(C_{1..sfmin*m})}, \dots, \frac{C_{sfmax*i} - C_{1..sfmax*m}}{SD(C_{1..sfmax*m})}\right)$$

$$= \max_{j=0}^{\lfloor m*(sfmax-sfmin) \rfloor} \left(\frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \right. \tag{9}$$

$$\left. \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})} \right)$$

$$LZ'_i = \min\left(\frac{C_{sfmin*i} - C_{1..sfmin*m}}{SD(C_{1..sfmin*m})}, \dots, \frac{C_{sfmax*i} - C_{1..sfmax*m}}{SD(C_{1..sfmax*m})}\right)$$

$$= \min_{j=0}^{\lfloor m*(sfmax-sfmin) \rfloor} \left(\frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \right. \tag{10}$$

$$\left. \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})}, \frac{C_{\lfloor sfmin*i+j*\frac{i}{m} \rfloor} - C_{1..\lfloor sfmin*m \rfloor+j}}{SD(C_{1..\lfloor sfmin*m \rfloor+j})} \right)$$

$$LBZ(Q', C) = \sum_{i=1}^m \begin{cases} (q_i - UZ'_i)^2 & \text{if } q_i > UZ'_i \\ (LZ'_i - q_i)^2 & \text{if } q_i < LZ'_i \\ 0 & \text{otherwise} \end{cases} \tag{11}$$

In case of US with DTW distance measure with normalization, we can simply change distance calculation function D in eq.(8) from the squared Euclidean function to DTW function. Additionally, its lower-bounding function is also quite straightforward that we can apply lower bounding of DTW over our envelope from eq.(9) and eq.(10), as shown in eq.(12); r is the size of a global constraint when using DTW distance calculation. The calculation of this lower-bounding function is similar to other functions as stated earlier. Due to space limitations, we omit full details of this distance function and its proof for brevity.

$$U'_i = \max(UZ'_{\max(1, i-r)}, \dots, UZ'_{\min(i+r, m)})$$

$$L'_i = \min(LZ'_{\max(1, i-r)}, \dots, LZ'_{\min(i+r, m)}) \tag{12}$$

Lastly, to validate the correctness of the proposed method, we complete this section with a proof of our lower bounding properties.

Proposition 1. Let Q' be a normalized query sequence of length m , and C be a candidate sequence of length n . In addition, a minimum scaling factor and a maximum scaling factor are $sfmin$ and $sfmax$ respectively, where $0 < sfmin \leq 1$ and $sfmax \geq 1$, i.e., the query can be scalable between $sfmin*m$ and $sfmax*m$. Then the value of $LBZ(Q',C)$ is a lower-bounding distance of $US_{norm}(Q',C,sfmin,sfmax)$.

Proof. Suppose $UZ'_i \geq c'_j \geq LZ'_i$, where UZ'_i is an upper envelope, LZ'_i is a lower envelope, and c'_j is a normalized data point of a candidate sequence at an arbitrary scaling between $sfmin$ and $sfmax$, i.e., $sfmin*i \leq j \leq sfmax*i$. Then $LBZ(Q',C) \leq US_{norm}(Q',C,sfmin, sfmax)$.

If a query is stretched to length $sf*m$, where sf is a scaling factor, a prefix of a candidate sequence with length $sf*m$ will be rescaled into length m and z-normalized by its mean and standard deviation, as shown in eq.(13).

$$c'_j = \frac{c_j - \overline{c_{1...sf*m}}}{SD(c_{1...sf*m})} \quad \text{where } 1 \leq j \leq m \tag{13}$$

From eq.(9) and eq.(10), the upper and lower envelopes are defined as follows.

$$UZ'_i \geq \frac{c_{sfmin*i} - \overline{c_{1...sfmin*m}}}{SD(c_{1...sfmin*m})}, \dots, \frac{c_{sfmax*i} - \overline{c_{1...sfmax*m}}}{SD(c_{1...sfmax*m})} \geq LZ'_i \tag{14}$$

From eq.(13) and (14), it follows that

$$UZ'_i \geq c'_k \geq LZ'_i \quad \text{where } sfmin*i \leq k \leq sfmax*i$$

$$\begin{aligned} (q_i - UZ'_i)^2 &< (q_i - c'_k)^2 && \text{if } q_i > UZ'_i \\ (LZ'_i - q_i)^2 &< (q_i - c'_k)^2 && \text{if } q_i < LZ'_i \\ 0 &&& \text{otherwise} \end{aligned}$$

$$LBZ(Q',C) \leq US_{norm}(Q',C,sfmin,sfmax) \tag{Q.E.D.}$$

5 Experiment

In the previous section, we introduce our proposed lower-bounding function as well as justification of its correctness in preserving the lower-bounding properties. In this section, we carefully evaluate the efficiency of our proposed method by conducting sets of experiments to observe the pruning power [11, 12]. Note that the pruning power is the fraction of the total candidate objects that can be discarded from further calculation, as defined in eq.(15).

$$Pruning\ Power = \frac{\text{Number of pruned candidates}}{\text{Total number of candidate sequences}} \tag{15}$$

In these experiments, we develop a simple Query-by-Humming system based on one-nearest-neighbor time series matching technique [3, 4, 6, 12, 17] in order to demonstrate the quality and utilities of our proposed method in multimedia database. We use 100 to 2,000 different international songs in MIDI file format and generate candidate sequences from this MIDI songs by using sliding windows of length $150 * sfmax$ data points, where $sfmax = 1.4$, and then store it in a database. For query sequences, we collect 55 sung queries from 12 subjects of both genders with various singing abilities and then extract sequences of pitch from these sung queries by using autocorrelation algorithm [18].

To carefully evaluate each factor that affects quality of the lower-bounding function, we conduct three experiments. In the first experiment, we examine an effect of different lengths of sequences with different ranges of scaling factors under 22441 sequences (generated from 100 songs), where a size of the global constraint is set to 4 percent of the sequences' length (see Fig. 3 a)). In the second experiment, we investigate an effect of different lengths of the sequences with different sizes of global constraint under 22441 sequences, where range of scaling factor is between 0.8 and 1.2 (see Fig. 3 b)). In the last experiment, we use 22441, 55595, 107993, 220378, and 442366 subsequences from 100, 250, 500, 1000, and 2000 songs to construct different-sized databases in order to observe their pruning powers, as shown in Fig. 4.

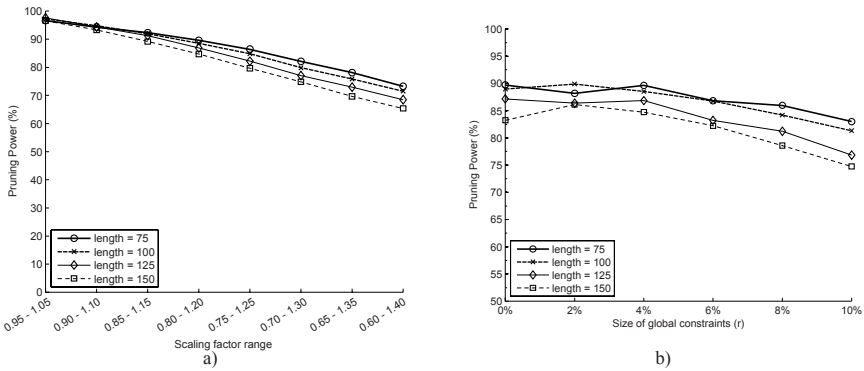


Fig. 3. a) The pruning powers of different time series lengths with various ranges of scaling factors. b) The pruning powers of different lengths with various sizes of global constraint.

According to these experiment results, our proposed lower-bounding function can prune a great number of candidate sequences in every parameter setting, as shown in Figs. 3 and 4. However, from these experiments, we found that there are several factors influencing the pruning power, and range of scaling factors is one of them. In Fig. 3 a), increases in range of scaling factors are likely to decrease the pruning power directly. Besides, the pruning power slightly decreases as the sizes of global constraint increase (see Fig. 3 b)). The rationale behind these results is that both increases in range of scaling factors and in size of global constraint definitely enlarge the size of

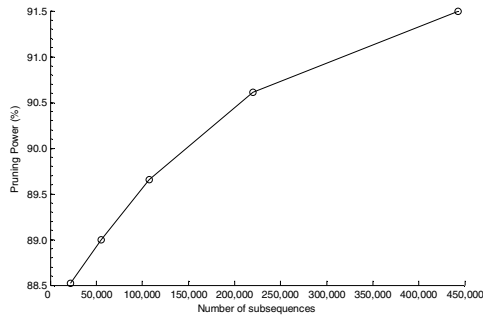


Fig. 4. The pruning powers of different database's sizes

the lower-bounding envelopes, causing a reduction in the lower bounding distance, and hence its pruning power. In addition, longer sequences appear to give smaller pruning power than that of the shorter sequences as illustrated in Figs. 3 and 4. In contrast, the pruning power is found to increase as the database size increases (see Fig. 4), which is a highly desirable property for lower-bounding functions.

Regardless of a few factors that decrease the pruning power, we discover no significant improvement in accuracy when we increase scaling factor range over 0.6-1.4 in our experiment. In addition, from recent research [19], wider sizes of the global constraint do not imply higher accuracy. In fact, in most cases, the size of the global constraint of less than 10 percent often yields optimal accuracy.

Notice that the normalization does affect the pruning power because the distances between the normalized query and the normalized candidate sequences are greatly reduced, comparing with the distances among unnormalized sequences. However, we would like to reemphasize that *normalization* is essential in many applications.

6 Discussion and Conclusions

We have shown that this proposed lower-bounding function of US with DTW under *normalization* requirement can efficiently prune a large number of candidates in the database, significantly reducing the time complexity in the data retrieval, especially for multimedia retrieval, while no false dismissals are also guaranteed. Furthermore, our approach can work well with other types of normalization. Nonetheless, we would like to reemphasize the importance and necessity of normalization, especially in multimedia applications.

Besides dramatically speeding up the calculations by pruning almost all candidates, this lower-bounding function is possible to utilize dimensionality reduction and indexing techniques [9] in order to be scalable to truly massive databases.

Acknowledgments. We would like to thank Dr. Boonserm Kijisirikul for valuable comments and enlightening discussions. This work is partially supported by the Thailand Research Fund (Grant No. MRG5080246).

References

1. Sakurai, Y., Yoshikawa, M., Faloutsos, C.: FTW: Fast Similarity Search under the Time Warping Distance. In: Proceedings of 24th ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 326–337. ACM Press, Baltimore, Maryland (2005)
2. Adams, N.H., Bartsch, M.A., Shifrin, J.B., Wakefield, G.H.: Time series alignment for music information retrieval. In: Proceedings of 5th International Conference on Music Information Retrieval, Barcelona, Spain (2004)
3. Zhu, Y., Shasha, D.: Warping indexes with envelope transforms for query by humming. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pp. 181–192. ACM Press, San Diego, California (2003)
4. Jang, J.-S.R., Lee, H.-R.: Hierarchical filtering method for content-based music retrieval via acoustic input. In: Proceedings of 9th ACM international conference on Multimedia, pp. 401–410. ACM Press, Ottawa, Canada (2001)
5. Lee, H.-R., Chen, C., Jang, J.-S.R.: Approximate lower-bounding functions for the speedup of DTW for melody recognition. In: International Workshop on Cellular Neural Networks and Their Applications, pp. 178–181 (2005)
6. Hu, N., Dannenberg, R.B.: A comparison of melodic database retrieval techniques using sung queries. In: Proceedings of 2nd ACM/IEEE-CS joint conference on Digital libraries, pp. 301–307. ACM Press, Portland, Oregon, USA (2002)
7. Keogh, E., Palpanas, T., Zordan, V.B., Gunopulos, D., Cardle, M.: Indexing Large Human-Motion Databases. In: Proceedings of 30th VLDB Conference, Toronto, Canada (2004)
8. Keogh, E.: Efficiently Finding Arbitrarily Scaled Patterns in Massive Time Series Databases. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, pp. 253–265. Springer, Heidelberg (2003)
9. Keogh, E., Celly, B., Ratanamahatana, C.A., Zordan, V.B.: A novel technique for indexing video surveillance data. In: 1st ACM SIGMM international workshop on Video surveillance, pp. 98–106. ACM Press, Berkeley, California, USA (2003)
10. Shu, S., Narayanan, S., Kuo, C.-C.J.: Efficient Rotation Invariant Retrieval of Shapes using Dynamic Time Warping with Applications in Medical Databases. In: Proceedings of 19th IEEE International Symposium on Computer-Based Medical Systems (CBMS), pp. 673–678 (2006)
11. Keogh, E., Ratanamahatana, C.A.: Exact indexing of dynamic time warping. *Knowledge and Information Systems: An International Journal (KAIS)* 7, 358–386 (2004)
12. Fu, A.W.-c., Keogh, E., Lau, L.Y.H., Ratanamahatana, C.A.: Scaling and time warping in time series querying. In: Proceedings of 31st international conference on Very large data bases. VLDB Endowment, Trondheim, Norway, pp. 649–660 (2005)
13. Wang, Z., Zhang, B.: Quotient space model of hierarchical query-by-humming system. In: Proceedings of IEEE International Conference on Granular Computing, vol. 2, pp. 671–674 (2005)
14. Ratanamahatana, C.A., Keogh, E.: Multimedia Retrieval Using Time Series Representation and Relevance Feedback. In: Fox, E.A., Neuhold, E.J., Premssmit, P., Wuwongse, V. (eds.) ICADL 2005. LNCS, vol. 3815, pp. 400–405. Springer, Heidelberg (2005)
15. Ratanamahatana, C.A., Keogh, E.: Making Time-Series Classification More Accurate Using Learned Constraints. In: Proceedings of SIAM International Conference on Data Mining (SDM), Lake Buena Vista, Florida, USA, pp. 11–22 (2004)

16. Meek, C., Birmingham, W.P.: The dangers of parsimony in query-by-humming applications. In: Proceedings of 4th International Symposium on Music Information Retrieval (ISMIR) (2003)
17. Dannenberg, R.B., Birmingham, W.P., Pardo, B., Hu, N., Meek, C., Tzanetakis, G.: A comparative evaluation of search techniques for query-by-humming using the MUSART testbed. *Journal of the American Society for Information Science and Technology* 58, 687–701 (2007)
18. Boersma, P., Weenink, D.: Praat: doing phonetics by computer (version 4.4.13) (2005), <http://www.praat.org/>
19. Ratanamahatana, C.A., Keogh, E.J.: Everything you know about dynamic time warping is wrong. In: 3rd Workshop on Mining Temporal and Sequential Data, in conjunction with 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2004) (2004)

Feature Construction Based on Closedness Properties Is Not That Simple

Dominique Gay¹, Nazha Selmaoui¹, and Jean-François Boulicaut²

¹ ERIM EA 3791, University of New Caledonia,
BP R4, F-98851 Nouméa, New Caledonia
{dominique.gay, nazha.selmaoui}@univ-nc.nc

² INSA-Lyon, LIRIS CNRS UMR5205
F-69621 Villeurbanne Cedex, France
jean-francois.boulicaut@insa-lyon.fr

Abstract. Feature construction has been studied extensively, including for 0/1 data samples. Given the recent breakthrough in closedness-related constraint-based mining, we are considering its impact on feature construction for classification tasks. We investigate the use of condensed representations of frequent itemsets (closure equivalence classes) as new features. These itemset types have been proposed to avoid set counting in difficult association rule mining tasks. However, our guess is that their intrinsic properties (say the maximality for the closed itemsets and the minimality for the δ -free itemsets) might influence feature quality. Understanding this remains fairly open and we discuss these issues thanks to itemset properties on the one hand and an experimental validation on various data sets on the other hand.

1 Introduction

Feature construction is one of the major research topics for supporting classification tasks. Based on a set of original features, the idea is to compute new features that may better describe labeled samples such that the predictive accuracy of classifiers can be improved. When considering the case of 0/1 data (i.e., in most of the cases, collections of attribute-value pairs that are true or not within a sample), several authors have proposed to look at feature construction based on patterns that satisfy closedness-related constraints [1,2,3,4,5,6]. Using patterns that hold in 0/1 data as features (e.g., itemsets or association rules) is not new. Indeed, pioneering work on classification based on association rules [7] or emerging pattern discovery [8,9] have given rise to many proposals. Descriptive pattern discovery from unlabeled 0/1 data has been studied extensively during the last decade: many algorithms have been designed to compute every set pattern that satisfies a given constraint (e.g., a conjunction of constraints whose one conjunct is a minimal frequency constraint). One breakthrough into the computational complexity of such mining tasks has been obtained thanks to condensed

representations for frequent itemsets, i.e., rather small collections of patterns from which one can infer the frequency of many sets instead of counting for it (see [10] for a survey). In this paper, we consider closure equivalence classes, i.e., frequent closed sets and their generators [11]. Furthermore, when considering the δ -free itemsets with $\delta > 0$ [12,13], we can consider a “near equivalence” perspective and thus, roughly speaking, the concept of almost-closed itemsets. We want to contribute to difficult classification tasks by using a method based on: (1) the efficient extraction of set patterns that satisfy given constraints, (2) the encoding of the original data into a new data set by using extracted patterns as new features. Clearly, one of the technical difficulties is to discuss the impact of the intrinsic properties of these patterns (i.e., closedness-related properties) on a classification process.

Our work is related to pattern-based classification. Since [7], various authors have considered the use of association rules. These proposals are based on a pruned set of extracted rules built w.r.t. support and confidence ranking. Differences between these methods mainly come from the way they use the selected set of rules when an unseen example x is coming. For example, CBA [7] ranks the rules and it uses the best one to label x . Other algorithms choose the class that maximizes a defined score (CMAR [14] uses $\frac{\text{support}(x)}{\text{confidence}(x)}$ of subsets of rules when CPAR [15] uses $\frac{\text{support}(x)}{\text{confidence}(x)}$ of the best k rules). Also, starting from ideas for class characterization [16], [17] is an in-depth formalization of all these approaches. Another related research stream concerns emerging patterns [18]. These patterns are frequent in samples of a given class and infrequent for samples from the other classes. Several algorithms have exploited this for feature construction. Some of them select essential ones (CAEP classifier [8]) or the most expressive ones (JEPs classifier [9]). Then, an incoming example is labeled with the class c which maximizes scores based on these sets. Moreover, a few researchers have considered condensed representations of frequent sets for feature construction. Garriga et al. [3] have proposed to characterize a target class with a collection of relevant closed itemsets. Li et al. [1] invoke MDL principle and suggest that free itemsets might be better than closed ones. However, classification experimental results to support such a claim are still lacking. It turns out that the rules studied in [17] are based on 0-free sets such that a minimal body property holds. The relevancy of such a minimality property is also discussed in terms of “near equivalence” in [19]. In [2], we have considered preliminary results on feature construction based on δ -freeness [12,13]. Feature construction approaches based on closedness properties differ in two main aspects: \blacktriangleright mining can be performed on the whole database or per class, and \blacktriangleright we can mine with or without the class labels. The pros and cons of these alternatives are discussed in this paper.

In Section 2, we provide more details on state-of-the-art approaches before introducing our feature construction method. Section 3 reports on our experimental results for UCI data sets [20] and a real-world medical database. Section 4 concludes.

2 Feature Construction Using Closure Equivalence Classes

A γ -frequent itemset I in r is defined as a binary relation $(\mathcal{T}, \mathcal{I}, R)$ where \mathcal{T} is a set of objects (or transactions), \mathcal{I} is a set of attributes (or items) and $R \subseteq \mathcal{T} \times \mathcal{I}$. The frequency of an itemset $I \subseteq \mathcal{I}$ in r is $freq(I, r) = |Objects(I, r)|$ where $Objects(I, r) = \{t \in \mathcal{T} \mid \forall i \in I, (t, i) \in R\}$. Let γ be an integer, an itemset I is said to be γ -frequent if $freq(I, r) \geq \gamma$.

Considering that frequent itemset mining is intuitive, Cheng et al. [4] brought some evidence to support such a claim and they have linked frequency with other interestingness measures such as Information Gain and Fisher score. Since the number of frequent itemsets can be huge in dense databases, it is now common to use condensed representations (e.g., free itemsets, closed ones, non derivable itemsets [10]) to save space and time during the frequent itemset mining task and to avoid some redundancy.

Definition 1 (Closed itemset). An itemset I is a closed itemset in r iff $\nexists I' \supset I$ such that $freq(I', r) = freq(I, r)$. The closure operator $cl : \mathcal{P}(\mathcal{I}) \rightarrow \mathcal{P}(\mathcal{I})$ is defined as $cl(I, r) = Items(Objects(I, r), r)$ where $Items(T, r) = \{i \in \mathcal{I} \mid \forall t \in T, (t, i) \in R\}$. It holds that $cl(I, r) \equiv Items(Objects(I, r), r)$ and $I = cl(I, r)$ iff I is a closed itemset.

Since [11], it is common to formalize the fact that many itemsets have the same closure by means of

Definition 2 (Closure equivalence). Two itemsets I and J are closure equivalent in r iff $cl(I, r) = cl(J, r)$. The closure equivalence class of I is $cl(I, r) = \{J \mid I \sim_{cl} J\}$. The set of all closure equivalence classes is denoted by $\mathcal{C}(r)$. It holds that $Objects(I, r) = Objects(J, r)$ iff $I \sim_{cl} J$.

Each CEC contains exactly one maximal itemset (w.r.t. set inclusion) which is a closed itemset. It may contain several minimal itemsets which are 0-free itemsets according to the terminology in [12] (also called key patterns in [11]).

Considering Tab. 1, we have $r = (\mathcal{T}, \mathcal{I}, R)$, $\mathcal{T} = \{t_1, \dots, t_6\}$, and $\mathcal{I} = \{A, B, C, D, c_1, c_2\}$, c_1 and c_2 being the class labels. For a frequency threshold $\gamma = 2$, itemsets AB and AC are γ -frequent. $ABCc_1$ is a γ -frequent closed itemset. Considering the equivalence class $\mathcal{C} = \{AB, AC, ABC, ABc_1, ACc_1, ABCc_1\}$, AB and AC are its minimal elements (i.e., they are 0-free itemsets) and $ABCc_1$ is the maximal element, i.e., one of the closed itemsets in this toy database.

2.1 Freeness or Closedness?

Two different approaches for feature construction based on condensed representations have been considered so far. In, e.g., [15], the authors mine free itemsets and closed itemsets (i.e., CECs) once the class attribute has been removed from

Table 1. A toy example of a binary labeled database

r	A	B	C	D	c_1	c_2
t_1	1	1	1	1	1	0
t_2	1	1	1	0	1	0
t_3	0	1	1	0	1	0
t_4	1	0	0	1	1	0
t_5	0	1	1	0	0	1
t_6	0	1	0	1	0	1

the entire database. Other proposals, e.g., [34], consider (closed) itemset mining from samples of each class separately.

Looking at the first direction of research, we may consider that closed sets, because of their maximality, are good candidates for characterizing labeled data, but not necessarily suitable to predict classes for unseen samples. Moreover, thanks to their minimality, free itemsets might be better for predictive tasks. Due to closedness properties, every itemset of a given closure equivalence class \mathcal{C} in r covers exactly the same set of objects. Thus, free itemsets and their associated closed are equivalent w.r.t. interestingness measures based on frequencies. As a result, it is unclear whether choosing a free itemset or its closure to characterize a class is important or not. Let us now consider an incoming sample x (test phase) that is exactly described by the itemset Y (i.e., all its properties that are true are in Y). Furthermore, assume that we have $F \subseteq Y \subseteq cl(F, r)$ where F is a free itemset from the closure equivalence class \mathcal{C}_F . Using free itemsets to label x will not lead to the same decision than using closed itemsets. Indeed, $x \supseteq F$ and it satisfies rule $F \Rightarrow c$ while $x \not\supseteq cl(Y, r)$ and it does not satisfy rule $cl(F, r) \Rightarrow c$. Following that direction of work, Baralis et al. have proposed classification rules based on free itemsets [17].

On the other hand, for the “per-class” approach, let us consider w.l.o.g a two-class classification problem. In such a context, the equivalence between free itemsets and their associated closed ones is lost. The intuition is that, for a given free itemset Y in r_{c_1} -database restricted to samples of class c_1 - and its closure $X = cl(Y, r_{c_1})$, X is more relevant than Y since $Objects(X, r_{c_1}) = Objects(Y, r_{c_1})$ and $Objects(X, r_{c_2}) \subseteq Objects(Y, r_{c_2})$. The closed itemsets (say $X = cl(X, r_{c_1})$) such that there is no other closed itemset (say $X' = cl(X', r)$) for which $cl(X, r_{c_2}) = cl(X', r_{c_2})$ are chosen as relevant itemsets to characterize c_1 . In some cases, a free itemset Y could be equivalent to its closure $X = cl(Y, r_{c_1})$, i.e., $Objects(X, r_{c_2}) = Objects(Y, r_{c_2})$. Here, for the same reason as above, a free itemset may be chosen instead of its closed counterpart. Note that relevancy of closed itemsets does not avoid conflicting rules, i.e., we can have two closed itemsets X relevant for c_1 and Y relevant for c_2 with $X \subseteq Y$.

Moreover, these approaches need for a post-processing of the extracted patterns. Indeed, we not only look for closedness-related properties but we have also to exploit interesting measures to keep only the ones that are discriminating. To avoid such a post-processing, we propose to use syntactic constraint (i.e., keeping

the class attribute during the mining phase) to mine class-discriminant closure equivalence classes.

2.2 What Is Interesting in Closure Equivalence Classes?

In Fig. 1, we report the different kinds of CECs that can be obtained when considering class attributes during the mining phase.

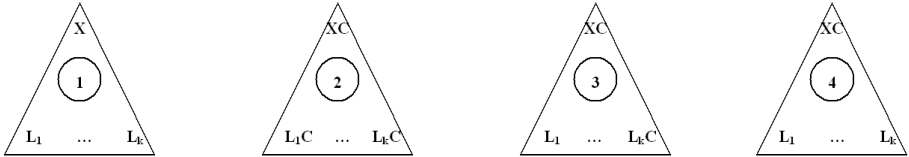


Fig. 1. Different types of CECs

These CECs have nice properties that are useful to our purpose: since association rules with a maximal confidence (no exception, also called hereafter exact rules) stand between a free itemset and its closure, we are interested in CECs whose closure contains a class attribute to characterize classes. Thus, we may neglect Case 1 in Fig. 1.

Definition 3 (Association rule). $r = \{T, I, R\}$ association rule π $I \Rightarrow J$ $I \subseteq \mathcal{I}$ $J \subseteq \mathcal{I} \setminus I$ frequency $freq(I \cup J, r)$ π $conf(\pi, r) = freq(I \cup J, r) / freq(I, r)$ confidence π J classification rule

From Case 3 (resp. Case 4), we can extract the exact classification rule $\pi_3 : L_1 \Rightarrow C$ (resp. the exact rules $\pi_{4_1} : L_1 \Rightarrow C \dots \pi_{4_k} : L_k \Rightarrow C$). Note that if we are interested in exact rules only, we also neglect Case 2: L_1C is a free itemset and it implies there is no exact rule $I \Rightarrow J$ such that $I \cup J \subseteq L_1C$. Thus, we are interested in CECs whose closed itemset contains a class attribute and whose free itemsets (at least one) do not contain a class attribute. This also leads to a closedness-related condensed representation of Jumping Emerging Patterns [21]. Unfortunately, in pattern-based classification (a fortiori in associative classification), for a given frequency threshold γ , mining exact rules is restrictive since they can be rare and the training database may not be covered by the rule set. In a relaxed setting, we consider association rules that enable exceptions.

Definition 4 (δ -strong rule, δ -free itemset). $I \Rightarrow^\delta J$ δ -strong rule $I \subseteq \mathcal{I}$ $J \subseteq \mathcal{I} \setminus I$ $I \subseteq \mathcal{I}$ δ -free itemset $\delta = 0$ strong rules free itemsets

When the right-hand side is a single item i , saying that $I \Rightarrow^\delta i$ is a δ -strong rule in r means that $freq(I, r) - freq(I \cup \{i\}) \leq \delta$. When this item is a class attribute, a δ -strong rule is called a δ -strong classification rule [16].

The set of δ -strong rules can be built from δ -free itemsets and their δ -closures.

Definition 5 (δ -closure). For $\delta \in \mathbb{N}$, the δ -closure of an itemset I is $cl_\delta(I, r) = \{i \in \mathcal{I} \mid freq(I, r) - freq(I \cup \{i\}) \leq \delta\}$. For $\delta = 0$, $cl_0(I, r) = \{i \in \mathcal{I} \mid freq(I, r) = freq(I \cup \{i\})\}$. The δ -closure equivalence classes are $I \sim_{cl_\delta} J$ if $cl_\delta(I, r) = cl_\delta(J, r)$.

The intuition is that the δ -closure of a set I is the superset X of I such that every added attribute is almost always true for the objects which satisfy the properties from I : at most δ false values (or exceptions) are enabled. The computation of every frequent δ -free set (i.e., sets which are both frequent and δ -free) can be performed efficiently [13]. Given threshold values for γ (frequency) and δ (freeness), the used `AC_like`¹ implementation outputs each δ -free frequent itemset and its associated δ -closure. Considering Table 1, a frequency threshold $\gamma = 3$ and a number of exceptions $\delta = 1$, itemset C is a 3-frequent 1-free itemset ; items B and c_1 belong to its δ -closure and $\pi : C \Rightarrow^\delta c_1$ is a 1-strong classification rule.

2.3 Information and Equivalence Classes

We get more information from δ -closure equivalence classes than with other approaches. Indeed, when considering contingency tables (See Tab. 2), for all the studied approaches, f_{*1} and f_{*0} are known (class distribution). However, if we consider the proposals from [3,4] based on frequent closed itemsets mined per class, we get directly the value f_{11} (i.e., $freq(X \cup c, r)$) and the value for f_{01} can be inferred. Closure equivalence classes in [5] only inform us on f_{1*} (i.e., $freq(X, r)$) and f_{0*} . In our approach, when mining γ -frequent δ -free itemsets whose closure contains a class attribute, $f_{1*} \geq \gamma$ and we have a lower bound $f_{11} \geq \gamma - \delta$ and an upper bound $f_{10} \leq \delta$ for frequencies on X . We can also infer other bounds for f_{01} and f_{00} ².

Table 2. Contingency table for a δ -strong classification rule $X \Rightarrow^\delta c$

$X \Rightarrow c$	c	\bar{c}	Σ
X	f_{11}	f_{10}	f_{1*}
\bar{X}	f_{01}	f_{00}	f_{0*}
Σ	f_{*1}	f_{*0}	f_{**}

Moreover, γ -frequent δ -free itemsets, bodies of δ -strong classification rules are known to have a minimal body property. Some constraints on γ and δ can help

¹ `AC_like` implementation is available at <http://liris.cnrs.fr/jeremy.besson/>

² Note the confidence of a δ -strong classification rule π is $f_{11}/f_{1*} \geq 1 - (\delta/\gamma)$.

to avoid some of the classification conflicts announced at the end of Section 2.1. Indeed, [16] has shown that setting $\delta \in [0; \lfloor \gamma/2 \rfloor[$ ensures that we can not have two classification rules $\pi_1 : I \Rightarrow^\delta c_i$ and $\pi_2 : I \Rightarrow^\delta c_j$ with $i \neq j$ s.t. $I \subseteq J$. This constraint also enforces confidence to be greater than $\frac{1}{2}$. Furthermore, we know that we can produce δ -strong classification rules that exhibit the discriminant power of emerging patterns if $\delta \in [0; \gamma \cdot (1 - \frac{|r_{c_i}|}{|r|})[$, r_{c_i} being the database restricted to objects of the majority class c_i [6]. One may say that the concept of γ -frequent δ -free itemsets ($\delta \neq 0$) can be considered as an interestingness measures (function of γ and δ) for feature selection.

2.4 Towards a New Space of Descriptors

Once γ -frequent (δ)-free itemsets have been mined, we can build a new representation of the original database using these new features. Each selected itemset I will generate a new attribute $NewAtt_I$ in the new database. One may encode $NewAtt_I$ to a binary attribute, i.e., for a given object t , $NewAtt_I$ equals 1 if $I \subseteq Items(t, r)$ else 0. In a relaxed setting and noise-tolerant way, we propose to compute $NewAtt_I$ as follows:

$$NewAtt_I(t) = \frac{|I \cap Items(t, r)|}{|I|}$$

This way, I is a multivalued ordinal attribute. It is obvious that for an object t , $NewAtt_I(t) \in \{0, 1, \dots, \frac{p-1}{p}, 1\}$ where $p = |I|$. Then, the value $NewAtt_I(t)$ is the proportion of items $i \in I$ that describe t . We think that multivalued encoding –followed by an entropy-based supervised discretization step³– should hold more information than binary encoding. Indeed, in the worst case, the split will take place between $\frac{p-1}{p}$ and 1, that is equivalent to binary case; in other better cases, split may take place between $\frac{j-1}{p}$ and $\frac{j}{p}$, $1 \leq j \leq p-1$ and this split leads to a better separation of data.

3 Experimental Validation

The frequency threshold γ and the accepted number of exceptions δ are important parameters for our Feature Construction (FC) proposal. Let us discuss how to set up sensible values for them. Extreme values for γ bring either (for lowest values) a huge amount of features –some of which are obviously irrelevant– or (for highest values) not enough features to correctly cover the training set. Furthermore, in both cases, these solutions are of limited interest in terms of Information Gain (see [4]). Then, δ varies from 0 to $\gamma \cdot (1 - \frac{|r_{c_i}|}{r})$ to capture discriminating power of emerging patterns. Once again, lowest values of δ lead to γ -frequent emerging patterns but a potentially low coverage proportion of data and features with high values of δ lacks of discriminating power.

³ The best split between 2 values is recursively chosen until no more information is gained.

Intuitively, a high coverage proportion implies a relatively good representation of data. In Fig. 2, we plotted proportion of the database coverage w.r.t. δ for a given frequency threshold. Results for **breast**, **cleve**, **heart** and **hepatic** data (from UCI repository) are reported. We easily observe that coverage proportion grows as δ grows. Then, it reaches a saturation point for δ_0 which is interesting: higher values of $\delta > \delta_0$ are less discriminant and lower values $\delta < \delta_0$ cover less objects. In our following experiments, we report (1) maximal accuracies over all γ and δ values (denoted Max), and (2) average accuracies of all γ values with $\delta = \delta_0$ (denoted Av).

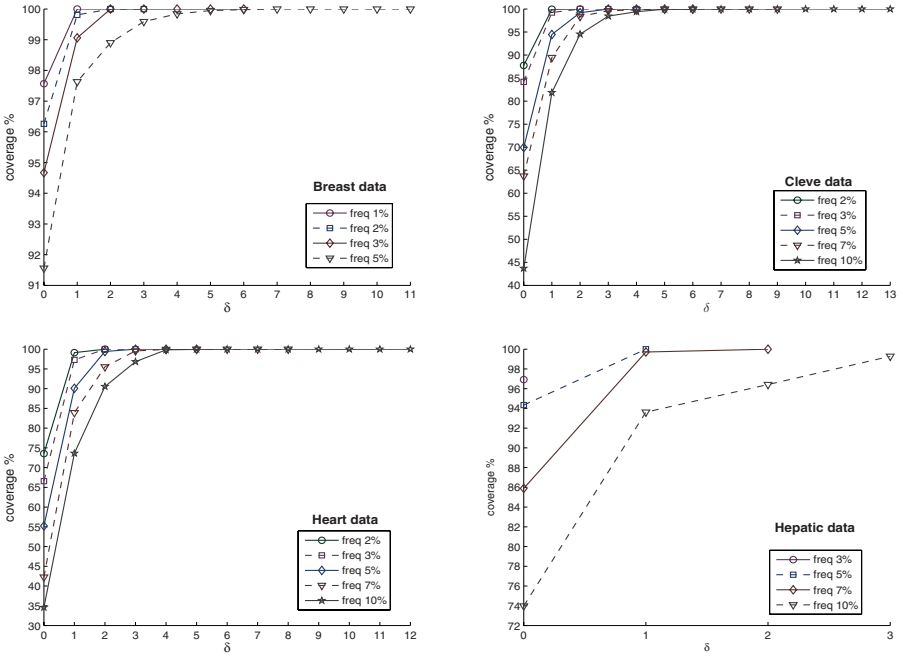


Fig. 2. Evolution of training database coverage proportion w.r.t. γ and δ

To validate our feature construction (FC) process, we used it on several data sets from UCI repository [20] and a real-world data set meningitis⁴. We have been using popular classification algorithms such as NB and C4.5 on both the original data and the new representation based on extracted features. As a result, our main objective criterion is the accuracy of the obtained classifiers.

Notice that before performing feature construction, we translated all attributes into binary ones. While the translation of nominal attributes is straightforward, we decided to discretize continuous attributes with the entropy-based method by Fayyad et al. [22]. Discretizations and classifier constructions have been performed with WEKA [23] (10-folds stratified cross validation).

⁴ meningitis concerns children hospitalized for acute bacterial or viral meningitis.

Table 3. Accuracy results improvement thanks to FC

databases	NB	FC & NB (Av/Max)	C4.5	FC & C4.5 (Av/Max)
breast	95.99	97.32/97.54	94.56	96.12/96.43
car	85.53	81.95/84.64	92.36	98.49/99.13
cleve	83.5	83.35/ 84.33	76.24	81.39/83.18
crx	77.68	85.91/86.46	86.09	83.95/ 86.33
diabetes	75.91	75.56/ 76.59	72.26	76.03/77.75
heart	84.07	83.62/ 84.81	80	84.56/85.55
hepatic	83.22	84.09/84.67	81.93	85.29/86.83
horse	78.8	81.09/83.74	85.33	83.35/ 85.40
iris	96	94.26/ 96	96	94.26/ 96.67
labor	94.74	93.5/ 95.17	78.95	83.07/87.17
lymph	85.81	83.35/85.46	76.35	81.08/83.46
meningitis	95.74	93.24/93.64	94.83	92.54/ 95.13
sonar	69.71	85.17/86.28	78.85	79.88/83.86
vehicle	45.03	59.72/62.88	71.04	70.70/71.28
wine	96.63	96.42/ 97.83	94.38	95.57/96.29

Table 4. Our FC Feature Construction proposal vs. state-of-the-art approaches

databases	BCEP	LB	FC&NB(Av/Max)	SJEP	CBA	CMAR	CPAR	FC&C4.5(Av/Max)
breast	–	96.86	97.32/97.54	96.96	96.3	96.4	96.0	96.12/96.43
car	–	–	81.95/84.64	–	88.90	–	92.65	98.49/99.13
cleve	82.41	82.19	83.35/84.33	82.41	82.8	82.2	81.5	81.39/ 83.18
crx	–	–	85.91/86.46	87.65	84.7	84.9	85.7	83.95/86.33
diabetes	76.8	76.69	75.56/76.59	76.18	74.5	75.8	75.1	76.03/ 77.75
heart	81.85	82.22	83.62/84.81	82.96	81.9	82.2	82.6	84.56/85.55
hepatic	–	84.5	84.09/ 84.67	83.33	81.8	80.5	79.4	85.29/86.83
horse	–	–	81.09/83.74	84.17	82.1	82.6	84.2	83.35/ 85.40
iris	–	–	94.26/96	–	94.7	94.0	94.7	94.26/ 96.67
labor	–	–	93.5/95.17	82	86.3	89.7	84.7	83.07/87.17
lymph	83.13	84.57	83.35/ 85.46	–	77.8	83.1	82.3	81.08/ 83.46
meningitis	–	–	93.24/93.64	–	91.79	–	91.52	92.54/95.13
sonar	78.4	–	85.17/86.28	85.10	77.5	79.4	79.3	79.88/83.86
vehicle	68.05	68.8	59.72/62.88	71.36	68.7	68.8	69.5	70.70/71.28
wine	–	–	96.42/97.83	95.63	95.0	95.0	95.5	95.57/ 96.29

We report in Tab. 3 the accuracy results obtained on both the original data and its new representation. NB, C4.5 classifiers built on the new representation often perform better (i.e., it lead to higher accuracies) than respective NB and C4.5 classifiers built from the original data. One can see that we have often (12 times among 15) a combination of γ and δ for which NB accuracies are improved by feature construction (column Max). And this is experimentally always the case for C4.5. Now considering average accuracies (column Av), improvement is still there w.r.t. C4.5 but it appears less obvious when using NB.

Then, we also compared our results with state-of-the-art classification techniques: FC & NB is compared with other bayesian approaches, LB [24] and BCEP [25]. When accessible, accuracies were reported from original papers within Tab. 4. Then, we have compared FC & C4.5 with other associative classification approaches, namely CBA [7], CMAR [14], CPAR [15], and an EPs-based classifier SJEP-classifier [26]. Accuracy results for associative classifiers are taken from [14]. Others results are taken from the published papers. FC allows to often achieve better accuracies than the state-of-the-art classifiers, e.g., FC & C4.5 wins 9 times over 15 against CPAR, 8 times over 13 against CMAR, 10 times over 15 against CBA when considering average accuracies (column Av). Considering optimal γ and δ values (column Max), it wins 10 times over 15 (see bold faced results).

4 Conclusion

We study the use of closedness-related condensed representations for feature construction. We pointed out that differences about “freeness or closedness” within existing approaches come from the way that condensed representations are mined : with or without class label, per class or in the whole database. We proposed a systematic framework to construct features. Our new features are built from mined (δ)-closure equivalence classes – more precisely from γ -frequent δ -free itemsets whose δ -closures involve a class attribute. Mining these types of itemsets differs from other approaches since (1) mined itemsets hold more information (such as emergence) and (2) there is no need for post-processing the set of features to select interesting features. We also proposed a new numeric encoding that is more suitable than binary encoding. Our FC process has been validated by means of an empirical evaluation. Using C4.5 and NB on new representations of various datasets, we demonstrated improvement compared with original data features. We have also shown comparable accuracy results w.r.t. efficient state-of-the-art classification techniques. We have now a better understanding of critical issues w.r.t. feature construction when considering closedness related properties. One perspective of this work is to consider our FC process in terms of constraints over sets of patterns and its recent formalization in [27].

Acknowledgments. The authors wish to thank B. Crémilleux for exciting discussions and the data set `meningitis`. They also thank J. Besson for technical support during this study. Finally, this work is partly funded by EU contract IST-FET IQ FP6-516169.

References

1. Li, J., Li, H., Wong, L., Pei, J., Dong, G.: Minimum description length principle: generators are preferable to closed patterns. In: Proceedings AAAI 2006, pp. 409–415. AAAI Press, Menlo Park (2006)
2. Selmaoui, N., Leschi, C., Gay, D., Boulicaut, J.F.: Feature construction and delta-free sets in 0/1 samples. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI), vol. 4265, pp. 363–367. Springer, Heidelberg (2006)

3. Garriga, G.C., Kralj, P., Lavrac, N.: Closed sets for labeled data. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 163–174. Springer, Heidelberg (2006)
4. Cheng, H., Yan, X., Han, J., Hsu, C.W.: Discriminative frequent pattern analysis for effective classification. In: Proceedings IEEE ICDE 2007, pp. 716–725 (2007)
5. Li, J., Liu, G., Wong, L.: Mining statistically important equivalence classes and delta-discriminative emerging patterns. In: Proceedings ACM SIGKDD 2007, pp. 430–439 (2007)
6. Gay, D., Selmaoui, N., Boulicaut, J.F.: Pattern-based decision tree construction. In: Proceedings ICDIM 2007, pp. 291–296. IEEE Computer Society Press, Los Alamitos (2007)
7. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings KDD 1998, pp. 80–86. AAAI Press, Menlo Park (1998)
8. Dong, G., Zhang, X., Wong, L., Li, J.: CAEP: Classification by aggregating emerging patterns. In: Arikawa, S., Furukawa, K. (eds.) DS 1999. LNCS (LNAI), vol. 1721, pp. 30–42. Springer, Heidelberg (1999)
9. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. *Knowledge and Information Systems* 3, 131–145 (2001)
10. Calders, T., Rigotti, C., Boulicaut, J.F.: A survey on condensed representations for frequent sets. In: Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.) Constraint-Based Mining and Inductive Databases. LNCS (LNAI), vol. 3848, pp. 64–80. Springer, Heidelberg (2006)
11. Bastide, Y., Taouil, R., Pasquier, N., Stumme, G., Lakhal, L.: Mining frequent patterns with counting inference. *SIGKDD Explorations* 2, 66–75 (2000)
12. Boulicaut, J.F., Bykowski, A., Rigotti, C.: Approximation of frequency queries by means of free-sets. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 75–85. Springer, Heidelberg (2000)
13. Boulicaut, J.F., 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)
14. Li, W., Han, J., Pei, J.: CMAR: Accurate and efficient classification based on multiple class-association rules. In: Proceedings IEEE ICDM 2001, pp. 369–376 (2001)
15. Yin, X., Han, J.: CPAR: Classification based on predictive association rules. In: Proceedings SIAM SDM 2003 (2003)
16. Boulicaut, J.F., Crémilleux, B.: Simplest rules characterizing classes generated by delta-free sets. In: Proceedings ES 2002, pp. 33–46. Springer, Heidelberg (2002)
17. Baralis, E., Chiusano, S.: Essential classification rule sets. *ACM Trans. on Database Systems* 29, 635–674 (2004)
18. Dong, G., Li, J.: Efficient mining of emerging patterns: discovering trends and differences. In: Proceedings ACM SIGKDD 1999, pp. 43–52 (1999)
19. Bayardo, R.: The hows, whys and whens of constraints in itemset and rule discovery. In: Boulicaut, J.-F., De Raedt, L., Mannila, H. (eds.) Constraint-Based Mining and Inductive Databases. LNCS (LNAI), vol. 3848, pp. 1–13. Springer, Heidelberg (2006)
20. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998)
21. Soulet, A., Crémilleux, B., Rioult, F.: Condensed representation of emerging patterns. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, pp. 127–132. Springer, Heidelberg (2004)

22. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proceedings IJCAI 1993, pp. 1022–1027 (1993)
23. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
24. Meretakakis, D., Wuthrich, B.: Extending naïve bayes classifiers using long itemsets. In: Proceedings ACM SIGKDD 1999, pp. 165–174 (1999)
25. Fan, H., Ramamohanarao, K.: A bayesian approach to use emerging patterns for classification. In: Proceedings ADC 2003, pp. 39–48. Australian Computer Society, Inc. (2003)
26. Fan, H., Ramamohanarao, K.: Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers. *IEEE Trans. on Knowledge and Data Engineering* 18, 721–737 (2006)
27. De Raedt, L., Zimmermann, A.: Constraint-based pattern set mining. In: Proceedings SIAM SDM 2007 (2007)

On Addressing Accuracy Concerns in Privacy Preserving Association Rule Mining

Ling Guo, Songtao Guo, and Xintao Wu

University of North Carolina at Charlotte
{lguo2,sguo,xwu}@uncc.edu

Abstract. Randomized Response techniques have been empirically investigated in privacy preserving association rule mining. In this paper, we investigate the accuracy (in terms of bias and variance of estimates) of both support and confidence estimates of association rules derived from the randomized data. We demonstrate that providing confidence on data mining results from randomized data is significant to data miners. We propose the novel idea of using interquartile range to bound those estimates derived from the randomized market basket data. The performance is evaluated using both representative real and synthetic data sets.

1 Introduction

Privacy is becoming an increasingly important issue in many data mining applications. A considerable amount of work on privacy preserving data mining [2,11,10] has been investigated recently. Among them, randomization has been a primary tool to hide sensitive private data for privacy preserving data mining. The issue of maintaining privacy in association rule mining has attracted considerable attention in recent years [7,8,4,13]. Most of techniques are based on a data perturbation or Randomized Response (RR) approach [5], wherein the 0 or 1 (0 denotes absence of an item while 1 denotes presence of an item) in the original user transaction vector is distorted in a probabilistic manner that is disclosed to data miners.

In [13,4,3], the authors proposed the MASK technique to preserve privacy for frequent itemset mining and addressed the issue of providing efficiency in calculating the estimated support values. Their results empirically showed a high degree of privacy to users and a high level of accuracy in the mining results can be simultaneously achieved. To evaluate the privacy, they defined a privacy metric and presented an analytical formula for evaluating the privacy obtained under the metric. However, accuracy metric on data mining results was only defined in an aggregate manner as support error and identity error computed over all discovered frequent itemsets.

Our paper moves one step further to address the issue of providing accuracy in privacy preserving mining of association rules. We investigate the issue of how the accuracy (i.e., support and confidence) of each association rule mined from randomized data is affected when the randomized response technique is applied.

Specifically, we present an analytical formula for evaluating the accuracy (in terms of bias and variance of estimates) of both support and confidence measures of association rules derived from the randomized data. From the derived bias and variance of estimates, we further derive approximate interquartile ranges. Data miners are ensured

that their estimates lie within these ranges with a high confidence, say 95%. We would emphasize that providing confidence on estimated data mining results is significant to data miners since they can learn how accurate their reconstructed results are. We illustrate the importance of those estimated interquartile ranges using an example.

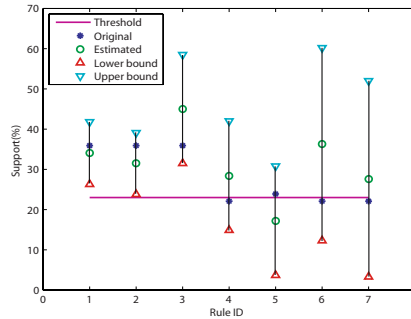


Fig. 1. Accuracy of the estimated support values of association rules derived from randomized data with $p=0.65$

Figure 1 shows the original support values, the estimated support values from the randomized data, and their corresponding 95% interquartile ranges of 7 association rules, which were derived from COIL data sets¹. A distortion parameter $p = 0.65$ and support threshold $sup_{min} = 23\%$ were used in the experiment. The interquartile range of each rule can give data miners confidence about their estimate derived from randomized data. For example, the estimated support of rule 2 is 31.5% and its 95% interquartile range is [23.8%,39.1%], which suggests the original support value lies in this range with 95% probability. Furthermore, we can observe the 95% interquartile ranges for rules 1-3 are above the support threshold, which guarantees those are true frequent itemsets (with at least 95% confidence).

We emphasize providing accuracy of data mining results is important for data miners during data exploration. When the support threshold is set as 23%, we may not only take rule 2 and 6 as frequent sets from the estimated support values, but also conclude rule 6 (35.9%) is more frequent than rule 2 (31.5%). However, rule 2 has the original support as 36.3% while rule 6 has the original support as 22.1%, we mistakenly assign the infrequent itemset 6 as frequent. By using the derived interquartile ranges, we can determine that rule 2 is frequent with high confidence (since its lower bound 23.8% is above the support threshold) and rule 6 may be infrequent (since its lower bound 12.3% is below the support threshold).

The remainder of this paper is organized as follows. In Section 2, we present the distortion framework and discuss how the Randomized Response techniques are applied to privacy preserving market association rule mining. We conduct the theoretical analysis on how distortion process affects the accuracy of both support and confidence values derived from the randomized data in Section 3. In Section 4, empirical evaluations on various datasets are given. We conclude our work in Section 5.

¹ <http://kdd.ics.uci.edu/databases/tic/tic.html>

2 Distortion Framework

2.1 Association Rule Revisited

Denoting the set of transactions in the database D by $\mathcal{T} = \{T_1, \dots, T_n\}$ and the set of items in the database by $\mathcal{I} = \{A_1, \dots, A_m\}$. An association rule $\mathcal{X} \Rightarrow \mathcal{Y}$, where $\mathcal{X}, \mathcal{Y} \subset \mathcal{I}$ and $\mathcal{X} \cap \mathcal{Y} = \phi$, has two measures: the support s defined as the $s(100\%)$ of the transactions in \mathcal{T} contain $\mathcal{X} \cup \mathcal{Y}$, and the confidence c is defined as $c(100\%)$ of the transactions in \mathcal{T} that contain \mathcal{X} also contain \mathcal{Y} .

2.2 Randomization Procedure

Let there be m sensitive items A_1, A_2, \dots, A_m , each being considered as one dichotomous variable with 2 mutually exclusive and exhaustive categories (0 = absence, 1 = presence). One transaction can be logically translated as a fixed-length sequence of 0's and 1's. For each transaction, we apply the Warner RR model [15] independently on each item using different settings of distortion. If the original value is in the *absence*(*presence*) category, it will be kept in such category with a probability θ_0 (θ_1) and changed to *presence*(*absence*) category with a probability $1 - \theta_0$ ($1 - \theta_1$). For item A_j ,

the distortion probability matrix P_j generally takes the form $P_j = \begin{pmatrix} \theta_0 & 1 - \theta_1 \\ 1 - \theta_0 & \theta_1 \end{pmatrix}$.

In this paper, we follow the original Warner RR model by setting $\theta_0 = \theta_1 = p_j$. This setting indicates users have the same level of privacy for both 1's and 0's. In general customers may expect more privacy for their 1's than for their 0's, since the 1's denote specific actions whereas the 0's are the default options.

Denote $\pi^{(j)} = (\pi_0^{(j)}, \pi_1^{(j)})'$ ($\lambda^{(j)} = (\lambda_0^{(j)}, \lambda_1^{(j)})'$) as the vector of marginal proportions corresponding to item A_j in the original (randomized) data set, where $j = 1, \dots, m$. We have

$$\lambda^{(j)} = P_j \pi^{(j)} \quad (1)$$

Note that each vector $\pi^{(j)}$ has two values $\pi_0^{(j)}, \pi_1^{(j)}$ and the latter corresponds to the support value of item A_j . For a market data set with n transactions, let $\hat{\lambda}^{(j)}$ be the vector of sample proportions corresponding to $\lambda^{(j)}$. Then an unbiased estimate of $\pi^{(j)}$ is $\hat{\pi}^{(j)} = P_j^{-1} \hat{\lambda}^{(j)}$.

2.3 Estimating k -Itemset Supports

We can easily extend Equation 1, which is applicable to one individual item, to compute the support of an arbitrary k -itemset. For simplicity, let us assume that we would compute the support of an itemset which contains the first k items $\{A_1, \dots, A_k\}$ (The general case with any k items is quite straightforward but algebraically messy).

Let π_{i_1, \dots, i_k} denote the true proportion corresponding to the categorical combination $(A_{1i_1}, \dots, A_{ki_k})$, where $i_1, \dots, i_k \in \{0, 1\}$. Let π be vectors with elements π_{i_1, \dots, i_k} arranged in a fixed order. The combination vector corresponds to a fixed order of cell entries in the contingency table formed by the k -itemset. When we have k items, the number of cells in the k -dimensional contingency table is 2^k . Table 1(a) shows one

Table 1. 2×2 contingency tables for two variables A,B

<p>(a) Original</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">\bar{B}</td> <td style="padding: 5px; text-align: center;">B</td> <td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px; text-align: center;">\bar{A}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">π_{00}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">π_{01}</td> <td style="padding: 5px; text-align: center;">π_{0+}</td> </tr> <tr> <td style="padding: 5px; text-align: center;">A</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">π_{10}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">π_{11}</td> <td style="padding: 5px; text-align: center;">π_{1+}</td> </tr> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">π_{+0}</td> <td style="padding: 5px; text-align: center;">π_{+1}</td> <td style="padding: 5px; text-align: center;">π_{++}</td> </tr> </table>		\bar{B}	B		\bar{A}	π_{00}	π_{01}	π_{0+}	A	π_{10}	π_{11}	π_{1+}		π_{+0}	π_{+1}	π_{++}	<p>(b) After randomization</p> <table style="margin-left: auto; margin-right: auto; border-collapse: collapse;"> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">\bar{B}</td> <td style="padding: 5px; text-align: center;">B</td> <td style="padding: 5px;"></td> </tr> <tr> <td style="padding: 5px; text-align: center;">\bar{A}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">λ_{00}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">λ_{01}</td> <td style="padding: 5px; text-align: center;">λ_{0+}</td> </tr> <tr> <td style="padding: 5px; text-align: center;">A</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">λ_{10}</td> <td style="padding: 5px; border: 1px solid black; text-align: center;">λ_{11}</td> <td style="padding: 5px; text-align: center;">λ_{1+}</td> </tr> <tr> <td style="padding: 5px;"></td> <td style="padding: 5px; text-align: center;">λ_{+0}</td> <td style="padding: 5px; text-align: center;">λ_{+1}</td> <td style="padding: 5px; text-align: center;">λ_{++}</td> </tr> </table>		\bar{B}	B		\bar{A}	λ_{00}	λ_{01}	λ_{0+}	A	λ_{10}	λ_{11}	λ_{1+}		λ_{+0}	λ_{+1}	λ_{++}
	\bar{B}	B																															
\bar{A}	π_{00}	π_{01}	π_{0+}																														
A	π_{10}	π_{11}	π_{1+}																														
	π_{+0}	π_{+1}	π_{++}																														
	\bar{B}	B																															
\bar{A}	λ_{00}	λ_{01}	λ_{0+}																														
A	λ_{10}	λ_{11}	λ_{1+}																														
	λ_{+0}	λ_{+1}	λ_{++}																														

contingency table for a pair of two variables. We use the notation $\bar{A} (\bar{B})$ to indicate that $A (B)$ is absent from a transaction. The vector $\pi = (\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11})'$ corresponds to a fixed order of cell entries π_{ij} in the 2×2 contingency table. π_{11} denotes the proportion of transactions which contain both A and B while π_{10} denotes the proportion of transactions which contain A but not B . The row sum π_{1+} represents the support frequency of item A while the column sum π_{+1} represents the support frequency of item B .

The original database D is changed to D_{ran} after randomization. Assume $\lambda_{\mu_1, \dots, \mu_k}$ is the probability of getting a response (μ_1, \dots, μ_k) and λ the vector with elements $\lambda_{\mu_1, \dots, \mu_k}$ arranged in a fixed order (e.g., the vector $\lambda = (\lambda_{00}, \lambda_{01}, \lambda_{10}, \lambda_{11})'$ corresponds to cell entries λ_{ij} in the randomized contingency table as shown in Table 1(b)), we can obtain

$$\lambda = (P_1 \times \dots \times P_k)\pi$$

where \times stands for the Kronecker product.

Let $P = P_1 \times \dots \times P_k$, an unbiased estimate of π follows as

$$\hat{\pi} = P^{-1} \hat{\lambda} = (P_1^{-1} \times \dots \times P_k^{-1}) \hat{\lambda} \tag{2}$$

where $\hat{\lambda}$ is the vector of sample proportions corresponding to λ and P_j^{-1} denotes the inverse of the matrix P_j . Note that although the distortion matrices P_1, \dots, P_k are known, they can only be utilized to estimate the proportions of itemsets of the original data, rather than precisely reconstruct the original 0-1 data.

In this paper we follow the Moment Estimation method as shown in Equation 2 to get the unbiased estimate of the distribution for original data. This method has been broadly adopted in the scenarios where RR is used to perturb data for preserving privacy. Although it has good properties as computational simplicity and unbiasedness, some awkward property exists due to random errors [5][6]. That is, the estimate may fall out of the parameter space, which makes the estimate meaningless. This is one reason that Maximum Likelihood Estimation (MLE) is adopted to estimate the distribution in literature [6].

It has been proved in [6] that a good relation holds between these two methods in the scenarios of RR: The moment estimate is equal to the MLE estimate within parameter space. Based on that, we can know that moment estimate from Equation 2 achieves the Cramér-Rao bound as MLE does. Therefore, moment estimate is the minimum variance unbiased (MVU) estimator in RR contexts. Our later analysis on accuracy of association rule is based on such unbiased estimate under the assumption that the estimate is within parameter space.

3 Theoretical Analysis on Accuracy of Association Rule

In this section, we theoretically analyze the variance of the estimates of both s and c for any individual association rule $\mathcal{X} \Rightarrow \mathcal{Y}$. To derive their interquantile ranges, we also analyze the distributions of those estimates derived from the randomized data.

3.1 Accuracy on Support s

From Equation 2, we know how to derive the estimate of support values of any itemset from the observed randomized data. Now we address the question how accurate the estimated support value is.

The whole contingency table is usually modeled as a multinomial distribution in statistics. When we have k items, the number of cells in the contingency table is 2^k . For each cell d , where $d = 1, 2, \dots, 2^k$, it has a separate binomial distribution with parameters n and η_i . The binomial distribution is the discrete probability distribution of the number of successes in a sequence of n independent 0/1 experiments, each of which yields success with probability η_i . When n is large enough (one rule of thumb is that both $n\eta_i$ and $n(1 - \eta_i)$ must be greater than 5), an approximation to $B(n, \eta_i)$ is given by the normal distribution $N(n\eta_i, n\eta_i(1 - \eta_i))$.

Result 1. *Since each cell π_{i_1, \dots, i_k} approximately follows normal distribution, its $(1 - \alpha)100\%$ interquantile range can be approximated as*

$$[\hat{\pi}_{i_1 \dots i_k} - z_{\alpha/2} * \sqrt{\hat{\text{var}}(\hat{\pi}_{i_1 \dots i_k})}, \hat{\pi}_{i_1 \dots i_k} + z_{\alpha/2} * \sqrt{\hat{\text{var}}(\hat{\pi}_{i_1 \dots i_k})}]$$

$z_{\alpha/2}$ is the upper $\alpha/2$ critical value for the standard normal distribution.

$\hat{\text{var}}(\hat{\pi}_{i_1 \dots i_k})$ can be derived from the covariance matrix [5]:

$$\begin{aligned} \text{cov}(\hat{\pi}) &= \Sigma_1 + \Sigma_2 \\ &= (n - 1)^{-1}(\hat{\pi}^\delta - \hat{\pi}\hat{\pi}') + (n - 1)^{-1}P^{-1}(\hat{\lambda}^\delta - P\hat{\pi}^\delta P')P'^{-1} \end{aligned}$$

Note that Σ_1 is the dispersion matrix of the direct estimator of π , which is only related to the data size for estimation. While the data size is usually large in most market basket analysis scenarios, it can be neglected. Σ_2 represents the component of dispersion associated with RR distortion.

We can simply use the derived $\hat{\pi}_{i_1 \dots i_m}$ (from Equation 2) as an estimate of μ and the derived $\sqrt{\hat{\text{var}}(\hat{\pi}_{i_1 \dots i_m})}$ as an estimate of σ , where μ and σ are unknown parameters of the normal distribution of each cell. An $(1 - \alpha)100\%$ interquantile range, say $\alpha = 0.05$, shows the interval contains the original π_{i_1, \dots, i_m} with 95% probability.

To illustrate this result, we use a simple example $G \Rightarrow H$ (rule 2 in Figure 1). The proportion of itemsets of the original data is given as

$$\pi = (\pi_{00}, \pi_{01}, \pi_{10}, \pi_{11})' = (0.415, 0.043, 0.183, 0.359)'$$

Using the RR scheme presented in the previous section, with the distortion parameters $p_1 = p_2 = 0.9$, we get the randomized responses

$$\hat{\lambda} = (0.368, 0.097, 0.218, 0.316)'$$

By applying Equation 2 we derive the unbiased estimate of π as

$$\hat{\pi} = (0.427, 0.031, 0.181, 0.362)'$$

The covariance matrix of $\hat{\pi}$ is unbiasedly estimated as

$$\hat{cov}(\hat{\pi}) = \begin{bmatrix} 7.113 & -1.668 & -3.134 & -2.311 \\ -1.668 & 2.902 & 0.244 & -1.478 \\ -3.134 & 0.244 & 5.667 & -2.777 \\ -2.311 & -1.478 & -2.777 & 6.566 \end{bmatrix} \times 10^{-5}$$

The diagonal elements of the above matrix represent the variances of the estimated $\hat{\pi}$, e.g., $\hat{var}(\hat{\pi}_{00}) = 7.113 \times 10^{-5}$ and $\hat{var}(\hat{\pi}_{11}) = 6.566 \times 10^{-5}$. Those off-diagonal elements indicate the estimated covariances, e.g., $\hat{cov}(\hat{\pi}_{11}, \hat{\pi}_{10}) = -2.777 \times 10^{-5}$.

From Result 1 we can derive 95% interquartile range of s_{GH} as

$$[\hat{\pi}_{11} - z_{0.025} \sqrt{\hat{var}(\hat{\pi}_{11})}, \hat{\pi}_{11} + z_{0.025} \sqrt{\hat{var}(\hat{\pi}_{11})}] = [0.346, 0.378]$$

We can also see this derived interquartile range [0.346, 0.378] for rule 2 with $p_1 = p_2 = 0.9$ is shorter than [0.238, 0.391] with $p_1 = p_2 = 0.65$ as shown in Figure 1.

3.2 Accuracy on Confidence c

We first analyze the accuracy on confidence of a simple association rule $A \Rightarrow B$ where A and B are two single items which have 2 mutually exclusive and exhaustive categories. We denote s_A, s_B , and s_{AB} as the support values of A, B , and AB respectively. Accordingly, we denote \hat{s}_A, \hat{s}_B , and \hat{s}_{AB} as the estimated support values from randomized data of A, B , and AB respectively.

Result 2. *The confidence (c) of a simple association rule $A \Rightarrow B$ has estimated value as*

$$\hat{c} = \frac{\hat{s}_{AB}}{\hat{s}_A} = \frac{\hat{\pi}_{11}}{\hat{\pi}_{1+}}$$

with the expectation of \hat{c} approximated as

$$\hat{E}(\hat{c}) \approx \frac{\hat{\pi}_{11}}{\hat{\pi}_{1+}} + \frac{\hat{\pi}_{11}}{\hat{\pi}_{1+}^3} \hat{var}(\hat{\pi}_{10}) - \frac{\hat{\pi}_{10}}{\hat{\pi}_{1+}^3} \hat{var}(\hat{\pi}_{11}) + \frac{\hat{\pi}_{11} - \hat{\pi}_{10}}{\hat{\pi}_{1+}^3} \hat{cov}(\hat{\pi}_{11}, \hat{\pi}_{10}) \quad (3)$$

and the variance of \hat{c} approximated as

$$\hat{var}(\hat{c}) \approx \frac{\hat{\pi}_{10}^2}{\hat{\pi}_{1+}^4} \hat{var}(\hat{\pi}_{11}) + \frac{\hat{\pi}_{11}^2}{\hat{\pi}_{1+}^4} \hat{var}(\hat{\pi}_{10}) - 2 \frac{\hat{\pi}_{10} \hat{\pi}_{11}}{\hat{\pi}_{1+}^4} \hat{cov}(\hat{\pi}_{11}, \hat{\pi}_{10}) \quad (4)$$

according to the delta method [12].

Confidence can be regarded as a ratio (W) of two correlated normal random variables (X, Y), $W = X/Y$. However, it is hard to derive the critical value for the distribution of W from its cumulative density function $F(w)$ [14], we provide an approximate interquartile range of confidence based on Chebyshev's Inequality.

Theorem 1. (*Chebyshev's Inequality*) For any random variable X with mean μ and variance σ^2

$$Pr(|X - \mu| \geq k\sigma) \leq 1/k^2 \quad k > 0$$

Chebyshev's Inequality gives a conservative estimate. It provides a lower bound to the proportion of measurements that are within a certain number of standard deviations from the mean.

Result 3. *The loose $(1 - \alpha)100\%$ interquantile range of confidence (c) of $A \Rightarrow B$ can be approximated as*

$$[\hat{E}(\hat{c}) - \frac{1}{\sqrt{\alpha}}\sqrt{\hat{var}(\hat{c})}, \hat{E}(\hat{c}) + \frac{1}{\sqrt{\alpha}}\sqrt{\hat{var}(\hat{c})}]$$

From Chebyshev's Inequality, we know for any sample, at least $(1 - 1/k^2)$ of the observations in the data set fall within k standard deviations of the mean. When we set $\alpha = \frac{1}{k^2}$, we have $Pr(|X - \mu| \geq \frac{1}{\sqrt{\alpha}}\sigma) \leq \alpha$. Hence, $Pr(|X - \mu| \leq \frac{1}{\sqrt{\alpha}}\sigma) \geq 1 - \alpha$. We can simply use the derived $\hat{E}(\hat{c})$ (from Equation 3) as an estimate of μ and the derived $\sqrt{\hat{var}(\hat{c})}$ (from Equation 4) as an estimate of σ , where μ and σ are unknown parameters of the distribution of confidence. An approximate $(1 - \alpha)100\%$ interquantile range of confidence c is then derived.

All the above results can be straightforwardly extended to the general association rule $\mathcal{X} \Rightarrow \mathcal{Y}$ and further details can be found in [9].

4 Empirical Evaluation

In our experiments, we use the COIL Challenge 2000 which provides data from a real insurance business. Information about customers consists of 86 attributes and includes product usage data and socio-demographic data derived from zip area codes. The training set consists of 5822 descriptions of customers, including the information of whether or not they have a Caravan insurance policy. Our binary data is formed by collapsing non-binary categorical attributes into binary form (the data can be found at www.cs.uncc.edu/~xwu/classify/b86.dat), with $n = 5822$ baskets and $m = 86$ binary items.

4.1 Accuracy of Individual Rule vs. Varying p

Table 2² shows the 7 randomly chosen association rules derived from the randomized COIL data with distortion parameter $p = 0.65$. In this table, s (\hat{s}) indicates the original (estimated) support value. \hat{s}_l (\hat{s}_u) denotes the lower bound (upper bound) of the 95% interquantile range of the estimated support value. Similarly, c (\hat{c}) indicates the original (estimated) confidence value. \hat{c}_l (\hat{c}_u) denotes the lower bound (upper bound) of the 95% estimated confidence value. We have shown how the accuracy of the estimated support values varies in Figure 1 (Section 1). One observation is that interquantile ranges of

² The meaning of these items can be found in Table 2 of [16].

Table 2. Accuracy of the estimated support and confidence for 7 representative rules of COIL

ID	\mathcal{X}	\mathcal{Y}	s	\hat{s}	\hat{s}_l	\hat{s}_u	c	\hat{c}	\hat{c}_l	\hat{c}_u
1	G	E	35.9	34.1	26.3	41.8	66.2	64.7	31.3	95.3
2	G	H	35.9	31.5	23.8	39.1	66.2	62.2	26.6	90.4
3	EH	G	35.8	45.0	31.5	58.5	89.3	77.5	33.5	100
4	EG	I	22.1	28.4	14.9	42.0	61.7	75.2	0	100
5	HF	I	23.9	17.2	3.7	30.8	100	91.0	0	100
6	EGH	F	22.1	36.3	12.3	60.2	61.7	99.4	0	100
7	FGI	E	22.1	27.6	3.32	52.0	77.9	86.3	0	100

confidence estimates are usually wider than that of support estimates. For example, the 95% interquartile range of the estimated confidence for rule 2 is [26.6%, 90.4%], which is much wider than that of the estimated support [23.8%, 39.1%]. This is due to three reasons. First, we set the distortion parameter $p = 0.65$ which implies a relatively large noise (the perturbed data will be completely random when $p = 0.5$). Second, the variance of the ratio of two variables is usually larger than the variance of either single variable. Third, the estimated support can be modeled as one approximate normal distribution so we can use the tight interquartile range. On the contrary, we derive the loose interquartile range of confidence using the general Chebyshev’s Theorem. We expect that the explicit form of the $F(w)$ distribution can significantly reduce this width. We will investigate the explicit form of the distribution of confidence and all other measures, e.g. correlation, lift, etc. to derive tight bounds in our future work.

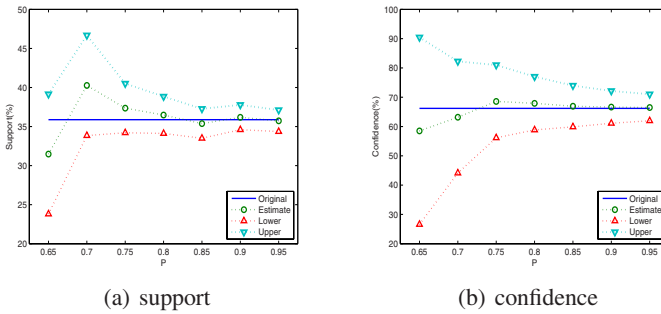


Fig. 2. Accuracy vs. varying p for rule $G \Rightarrow H$

Our next experiment shows how the derived estimates (support, confidence, and their corresponding interquartile ranges) of one individual rule vary with the distortion parameter p . We vary the distortion parameter p from 0.65 to 0.95. Figure 2(a) 2(b) shows the accuracy of the estimated support (confidence) values with varied distortion p values for a particular rule $G \Rightarrow H$. As expected, the larger the p , the more accurate the estimate and the tighter the interquartile range is. It was empirically shown in

[13] that a distortion probability of $p = 0.9$ (equivalently $p = 0.1$) is ideally suited to provide both privacy and good data mining results for the sparse market basket data. We can observe from Figure 2(b) that the 95% interquartile range of the confidence estimate with $p \geq 0.9$ is tight.

4.2 Accuracy of All Rules vs. Varying p

The above study of the accuracy of the estimate in terms of each individual rule is based on the variance as criterion. In the case of all rules together, we can evaluate the overall accuracy of data mining results using the average support error, the average confidence error, percentage of false positives, percentage of false negatives etc. as defined in [4].

The metric $\rho = \frac{1}{|R|} \sum_{r \in R} \frac{|\hat{s}_r - s_r|}{s_r} \times 100$ represents the average relative error in the reconstructed support values for those rules that are correctly identified. The identity error σ reflects the percentage error in identifying association rules. $\sigma^+ = \frac{|R-F|}{|F|} \times 100$ indicates the percentage of false positives and $\sigma^- = \frac{|F-R|}{|F|} \times 100$ indicates the percentage of false negatives where R (F) denotes the reconstructed (actual) set of association rules. In addition to the support error (ρ) and the identity error (σ^+ , σ^-), we define the following three measures.

- γ : the confidence error $\gamma = \frac{1}{|R|} \sum_{r \in R} \frac{|\hat{c}_r - c_r|}{c_r} \times 100$ represents the average relative error in the reconstructed confidence values for those rules that are correctly identified.
- s-p: the number of pairs of conflict support estimates. We consider \hat{s}_1, \hat{s}_2 as a pair of conflict estimates if $\hat{s}_1 < \hat{s}_2$ but $s_1 > \hat{s}_{1l} > s_{min} > s_2$ where \hat{s}_{1l} denotes the lower bound of interquartile range for s_1 .
- c-p: the number of pairs of conflict confidence estimates (similarly defined as the above s-p).

Errors in support estimation due to the distortion procedure can result in falsely identified frequent itemsets. This becomes especially an issue when the support threshold setting is such that the support of a number of frequent itemsets lie very close to this threshold value (s_{min}). Such border-line itemsets can cause many *false positives* and *false negatives*. Even worse, an error in identifying a frequent itemset correctly in early passes has a ripple effect in terms of causing errors in later passes.

Table 3(a) shows how the above measures are varied by changing distortion parameter p from 0.65 to 0.95. We can observe all measures (the support error ρ , the confidence error γ , the false positives σ^+ , the false negatives σ^-) decrease when p increases. The number of conflict support pairs (s-p) and conflict confidence pairs (c-p) also have the same trend. Our experiment shows that when $p \geq 0.85$, there are no or very few conflict support (confidence) pairs, which implies the reconstructed set of association rules is close to the original set. However, when $p \leq 0.80$, there are significant number of conflict pairs, which implies the reconstructed set may be quite different from the original one. By incorporating the derived interquartile range for each estimate, we can decrease the error caused by conflict pairs. In Section 1, we have shown one conflict support pair: rule 2 and rule 6. We can see that $\hat{s}_2 < \hat{s}_6$ (but $s_2 > s_6$). As $\hat{s}_{2l} > s_{min}$ and

Table 3. $sup_{min} = 25\%$ $conf_{min} = 65\%$ for COIL

(a)							(b)						
p	ρ	σ^-	σ^+	s-p	γ	c-p	p	σ^-	σ_l^-	σ_u^-	σ^+	σ_l^+	σ_u^+
0.65	25.6	34.0	53.8	27817	9.90	737	0.65	34.0	98.8	1.25	53.8	0.00	110.7
0.70	12.3	21.2	38.1	4803	6.39	393	0.70	21.2	90.9	0.08	38.1	0.08	105.7
0.75	7.35	11.8	30.8	729	4.44	85	0.75	11.8	66.3	0.00	30.8	1.18	96.5
0.80	3.64	6.82	16.9	0	2.47	28	0.80	6.82	50.7	0.31	16.9	0.24	80.9
0.85	2.64	6.67	7.76	0	1.76	0	0.85	6.67	37.7	0.00	7.76	0.55	53.0
0.90	1.91	5.18	4.24	0	1.10	0	0.90	5.18	31.8	0.00	4.24	0.00	35.0
0.95	0.84	4.63	1.02	0	0.51	0	0.95	4.63	26.8	0.00	1.02	0.00	25.7

$\hat{s}_{6l} < s_{min}$, data miners can safely determine rule 2 is frequent but rule 6 may be infrequent. We would emphasize again that providing estimates together with their interquartile ranges (especially for those conflict pairs) through some visualization is very useful for data exploration tasks conducted on the randomized data.

Table 3(b) shows the comparison between the identity errors derived using lower bound and upper bound respectively. We define $\sigma_l^+ = \frac{|R_l - F|}{|F|} \times 100$ ($\sigma_u^+ = \frac{|R_u - F|}{|F|} \times 100$) as the false positives calculated from R_l (R_u) where R_l (R_u) denotes the reconstructed set of association rules using lower (upper) bound of interquartile range respectively. Similarly we define σ_l^- and σ_u^- . We can observe from Table 3(b) that σ_u^- is significantly lower than σ_- while σ_l^+ is significantly lower than σ_+ . In other words, using the upper bound of the derived interquartile range can decrease the false negatives while using the lower bound can decrease the false positives. In some scenario, we may emphasize more on decreasing the false positive error. Hence, we can use the lower bound of the derived interquartile range, rather than the estimated value, to determine whether the set is frequent or not (i.e., frequent only if $\hat{s}_l \geq s_{min}$, infrequent otherwise).

4.3 Other Datasets

Since the COIL Challenge data is very sparse (5822 tuples with 86 attributes), we also conducted evaluations on the following representative databases used for association rule mining.

1. BMS-WebView-1³. Each transaction in the data set is a web session consisting of all the product detail pages viewed in that session. There are about 60,000 transactions with close 500 items.
2. A synthetic database generated from the IBM Almaden market basket data generator with parameters T10.I4.D0.1M.N0.1K., resulting in 10k customer tuples with each customer purchasing about ten items on average.

Tables 4 and 5 show our results on these two data sets respectively. We can observe similar patterns as shown in COIL data set.

³ <http://www.ecn.purdue.edu/KDDCUP>

Table 4. $sup_{min} = 0.20\%$ $conf_{min} = 20\%$ for BMS-WebView-1

(a)

p	ρ	σ^-	σ^+	s-p	γ	c-p
0.65	362.4	64.1	80.6	632	114.7	11
0.75	72.9	39.9	68.7	418	57.9	2
0.85	19.5	27.9	54.0	67	24.5	0
0.95	5.47	9.66	16.5	56	7.23	0

(b)

p	σ^-	σ_l^-	σ_u^-	σ^+	σ_l^+	σ_u^+
0.65	63.9	100.0	1.34	81.8	0.0	187.6
0.75	40.1	100.0	1.07	69.8	0.0	155.3
0.85	27.9	99.1	0.40	54.0	0.0	152.8
0.95	9.66	70.6	0.00	16.5	0.0	123.8

Table 5. $sup_{min} = 0.20\%$ $conf_{min} = 60\%$ for IBM data with T10.I4.D0.1M.N0.1K

(a)

p	ρ	σ^-	σ^+	s-p	γ	c-p
0.65	1234.9	73.4	171.9	971	47.8	7
0.75	99.7	57.8	168.0	11	38.3	0
0.85	19.9	49.7	165.6	3	18.6	0
0.95	5.14	21.3	50.3	0	4.61	0

(b)

p	σ^-	σ_l^-	σ_u^-	σ^+	σ_l^+	σ_u^+
0.65	73.7	100.0	2.99	172.8	0.0	722.5
0.75	57.8	100.0	1.20	167.9	0.0	674.3
0.85	49.7	100.0	0.90	165.6	0.0	673.4
0.95	21.3	99.7	0.00	50.3	0.0	460.8

5 Conclusion and Future Work

In this paper, we have considered the issue of providing confidence ranges of support and confidence in privacy preserving association rule mining. Providing the accuracy of discovered patterns from randomized data is important for data miners. To the best of our knowledge, this has not been previously explored in the context of privacy preserving data mining.

Randomization still runs certain risk of disclosures. It was observed as a general phenomenon that maintenance of item privacy and precise estimation were in conflict. We will investigate how to determine distortion parameters optimally to satisfy both privacy and accuracy constraints. We will explore some scenario where some sensitive items are randomized while the remaining are released directly or where some transactions are randomized while the remaining are unperturbed. We also plan to investigate the extension of our results to generalized and quantitative association rules.

Acknowledgments

This work was supported in part by U.S. National Science Foundation IIS-0546027.

References

1. Agrawal, D., Agrawal, C.: On the design and quantification of privacy preserving data mining algorithms. In: Proceedings of the 20th Symposium on Principles of Database Systems (2001)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, Texas, May 2000, pp. 439–450 (2000)

3. Agrawal, S., Haritsa, J.: A framework for high-accuracy privacy-preserving mining. In: Proceedings of the 21st IEEE International Conference on Data Engineering, pp. 193–204 (2005)
4. Agrawal, S., Krishnan, V., Haritsa, J.: On addressing efficiency concerns in privacy-preserving mining. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 113–124. Springer, Heidelberg (2004)
5. Chaudhuri, A., Mukerjee, R.: Randomized Response Theory and Techniques. Marcel Dekker, New York (1988)
6. den Hout, A.V., der Heijden, P.G.M.V.: Randomized response, statistical disclosure control and misclassification: A review. *International Statistical Review* 70(2), 269–288 (2002)
7. Evfimievski, A., Gehrke, J., Srikant, R.: Limiting privacy breaches in privacy preserving data mining. In: Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pp. 211–222 (2003)
8. Evfimievski, A., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 217–228 (2002)
9. Guo, L., Guo, S., Wu, X.: On addressing accuracy concerns in privacy preserving association rule mining. Technical Report, CS Dept., UNC Charlotte (March 2007)
10. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: Proceedings of the ACM SIGMOD Conference on Management of Data, Baltimore, MA (2005)
11. Kargupta, H., Datta, S., Wang, Q., Sivakumar, K.: On the privacy preserving properties of random data perturbation techniques. In: Proceedings of the 3rd International Conference on Data Mining, pp. 99–106 (2003)
12. Kendall, M.G., Stuart, A.: The advanced theory of statistics, vol. 1. Hafner Pub. Co., New York (1969)
13. Rizvi, S., Haritsa, J.: Maintaining data privacy in association rule mining. In: Proceedings of the 28th International Conference on Very Large Data Bases (2002)
14. Springer, M.D.: The Algebra of Random Variables. John Wiley and Sons, New York (1979)
15. Warner, S.: Randomized response: a survey technique for eliminating evasive answer bias. *J. Amer. Statist. Assoc.* 60, 63–69 (1965)
16. Wu, X., Barbará, D., Ye, Y.: Screening and interpreting multi-item associations based on log-linear modeling. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washinton, August 2003, pp. 276–285 (2003)

Privacy-Preserving Linear Fisher Discriminant Analysis

Shuguo Han and Wee Keong Ng

School of Computer Engineering, Nanyang Technological University, Singapore
{hans0004,awkng}@ntu.edu.sg

Abstract. Privacy-preserving data mining enables two or more parties to collaboratively perform data mining while preserving the data privacy of the participating parties. So far, various data mining and machine learning algorithms have been enhanced to incorporate privacy preservation. In this paper, we propose privacy-preserving solutions for Fisher Discriminant Analysis (FDA) over horizontally and vertically partitioned data. FDA is one of the widely used discriminant algorithms that seeks to separate different classes as much as possible for discriminant analysis or dimension reduction. It has been applied to face recognition, speech recognition, and handwriting recognition. The secure solutions are designed based on two basic secure building blocks that we have proposed—the Secure Matrix Multiplication protocol and the Secure Inverse of Matrix Sum protocol—which are in turn based on cryptographic techniques. We conducted experiments to evaluate the scalability of the proposed secure building blocks and overheads to achieve privacy when performing FDA.

1 Introduction

Data mining is a powerful tool to discover interesting, useful, and even hidden patterns that has been applied to various domain such as business intelligence, bioinformatics, and homeland security. While conventional data mining assumes that the data miner has full access rights to data that are collected from different sources or that are distributed among multiple parties, privacy or security issues render this assumption infeasible when the parties cannot be fully trusted, as some parties may have malicious intent. How to collaboratively perform data mining without compromising the data privacy of the participating parties has become an interesting topic of research in the data mining community.

Privacy-preserving data mining (PPDM) is a response from the data mining community to address data privacy issues. Approaches in PPDM are generally based on Secure Multi-party Computations (SMC) [12] and/or randomization techniques [1]. The former uses specialized, proven protocols to achieve various types of computation without losing data privacy. The latter introduces noise to the original private data to achieve security but lose accuracy. As the former approach achieves a higher degree of accuracy, we focus on SMC in this paper. To

date, various data mining algorithms have been enhanced to incorporate privacy preservation based on SMC techniques.

In machine learning and data mining, Fisher Discriminant Analysis (FDA) is one of the widely used discriminant algorithms that seeks to find directions so that data in the same classes are projected near to each other while ones in different classes are projected as far as possible for classification or dimension reduction. It has wide applications in face recognition [13], speech recognition [11], and digit recognition [2]. In this paper, we enhance Fisher Discriminant Analysis to incorporate the privacy-preserving feature. To the best of our knowledge, there has not been any work that extends privacy preservation to FDA.

Our contributions in this paper are summarized as follows:

1. We propose two protocols—the Secure Matrix Multiplication protocol and the Secure Inverse of Matrix Sum protocol—as secure basic building blocks for privacy-preserving FDA. The underlying algorithms of these protocols are novel and more secure than those by Du et al. [3].
2. Based on the two secure building blocks, we propose protocol for privacy-preserving FDA over horizontally and vertically partitioned data.

We have evaluated the computational complexity and scalability of the proposed protocols both analytically and empirically and show that the protocols are efficient and scalable for small to medium size data. We also addressed some specific implementation issues such as methods to handle real numbers and negative numbers in cryptography. We believe this work is significant as it serves as a guide to the investigation of extending data privacy preservation to related methods such as Principal Component Analysis, Independent Component Analysis, and so on.

The organization of this paper is as follows: In Section 2, we present an overview of background knowledge about linear FDA and related work. Section 3 proposes two secure building blocks of matrix computation. We also present protocols for Privacy-Preserving FDA (PPFDA) over horizontally partitioned data and vertically partitioned data in Section 4. In Section 5, we perform experiments to evaluate the proposed secure building blocks and protocols. Finally, Section 6 concludes the paper.

2 Background and Related Work

2.1 Linear Fisher Discriminant Analysis

Fisher Discriminant Analysis (FDA) as introduced by Fisher [5] seeks to separate different classes as much as possible using some criterion function (Eq. 1). As the technique of applying FDA on a two-class dataset is used repeatedly for the analysis of any pairs of data in a multi-class dataset, we focus on the two-class problem using FDA in this paper. It is non-trivial to extend the two-class problem approach to multi-class problems. This will be part of our future work. This section provides an overview of background knowledge about linear FDA.

We present the conventional mathematical model of linear FDA for two-class data [4]. Suppose we have a set of two-class n data samples of d dimensions: $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ where $\mathbf{x}_i = [x_{1,i}, x_{2,i}, \dots, x_{d,i}]^T$ such that n_1 samples are in the subset $\varphi_1 = \{\mathbf{x}_1^1, \mathbf{x}_2^1, \dots, \mathbf{x}_{n_1}^1\}$ and n_2 samples are in the subset $\varphi_2 = \{\mathbf{x}_1^2, \mathbf{x}_2^2, \dots, \mathbf{x}_{n_2}^2\}$, $n_1 + n_2 = n$. Assuming that column vector \mathbf{w} is the direction of the projection from \mathbf{X} to $\mathbf{y} = \{y_1, y_2, \dots, y_n\}$, we have $\mathbf{y} = \mathbf{w}^T \mathbf{X}$. The d -dimensional sample mean \mathbf{m}_i for class i is $\mathbf{m}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} \mathbf{x}_j^i$.

Fisher Discriminant Analysis aims to maximize between-class separability and minimize within-class variability. Formally, the criterion function in Eq. 1 is to be maximized for the function $\mathbf{w}^T \mathbf{X}$:

$$J(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_B \mathbf{w}}{\mathbf{w}^T \mathbf{S}_W \mathbf{w}} \tag{1}$$

where

$$\begin{aligned} \mathbf{S}_B &= (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^T \\ \mathbf{S}_W &= \sum_{i=1,2} \sum_{j=1}^{n_i} (\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T \end{aligned} \tag{2}$$

is the

The objective of FDA is to find a projection vector \mathbf{w} such that $J(\mathbf{w})$ in Eq. 1 is a maximum. The solution for such \mathbf{w} can be obtained by differentiating $J(\mathbf{w})$ with respect to \mathbf{w} yielding

$$\mathbf{w} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{3}$$

We note that only the direction, not the length of \mathbf{w} , is important.

To incorporate the privacy-preserving feature to linear FDA, the challenge is to securely compute \mathbf{S}_W^{-1} and $\mathbf{m}_1 - \mathbf{m}_2$ so that \mathbf{w} can be securely computed. Clearly, what we need is a method to perform matrix multiplication and matrix inverse securely. In Section 4, we propose a secure approach to address the problem.

2.2 Secure Building Blocks

Various data mining algorithms have been enhanced to incorporate privacy preservation, including classification using decision tree [12], association rule mining [16], clustering using k -means [10], and so on. Recently, the approach has been extended to several machine learning algorithms such as linear regression [3], gradient descent methods [17], self-organizing maps [8], and genetic algorithms [7]. Many of these privacy-enabled algorithms rely on secure building blocks to enforce privacy. Secure building blocks are basic common operations that underly many algorithms. Examples include secure sum, secure comparison, secure scalar product, secure matrix multiplication, and so on.

Fisher Discriminant Analysis—the focus of this paper—requires two secure building blocks: Secure matrix multiplication and secure inverse of matrix sum.

Du et al. [3] has proposed a secure protocol for secure matrix multiplication using linear algebraic methods. It uses a random and invertible matrix to disguise the original matrices to achieve privacy. For security, a concept called “ k -secure” was introduced to generate the random matrix. Assuming that Party B wants to attack private matrix \mathbf{A} of Party A, a k -secure matrix \mathbf{M} (jointly generated by both parties) means that (1) any equation from \mathbf{MA} includes at least $k + 1$ unknown elements of \mathbf{A} , and (2) any k combined equations include at least $2k$ unknown elements of \mathbf{A} . Therefore, it is impossible to know any elements of matrix \mathbf{A} as there are infinite possible solutions due to insufficient equations.

An issue with Du’s approach is that constructing such a matrix is a complex process [3]. More importantly, Du’s approach may have a security problem. If Party A and the same Party B or different Party Bs (a group of colluding parties) perform secure matrix multiplication more than once, more \mathbf{M} s (more equations) are available for attacking the fixed unknown elements matrix \mathbf{A} . In response to this problem, we propose another more secure and efficient protocol for matrix multiplication in this paper.

3 Secure Building Blocks

In this section, we propose the Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol to support the secure computation of Eq. 3 which we have identified to be the key to incorporating privacy preservation in FDA. Our proposed protocols are based on cryptographic techniques and are improvements over existing protocols [3] for secure matrix multiplication and inverse of matrix sum.

3.1 Secure Matrix Multiplication

Parties A and B each hold private $d \times N$ matrix \mathbf{A} and private $N \times n$ matrix \mathbf{B} respectively. They want to securely compute matrix multiplication so that at the end of the computation, party A and B each only holds a portion of the product matrix \mathbf{M}^a and \mathbf{M}^b respectively such that their matrix sum $\mathbf{M}^a + \mathbf{M}^b = \mathbf{AB}$ is the desired product matrix, which is unknown to both parties.

Given any $m \times n$ matrix \mathbf{H} , its i th row vector $\mathbf{h}(i, :) = (h_{i,1}, h_{i,2}, \dots, h_{i,n})$ and j th column vector $\mathbf{h}(:, j) = (h_{1,j}, h_{2,j}, \dots, h_{m,j})$. By definition of matrix multiplication $\mathbf{M}=\mathbf{AB}$, we have

$$\mathbf{M} = \mathbf{AB} = \begin{bmatrix} \mathbf{a}(1, :) \cdot \mathbf{b}(:, 1) & \mathbf{a}(1, :) \cdot \mathbf{b}(:, 2) & \cdots & \mathbf{a}(1, :) \cdot \mathbf{b}(:, n) \\ \mathbf{a}(2, :) \cdot \mathbf{b}(:, 1) & \mathbf{a}(2, :) \cdot \mathbf{b}(:, 2) & \cdots & \mathbf{a}(2, :) \cdot \mathbf{b}(:, n) \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{a}(d, :) \cdot \mathbf{b}(:, 1) & \mathbf{a}(d, :) \cdot \mathbf{b}(:, 2) & \cdots & \mathbf{a}(d, :) \cdot \mathbf{b}(:, n) \end{bmatrix}$$

Clearly, each element of \mathbf{M} above is a scalar product of two vectors. To securely perform the matrix multiplication \mathbf{AB} , we may apply the Secure Scalar Product

Protocol 1. Secure Matrix Multiplication Protocol

Input: Party A has private $d \times N$ matrix \mathbf{A} and Party B has private $N \times n$ matrix \mathbf{B} .

Output: Party A obtains private matrix \mathbf{M}^a and Party B obtains private matrix \mathbf{M}^b such that their sum $\mathbf{M}^a + \mathbf{M}^b = \mathbf{AB}$ yields the product matrix.

- 1: **for** $i = 1$ to d **do**
 - 2: **for** $j = 1$ to n **do**
 - 3: Party A and Party B securely compute the scalar product of vector $\mathbf{a}(i, :)$ and vector $\mathbf{b}(:, j)$. At the end, Party A and Party B each hold a private value $m_{i,j}^a$ and $m_{i,j}^b$ respectively. Part A designates $\mathbf{M}_{i,j}^a = m_{i,j}^a$ and Party B designates $\mathbf{M}_{i,j}^b = m_{i,j}^b$.
 - 4: **end for**
 - 5: **end for**
-

protocol [6] so that each scalar product is the sum of two portions as follows:

$$\mathbf{AB} = \begin{bmatrix} m_{1,1}^a + m_{1,1}^b & m_{1,2}^a + m_{1,2}^b & \cdots & m_{1,n}^a + m_{1,n}^b \\ m_{2,1}^a + m_{2,1}^b & m_{2,2}^a + m_{2,2}^b & \cdots & m_{2,n}^a + m_{2,n}^b \\ \vdots & \vdots & \ddots & \vdots \\ m_{d,1}^a + m_{d,1}^b & m_{d,2}^a + m_{d,2}^b & \cdots & m_{d,n}^a + m_{d,n}^b \end{bmatrix} = \mathbf{M}^a + \mathbf{M}^b$$

In this way, we securely obtain the matrix multiplication (which is unknown to both parties) as the sum of two private portions \mathbf{M}^a and \mathbf{M}^b held by Party A and B respectively. The details are shown in Protocol [1].

This method is more straightforward and less complex than the secure matrix multiplication protocol by Du et al. [3]. Moreover, the execution of secure scalar product of each matrix element can be performed concurrently to increase efficiency. In Section [5], we show that the approach is efficient for computing the product of two small and medium size matrices.

3.2 Secure Inverse of Matrix Sum

Party A and B each hold a private $d \times d$ matrix \mathbf{A} and \mathbf{B} respectively. They want to securely compute the inverse of $\mathbf{A} + \mathbf{B}$. At the end of the secure computation, Party A and B each only holds a portion of the inverse matrix \mathbf{M}^a and \mathbf{M}^b respectively such that their sum $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{A} + \mathbf{B})^{-1}$; the inverse matrix is not known to both parties.

The steps to securely perform the inverse of matrix sum by two parties are shown in Protocol [2]. In Steps 1 to 3, Party B uses a random, non-singular matrix \mathbf{P} to hide its private matrix \mathbf{B} before sending it to Party A. In Steps 4 and 5, both both parties securely compute the inverse of $(\mathbf{A} + \mathbf{B})\mathbf{P}$ and then the product $\mathbf{P}(\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1})$, essentially eliminating the random matrix \mathbf{P} in the process. This yields the desired result $(\mathbf{A} + \mathbf{B})^{-1}$ in the form of two private portions \mathbf{M}^a and \mathbf{M}^b held by each party respectively.

In the case when the sum matrix $\mathbf{A} + \mathbf{B}$ is singular, a simple perturbation can be introduced to the sum matrix to make it non-singular. For instance, the

Protocol 2. Secure Inverse of Matrix Sum Protocol

Input: Party A has private $d \times d$ matrix \mathbf{A} and Party B has private $d \times d$ matrix \mathbf{B} .

Output: Party A obtains private matrix \mathbf{M}^a and Party B obtains private matrix \mathbf{M}^b such that their sum $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{A} + \mathbf{B})^{-1}$ yields the inverse of the sum of their private matrices.

- 1: Party B randomly generates a non-singular $d \times d$ matrix \mathbf{P} .
 - 2: Party A and Party B jointly perform secure matrix multiplication (using Protocol 1) to compute \mathbf{AP} , at the end of which, Party A and Party B each obtain \mathbf{S}^a and \mathbf{S}^b respectively such that $\mathbf{S}^a + \mathbf{S}^b = \mathbf{AP}$.
 - 3: Party B computes $\mathbf{S}^b + \mathbf{BP}$ and sends it to Party A.
 - 4: Party A computes $\mathbf{S}^a + \mathbf{S}^b + \mathbf{BP}$; i.e., $(\mathbf{A} + \mathbf{B})\mathbf{P}$, and then its inverse $\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1}$.
 - 5: Party B and Party A jointly perform secure matrix multiplication (using Protocol 1) on \mathbf{P} and $\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1}$, at the end of which, Party A and Party B each hold private portions \mathbf{M}^b and \mathbf{M}^a respectively such that $\mathbf{M}^a + \mathbf{M}^b = \mathbf{P}(\mathbf{P}^{-1}(\mathbf{A} + \mathbf{B})^{-1}) = (\mathbf{A} + \mathbf{B})^{-1}$.
-

perturbation method proposed by Hong and Yang [9] can be used to stabilize $\mathbf{A} + \mathbf{B}$ by adding a small perturbation matrix to \mathbf{A} or \mathbf{B} .

In contrast to the secure inverse of matrix sum protocol by Du et al. [3], Protocol 2 is more efficient and accurate as it uses only one random matrix \mathbf{P} instead of two matrices in Du’s protocol. Clearly, less (one random matrix less) algebraic operations yields more accurate computations results as less errors are introduced due to roundoff errors.

4 Privacy-Preserving FDA

4.1 PPFDA over Horizontally Partitioned Data

In this scenario, we have n data samples of d dimensions held by two parties. Let Party A hold the first n_1 data samples and Party B hold the remaining n_2 data samples; $n = n_1 + n_2$.

In Protocol 3, we show how $\mathbf{m}_1 - \mathbf{m}_2$ and \mathbf{S}_W^{-1} can be securely computed so as to yield \mathbf{w} in a secure manner. In addition to using Protocols 1 and 2, we also make use of the secure dot product technique by Jagannathan and Wright [10]. In this technique, all intermediate results are splitted into two random portions where each party holds one portion so that neither party is able to speculate anything about the intermediate results using only its private portion.

In Step 1, we show how $\mathbf{m}_1 - \mathbf{m}_2$ can be splitted into two random portions. As Party A holds n^a data samples (with n_i^a data samples of class i) and Party B holds n^b data samples (with n_i^b data samples of class i), $n_i = n_i^a + n_i^b$, the mean vector of class i as computed by Party A using only its private data samples is \mathbf{m}_i^a . Likewise, the mean vector of class i computed by Party B using its private data samples is \mathbf{m}_i^b . Hence, we have

Protocol 3. PPFDA over Horizontally Partitioned Data

Input: Party A has n^a private d -dimensional data samples. Party B has n^b private d -dimensional data samples.

Output: Party A and Party B securely compute a projection vector \mathbf{w} for the data samples held by them.

- 1: Party A computes $\mathbf{t}^a = (n_1^a/n_1)\mathbf{m}_1^a - (n_2^a/n_2)\mathbf{m}_2^a$. Party B computes $\mathbf{t}^b = (n_1^b/n_1)\mathbf{m}_1^b - (n_2^b/n_2)\mathbf{m}_2^b$.
- 2: Party A sets $\mathbf{S}_W^a = \mathbf{0}$ and Party B sets $\mathbf{S}_W^b = \mathbf{0}$.
- 3: **for** $i = 1$ to 2 **do**
- 4: **for** $j = 1$ to n_i **do**
- 5: **if** (\mathbf{x}_j^i is held by Party A) **then**
- 6: $\mathbf{u}^a = \mathbf{x}_j^i - (n_i^a/n_i)\mathbf{m}_i^a$ and $\mathbf{u}^b = -(n_i^b/n_i)\mathbf{m}_i^b$
- 7: **else**
- 8: $\mathbf{u}^a = -(n_i^a/n_i)\mathbf{m}_i^a$ and $\mathbf{u}^b = \mathbf{x}_j^i - (n_i^b/n_i)\mathbf{m}_i^b$
- 9: **end if**
- 10: Using Eq. [4](#), $\mathbf{M}^a + \mathbf{M}^b = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$
- 11: Update $\mathbf{S}_W^a = \mathbf{S}_W^a + \mathbf{M}^a$ and $\mathbf{S}_W^b = \mathbf{S}_W^b + \mathbf{M}^b$
- 12: **end for**
- 13: **end for**
- 14: Both parties jointly perform secure inverse of matrix sum (Protocol [2](#)) to obtain $\mathbf{S}^a + \mathbf{S}^b = (\mathbf{S}_W^a + \mathbf{S}_W^b)^{-1}$.
- 15: Both parties jointly perform secure matrix multiplication (Protocol [1](#)) to obtain $\mathbf{S}^a \mathbf{t}^b$ and $\mathbf{S}^b \mathbf{t}^a$; projection vector $\mathbf{w} = \mathbf{S}^a \mathbf{t}^a + \mathbf{S}^a \mathbf{t}^b + \mathbf{S}^b \mathbf{t}^a + \mathbf{S}^b \mathbf{t}^b$ (Eq. [6](#)) may now be computed.

Notations: n_1^a and n_2^a refer to the number of data samples of classes 1 and 2 respectively held by Party A; n_1^b and n_2^b refer to the number of data samples of classes 1 and 2 respectively held by Party B.

$$\begin{aligned}
 \mathbf{m}_1 - \mathbf{m}_2 &= \frac{n_1^a \mathbf{m}_1^a + n_1^b \mathbf{m}_1^b}{n_1} - \frac{n_2^a \mathbf{m}_2^a + n_2^b \mathbf{m}_2^b}{n_2} \\
 &= \left(\frac{n_1^a}{n_1} \mathbf{m}_1^a - \frac{n_2^a}{n_2} \mathbf{m}_2^a \right) + \left(\frac{n_1^b}{n_1} \mathbf{m}_1^b - \frac{n_2^b}{n_2} \mathbf{m}_2^b \right) \\
 &= \mathbf{t}^a + \mathbf{t}^b
 \end{aligned}$$

Next, Steps 2 to 13 securely compute $\mathbf{S}_W = \sum_{i=1,2} \sum_{j=1}^{n_i} (\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T$ (Eq. [2](#)). The secure manner to compute \mathbf{S}_W is to obtain two portion matrices \mathbf{S}_W^a and \mathbf{S}_W^b each held by Party A and Party B respectively. This is performed using the two **for** loops as shown in the protocol.

In Step 5, if \mathbf{x}_j^i belongs to Party A, then we have

$$\begin{aligned}
 \mathbf{x}_j^i - \mathbf{m}_i &= \left(\mathbf{x}_j^i - \frac{n_i^a}{n_i} \mathbf{m}_i^a \right) + \left(-\frac{n_i^a}{n_i} \mathbf{m}_i^a \right) \\
 &= \mathbf{u}^a + \mathbf{u}^b
 \end{aligned}$$

which yields $(\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$. The same process is performed if \mathbf{x}_j^i belongs to Party B.

Step 10 shows the secure manner to split the resultant product matrix of two vectors $(\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T$ into two portions such that

$$\mathbf{M} = (\mathbf{u}^a + \mathbf{u}^b)(\mathbf{u}^a + \mathbf{u}^b)^T = \mathbf{M}^a + \mathbf{M}^b \quad (4)$$

The element $m_{i,j}$ of matrix \mathbf{M} is computed as follows:

$$\begin{aligned} m_{i,j} &= (u_i^a + u_i^b)(u_j^a + u_j^b) \\ &= u_i^a \times u_j^a + \begin{bmatrix} u_i^a \\ u_j^a \end{bmatrix} \cdot \begin{bmatrix} u_j^b \\ u_i^b \end{bmatrix} + u_i^b \times u_j^b \\ &= m_{i,j}^a + m_{i,j}^b \end{aligned} \quad (5)$$

After securely computing the scalar product of vectors in Eq. 5, each element of matrix \mathbf{M} is splitted into two portions. Hence, the matrix \mathbf{M} is splitted into two private matrices. Overall, \mathbf{S}_W is securely splitted into two private portions \mathbf{S}_W^a and \mathbf{S}_W^b .

Using Protocol 2, $(\mathbf{S}_W)^{-1} = (\mathbf{S}_W^a + \mathbf{S}_W^b)^{-1}$ can be securely splitted into \mathbf{S}^a and \mathbf{S}^b such that $(\mathbf{S}_W)^{-1} = \mathbf{S}^a + \mathbf{S}^b$. Therefore

$$\begin{aligned} \mathbf{w} &= (\mathbf{S}_W)^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \\ &= (\mathbf{S}^a + \mathbf{S}^b)(\mathbf{t}^a + \mathbf{t}^b) \\ &= \mathbf{S}^a \mathbf{t}^a + \mathbf{S}^a \mathbf{t}^b + \mathbf{S}^b \mathbf{t}^a + \mathbf{S}^b \mathbf{t}^b \end{aligned} \quad (6)$$

Using Protocol 1, we securely compute $\mathbf{S}^a \mathbf{t}^b$ and $\mathbf{S}^b \mathbf{t}^a$. Thus, we are able to securely compute \mathbf{w} .

Analysis: Two parities are assumed to be semi-honest who strictly follow the protocol but collect all intermediate results during the execution of protocols to attack the private data of honest parties. As we observe, Protocol 3 applies two main secure building blocks: Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol. Both protocols depend on the Secure Scalar Product protocol that is provably secure [6]. Based on random share technique, we actually split all the intermediate results into two random shares (portions) except the final \mathbf{w} in Protocol 3. The private variables of one party are protected by the equivalent numbers of random portions known by itself only. Therefore we claim data privacy of honest parties are preserved.

We derive computational complexity of Protocol 3 here. As in Steps 3 to 13, the Secure Scalar Product protocol is invoked once to compute the scalar product of two vectors (2×1) (in Eq. 5), then one element of matrix \mathbf{M} $(d \times d)$ is securely split. As we know, there are $n_1 + n_2 = n$ data. Therefore, the Secure Scalar Product protocol is invoked $n \times d^2$ times in Steps 3 to 13 for computing the scalar product of two vectors (2×1) .

In Step 14, the Secure Inverse of Matrix Sum protocol is invoked once for splitting $(\mathbf{S}_W^a + \mathbf{S}_W^b)^{-1}$ $(d \times d)$. It requires to run the Secure Matrix Multiplication protocol twice (Step 2 and 5 in Protocol 2). In Step 14, the Secure Matrix

Multiplication protocol is invoked twice for splitting items $\mathbf{S}^a \mathbf{t}^b$ ($d \times 1$) and $\mathbf{S}^b \mathbf{t}^a$ ($d \times 1$) securely. In the Secure Matrix Multiplication protocol, it requires to perform the Secure Scalar Product protocol once to split one element of the desired matrix. The overall number of invoking Secure Scalar Product protocol in Step 14 and 15 is $(2d^2 + 2d)$ for computing the scalar product of two vectors ($d \times 1$).

Therefore, the overall computational complexity is $O(nd^2 + d^3)$ as the computational complexity of the Secure Scalar Product protocol is $O(\tau)$ for two vectors of length τ [6].

The communication of Protocol 3 between two parties mainly comes from depends on Secure Scalar Product protocol invoked in the protocol. Based on the analysis above, the the communication complexity of Protocol 3 depends on the overall number of the Secure Scalar Product protocol invoked, which is $O(nd^2 + d^3)$ as the communication complexity of the Secure Scalar Product protocol is $O(\tau)$ for two vectors of length τ [6].

In Section 5 we experimentally evaluate the efficiency and scalability of the secure building blocks.

4.2 PPFDA over Vertically Partitioned Data

In this scenario, d dimensions of data are distributed between two parties. Party A holds d_1 dimensions and Party B holds d_2 dimensions; $d = d_1 + d_2$. We show how \mathbf{w} can be securely computed in such a scenario.

In vertically partitioned data, we assume the first d_1 dimensions of data sample $\mathbf{x} = [x_1, x_2, \dots, x_d]$ are held by Party A: $\mathbf{x}^a = [x_1, x_2, \dots, x_{d_1}]^T$ and the remaining d_2 dimensions of \mathbf{x} are held by Party B: $\mathbf{x}^b = [x_{d_1+1}, x_{d_1+2}, \dots, x_{d_1+d_2}]^T$. We show that Party A and Party B may exchange their vertical data partitions with empty dimensions so that both parties have d dimensional partitions. In this way, the problem of vertically partitioned data is transformed to a horizontally partitioned problem so that the method in Section 4.1 can be applied to securely compute \mathbf{w} .

The transformation is as follows: For each d_1 dimension data sample \mathbf{x}^a of Party A, additional d_2 zeroes can be appended so that the data sample has d dimension:

$$(\mathbf{x}^a)' = [x_1, x_2, \dots, x_{d_1}, \overbrace{0, 0, \dots, 0}^{d_2}]^T$$

Likewise, data samples of Party B can be prepended with d_1 zeroes to become d dimensional:

$$(\mathbf{x}^b)' = [\overbrace{0, 0, \dots, 0}^{d_1}, x_{d_1+1}, x_{d_1+2}, \dots, x_{d_1+d_2}]^T$$

After the transformation, we have a total of $2n$ data samples of d dimensions rather than n data samples of d_1 held by Party A and n data samples of d_2 held by Party B.

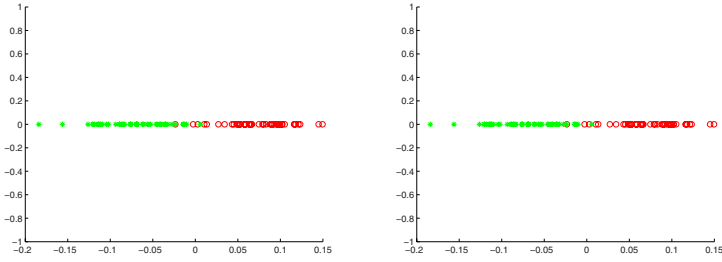


Fig. 1. Accuracy comparison of FDA without and with privacy

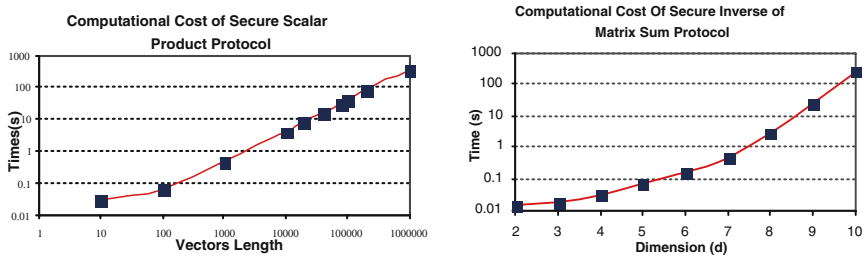


Fig. 2. Scalability of the Secure Scalar Product Protocol and Secure Inverse of Matrix Sum

5 Experiments

In this section, we discuss the implementation issues and evaluate the performance of the proposed protocols. All protocols were implemented in the C# language running under Microsoft Visual Studio 2005 environment. All experiments are performed on the Window XP operating system with 3.40GHz CPU and 1GB memory. As network performance mainly depends on the network speed and physical distance of two parties, we simply implemented parties as threads that exchange data directly by shared memory.

The dataset used is the Iris Plants Database from the UCI Machine Learning Depository. There are 150 data samples in three classes: “Iris Setosa”, “Iris Versicolour”, and “Iris Virginica”. As the latter two classes are not linearly separable, we select them as our analysis data. There are 4 numeric predictive attributes: “sepal length”, “sepal width”, “petal length”, and “petal width”.

The Paillier cryptosystem [14] was selected as our choice in the implementation. As the Paillier cryptosystem only encrypts non-negative integers, we have to deal with issues when real numbers and negative numbers occur. For real numbers, two parties multiply some large constants (e.g., 1000) to transform the real numbers to integers. We remove the effects of the constants by dividing the (intermediate) results by the constants. For negative numbers, the basic property of congruence $a + kn = a \pmod n$ is applied to transform negative integer a to positive integers by adding multiples of n .

Table 1. Efficiency analysis of Secure Inverse of Matrix Sum protocol

Dimension ($d \times d$)	Secure Inverse of Matrix Sum	Matrix Inverse	Overhead from Secure Matrix Multiplication
5	0.069516s	0.015586s	0.05393s
6	0.15586s	0.046758s	0.109102s
7	0.436408s	0.249376s	0.187032s
8	2.72755s	2.228798s	0.498752s
9	24.095956s	21.867158	2.228798s
10	244.8105s	240.055572s	4.754928s

Accuracy: To show the accuracy of performing FDA with privacy preservation using Protocol [3](#), we evaluated horizontally partitioned data where data instances of data set are uniformly distributed between two parties. The first figure in Fig. [1](#) was obtained by performing FDA using MATLAB. The second figure was obtained by Protocol [3](#). We clearly observe that accuracy is not reduced when we preserve the data privacy of the participant parties.

Scalability: We investigate the scalability of the two protocols proposed in this paper. For the Secure Matrix Multiplication protocol, we observe that the bulk of its operations are secure scalar products. Hence, we evaluated the scalability of the Secure Scalar Product protocol as shown in the first figure in Fig. [2](#). The running time is linear to the length of vectors as expected. Some random numbers in our implementation were generated offline. The time for two vectors of length 100,000 was estimated at 41 seconds, which is sufficiently low for small and medium data sets. The second figure in Fig. [2](#) shows the efficiency of the Secure Inverse of Matrix Sum protocol. We observe that the time to execute the protocol for more than 10×10 dimensions matrices becomes impractical. From Table [1](#), it is shown that the matrix inverse algorithm we used is time consuming due to the computation of matrix inverse and not due to overhead of the Secure Matrix Multiplication protocol. In our experiment, we use adjoint method [15](#) to perform matrix inverse as follows: $\mathbf{A}^{-1} = (1/\det)\mathbf{A}(\text{adjoint of } \mathbf{A})$ which is very computationally slow, comparing with other methods, such as Gauss-Jordan elimination and LU decomposition.

In these experiments, we only evaluated privacy-preserving FDA over horizontally partitioned data for low dimension (4×4). To apply the proposed protocol to higher dimension data would be part of our future work.

6 Conclusions

In this paper, we have proposed the privacy-preserving version of Fisher Discriminant Analysis over horizontally and vertically partitioned data. We have also proposed two basic secure building blocks for matrix computation: the Secure Matrix Multiplication protocol and Secure Inverse of Matrix Sum protocol. Finally, we have conducted experiments to demonstrate the scalability of the proposed secure building blocks and overheads to achieve the privacy when

performing FDA. Our future work includes applying the proposed protocol to high-dimensional data and extending the proposed protocols to multiple parties.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: Proceedings of the ACM SIGMOD, Dallas, Texas, United States, pp. 439–450 (2000)
2. Berkes, P.: Handwritten digit recognition with nonlinear fisher discriminant analysis. In: Proceedings of the International Conference on Artificial Neural Networks, pp. 285–287 (2005)
3. Du, W., Han, Y., Chen, S.: Privacy-preserving multivariate statistical analysis: Linear regression and classification. In: Proceedings of the 4th SIAM International Conference on Data Mining, Florida, April 22–24, 2004, pp. 222–233 (2004)
4. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern classification, 2nd edn. Wiley-Interscience, Chichester (2000)
5. Fisher, R.A.: The use of multiple measurements in taxonomic problems. *Annals of Eugenics* 7, 179–188 (1936)
6. Goethals, B., Laur, S., Lipmaa, H., Mielikainen, T.: On private scalar product computation for privacy-preserving data mining. In: Proceedings of the 7th Annual International Conference in Information Security and Cryptology, pp. 104–120
7. Han, S., Ng, W.K.: Privacy-Preserving Genetic Algorithms for Rule Discovery. In: Song, I.-Y., Eder, J., Nguyen, T.M. (eds.) DaWaK 2007. LNCS, vol. 4654, pp. 407–417. Springer, Heidelberg (2007)
8. 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)
9. Hong, Z.-Q., Yang, J.-Y.: Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition* 24(4), 317–324 (1991)
10. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: Proceedings of the 8th ACM SIGKDD, Chicago, Illinois, USA, pp. 593–599 (2005)
11. Katz, M., Meier, H.G., Dolfing, H., Klakow, D.: Robustness of linear discriminant analysis in automatic speech recognition. In: Proceedings of the 16th International Conference on Pattern Recognition, pp. 371–374 (2002)
12. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–53. Springer, Heidelberg (2000)
13. Lu, J., Plataniotis, K.N., Venetsanopoulos, A.N.: Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks* 14(1), 117–126 (2003)
14. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
15. Strang, G.: Linear algebra and its applications. Thomson, Brooks/Cole (2006)
16. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: Proceedings of the 8th ACM SIGKDD, Edmonton, Alberta, Canada, July 23–26, 2002, pp. 639–644 (2002)
17. Wan, L., Ng, W.K., Han, S., Lee, V.C.S.: Privacy-preservation for gradient descent methods. In: Proceedings of the 13th ACM SIGKDD, San Jose, California, USA, August 2007, pp. 775–783 (2007)

Unsupervised Change Analysis Using Supervised Learning

Shohei Hido, Tsuyoshi Idé, Hisashi Kashima,
Harunobu Kubo, and Hirofumi Matsuzawa

IBM Research, Tokyo Research Laboratory,
1623-14 Shimo-tsuruma, Yamato-shi, Kanagawa, 242-8502 Japan
{hido, goodidea, hkashima, kuboh, matuzawa}@jp.ibm.com

Abstract. We propose a formulation of a new problem, which we call *change analysis*, and a novel method for solving the problem. In contrast to the existing methods of change (or outlier) detection, the goal of change analysis goes beyond detecting whether or not any changes exist. Its ultimate goal is to find the explanation of the changes. While change analysis falls in the category of unsupervised learning in nature, we propose a novel approach based on *supervised* learning to achieve the goal. The key idea is to use a supervised classifier for interpreting the changes. A classifier should be able to discriminate between the two data sets if they actually come from two different data sources. In other words, we use a hypothetical label to train the supervised learner, and exploit the learner for interpreting the change. Experimental results using real data show the proposed approach is promising in change analysis as well as concept drift analysis.

Keywords: change analysis, two-sample test, concept drift.

1 Introduction

Outlier (or novelty) detection is one of the typical unsupervised learning tasks. It aims at deciding on whether or not an observed sample is “strange” based on some distance metric to the rest of the data. Change detection is similar to outlier detection, which is typically formulated as a statistical test for the probability distribution of a data set under some online settings.

In many practical data analysis problems, however, the problem of change detection appears with a slightly different motivation. For example, a marketing researcher may be interested in comparing the current list of customers’ profile with a past list to get information about changes. Here, detecting the changes itself is not of particular interest. What the researcher really wants is the detailed information about *how* they changed.

In this paper, we formulate this practically important problem, which we call *change analysis*. In contrast to change detection, we focus on developing a general framework of how to describe a change between two data sets. Clearly, the change analysis problem is an unsupervised learning task in nature. We assume that we are given two data sets, each of which contains a set of unlabeled vectors. Our goal is to find some diagnosis information based on the comparisons between the two data sets, without using side information about the internal structure of the system. The main contribution of this

paper is to show that this essentially unsupervised problem can be solved with *supervised learners*.

To date, the problem of comparing two data sets has been addressed in various areas. For example, the two-sample test [1][2][3], which is essentially to tell whether or not two (unlabeled) data sets are distinct, has a long history in statistics [4]. Another example is concept drift analysis [5][6][7], which basically addresses changes in supervised learners when the (labeled) training data set changes over time. However, most of the existing approaches have a serious drawback in practice in that they focus almost only on whether or not any change exists. As mentioned before, in most of the practical problems, what we really want to know is which variables could explain the change and how.

The layout of this paper is as follows. In Section 2, we describe the definition of the change analysis problem, and give an overview of our approach. Unexpectedly, this unsupervised problem can be solved using supervised learners, as explained in Section 3 in detail. Based on these sections, in Section 4, we present experimental results using real data to show the proposed approach is promising. Finally, we give a brief review of related work in Section 5 and conclude the paper in Section 6.

2 Problem Setting and Overview

In this section, we define a task of change analysis somewhat formally, and give an overview of our approach.

2.1 The Change Analysis Problem

Suppose that we are given two sets of unlabeled data, $X_A \equiv \{\mathbf{x}_A^{(1)}, \mathbf{x}_A^{(2)}, \dots, \mathbf{x}_A^{(N_A)}\}$ and $X_B \equiv \{\mathbf{x}_B^{(1)}, \mathbf{x}_B^{(2)}, \dots, \mathbf{x}_B^{(N_B)}\}$, where N_A and N_B are the numbers of data items in X_A and X_B , respectively. Each of $\mathbf{x}_A^{(i)}$ and $\mathbf{x}_B^{(i)}$ is an i.i.d. sample in a d -dimensional feature space.

This paper addresses two problems about these data sets. The first one is the change detection problem, which is basically the same as the two-sample problem:

Definition 1 (change detection problem). *Given nonidentical data sets X_A and X_B , tell whether or not the difference is significant, and compute the degree of discrepancy between X_A and X_B .*

Note that, unlike concept drift studies, we focus on unlabeled data in this problem. The second problem we address is the *change analysis problem*, which is stated as follows:

Definition 2 (change analysis problem). *Given nonidentical data sets X_A and X_B , output a set of decision rules that explain the difference in terms of individual features.*

Since no supervised information is given in getting the decision rules, this is an unsupervised learning task.

¹ The term of “difference analysis” could be more appropriate here, since we do not necessarily confine ourselves within online settings. However, to highlight the contrast to change detection, which is a well-known technical term, we will call the concept change analysis.

To understand the difference between these two problems, let us think about limitations of the two-sample test, which has been thought of as a standard approach to the change detection problem. The two-sample test is a statistical test which attempts to detect the difference between two data sets. Formally, it attempts to decide whether $P_A = P_B$ or $P_A \neq P_B$, where P_A and P_B are probability distributions for X_A and X_B , respectively. In statistics, two-sample tests are classified into two categories [4]. The first category is the parametric method, where a parametric functional form is explicitly assumed to model the distribution. In practice, however, such density modeling is generally difficult, since the distribution of real-world data does not have a simple functional form. In addition, even if a good parametric model such as Gaussian had been obtained, explaining the origin of the difference in terms of individual variables is generally a tough task, unless the variables are independent.

The second category is the nonparametric method, which allows us to conduct a statistical test without density modeling. If our interest were to detect only the discrepancy between two data sets, distance-like metrics such as the maximum mean discrepancy [3], the Kolmogorov-Smirnov statistic [1], energy-based metrics [8], and nearest neighbor statistics [2] are available for solving the change detection problem. However, these methods are not capable of handling the change analysis problem. While some of the two-sample tests offer asymptotic distributions for the data in such limit as large number of samples, it is generally very hard to answer the change analysis problem in practice. This is because, first, such a distribution is an asymptotic distribution, so it generally cannot be a good model for real-world data, where, e.g., the number of samples is finite. Second, since the nonparametric approach avoids density modeling, little information is obtained about the internal structure of the data.

2.2 Overview of Our Approach

Considering the limitations of the two-sample test, we propose a simple approach to these two problems. Our key idea is just as follows: Attach a hypothetical label $+1$ to each sample of X_A , and -1 to each sample of X_B . Then train a classifier in a supervised fashion. We call this classifier the *virtual classifier* (VC) hereafter.

Figure 1 shows a high-level overview of our approach, where \circ and \square indicate labels of $+1$ and -1 , respectively. In our approach, if two data sets actually have the differences, they should be correctly classified by the classifier. Thus a high classification accuracy p indicates a difference between X_A and X_B . For example, if $P_A = P_B$, the classification accuracy will be about 0.5 when $N_A = N_B$. However, if p is significantly larger than 0.5, we infer that the labels make difference, so $P_A \neq P_B$.

In addition, to solve the change analysis problem, we take advantage of the interpretability of classification algorithms. For example, the logistic regression algorithm gives the weight of each feature representing the importance. For another example, if the decision tree is employed, variables appearing in such nodes that are close to the root should have a major impact. In this way, we can get decision rules about the changes from the VC.

The advantages of this VC approach are as follows: First, it can solve the change detection and analysis problem at the same time. The classifier readily gives the degree of change as the classification accuracy, and also provides diagnosis information about

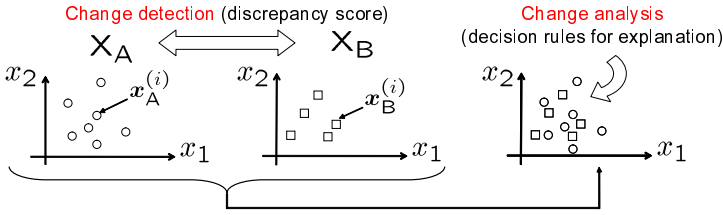


Fig. 1. High-level overview of the virtual classifier approach

changes through its feature selection functions. Second, the VC approach does not need density estimation, which can be hard especially for high dimensional data. Finally, the VC approach allows us to evaluate the significance of changes simply by a binomial test. This is an advantage over traditional nonparametric two-sample tests, which have focused on asymptotic distributions that hold only in some limit.

3 Virtual Classifier Approach to Change Analysis

This section presents details of our supervised learning approach to change analysis. For notations, we use bars to denote data sets including the hypothetical labels, such as $\bar{X}_A \equiv \{(\mathbf{x}_A^{(i)}, +1) \mid i = 1, \dots, N_A\}$ and $\bar{X}_B \equiv \{(\mathbf{x}_B^{(i)}, -1) \mid i = 1, \dots, N_B\}$. The prediction accuracy of VCs is represented by p .

3.1 Condition of No Change

Suppose that we are given the combined data set $\bar{X} \equiv \bar{X}_A \cup \bar{X}_B$, and a binary classification algorithm L . We train L using \bar{X} , and evaluate the classification accuracy p , making use of k -fold cross validation (CV). In particular, randomly divide \bar{X} into k equi-sized portions, leave out one portion for test, and use the remaining $(k - 1)$ portions for training. The overall prediction accuracy p is computed as the average of those of the k classifiers.

If $P_A = P_B$, classification of each of the samples in \bar{X} by L can be viewed as a Bernoulli trial. Thus the log-likelihood of $N_A + N_B$ trials over all the members of \bar{X} will be

$$\ln [q^{N_A}(1 - q)^{N_B}]$$

under the assumption of i.i.d. samples, where q is the probability of the class A. By differentiating this w.r.t. q , and setting the derivative zero, we have the maximum likelihood solution of this binomial process as $q = N_A / (N_A + N_B)$. Since the classification accuracy p should be $\max\{q, 1 - q\}$, we see that p is given by

$$p_{\text{bin}} \equiv \frac{\max\{N_A, N_B\}}{N_A + N_B}, \tag{1}$$

where the subscript represents binomial.

If $P_A \neq P_B$, so the information of the class labels is important, the classification accuracy will be considerably higher than p_{bin} . Specifically, the larger the differences

they have, the higher the prediction accuracy will become. One of the major features of our VC approach is that it enables us to evaluate the significance of p via a binomial test. Consider a null hypothesis that the prediction accuracy is given by p_{bin} , and assume $N_A > N_B$ for simplicity. For a value of the confidence level $\alpha > 0$, we reject the null hypothesis if

$$\sum_{n_A=Np}^N \frac{N!}{n_A!(N-n_A)!} p_{\text{bin}}^{n_A} (1-p_{\text{bin}})^{N-n_A} \leq \alpha, \quad (2)$$

where $N = N_A + N_B$. This means that the class labels are so informative that p is sufficiently higher than p_{bin} . If we parameterize the critical probability as $(1 + \gamma_\alpha)p_{\text{bin}}$, the condition of no change is represented as

$$p < (1 + \gamma_\alpha)p_{\text{bin}}. \quad (3)$$

For a numerical example, if $N = 1000$ and $p_{\text{bin}} = 0.5$, the 5% and 1% confidence levels correspond to $\gamma_{0.05} = 0.054$ and $\gamma_{0.01} = 0.076$, respectively. For relatively large N , Gaussian approximation can be used for computing γ_α [4].

3.2 Change Analysis Algorithm

Once the binomial test identifies that the difference between X_A and X_B is significant, we re-train L (or another type of classification algorithm) using all the samples in \bar{X} . If some features play a dominant role in the classifier, then they are the ones that characterize the difference. As an example, imagine that we have employed the C4.5 decision trees [9] as L . The algorithm iteratively identifies the most important feature in terms of information gain, so such features that appear closest to the root will be most important. Thus focusing on such nodes amounts to feature selection, and the selected features are the ones that explain the difference. In this way, feature selection and weighting functions of L are utilized in change analysis.

We summarize our change analysis algorithm in Fig. 2. The first half (1-3) essentially concerns change detection by evaluating the significance of the changes through the binomial test, while the second half (4-5) addresses change analysis. As shown, there are two input parameters, α and k .

3.3 Application to Labeled Data

While the algorithm in Fig. 2 is for unlabeled data, we can extend the algorithm for labeled data. This extension is practically important since it enables us to do change analysis between classifiers. Suppose that we are given a classification algorithm M , and two labeled data sets D_A and D_B , defined as $\{(\mathbf{x}_A^{(i)}, y_A^{(i)}) | i = 1, \dots, N_A\}$ and $\{(\mathbf{x}_B^{(i)}, y_B^{(i)}) | i = 1, \dots, N_B\}$, respectively, where $y_A^{(i)}$ and $y_B^{(i)}$ represent class labels. We train M based on D_A and D_B to obtain classifiers M_A and M_B , respectively. What we wish to solve is a change analysis problem between M_A and M_B : Output a set of decision rules that explain the difference between M_A and M_B in terms of individual features.

Algorithm: Change Analysis

INPUT:

- Two data sets X_A and X_B
 - Binary classification algorithm L
 - Number of folds k
 - Significance level $\alpha > 0$
1. Give the positive label to each sample of X_A , and the negative label to each sample of X_B .
 2. Train L based on k -fold cross-validation to obtain the estimated predictive accuracy p .
 3. If $p < p_{\text{bin}}(1 + \gamma_\alpha)$, then quit. Otherwise, report that X_A and X_B have different distributions.
 4. Re-train L on all of the data.
 5. Investigate the trained classifier to understand the differences between X_A and X_B .

Fig. 2. The virtual classifier algorithm for change analysis

To solve this, we create unlabeled data sets based on the following strategy. For each sample, $\mathbf{x}_A^{(i)}$ or $\mathbf{x}_B^{(i)}$, we make classification with both M_A and M_B . If the predictions are inconsistent, then we put the sample into a set X_A , otherwise into X_B . Scanning all the samples, we have two unlabeled data sets X_A and X_B . By construction, X_A characterizes the inconsistencies between M_A and M_B , while X_B characterizes their commonalities. Thus, by making use of the change analysis algorithm in Fig. 2 for these X_A and X_B , detailed information about the inconsistencies will be obtained. In our context, the quantity

$$\rho \equiv N_{\text{inc}} / (N_A + N_B) \quad (4)$$

works as the degree of the inconsistencies between M_A and M_B (or D_A and D_B), where N_{inc} represents the number of samples whose predictions are inconsistent.

When the number of possible values for the target variable is small, it is useful to extend the change analysis algorithm to include multi-class classification. As an example, suppose that the given label is binary, i.e. $y_A^{(i)}, y_B^{(i)} \in \{\pm 1\}$. We separate the inconsistent set X_A into two subsets X_{A1} and X_{A2} . Here, X_{A1} consists of the inconsistent samples whose original prediction is $+1$ but cross-classification gives -1 . Similarly, X_{A2} consists of the inconsistent samples that make a transition from -1 to $+1$. Then we apply a three-class classification algorithm L to classify X_{A1} , X_{A2} , and X_B . Finally, we examine the resulting classifier for each type of disagreement.

4 Experiment

We evaluated the utility of the VC approach for change analysis using synthetic as well as real-world data. In the following experiments, we used $\alpha = 0.05$ and $k = 10$ unless otherwise noted. For a classification algorithm L , we mainly used the C4.5 decision trees (DT) algorithm implemented as `J48` in Weka [9], which has a parameter named *minNumObj* meaning the minimum number of instances per leaf. To see the degree of linear separability between the two data sets, we additionally used logistic regression (LR) also implemented as `Logistic` in Weka. Two parameters in `Logistic` (a ridge parameter and the maximum iterations) were left unchanged to the default values (10^{-8} and infinity, respectively).

4.1 Synthetic Data

We conducted two experiments based on synthetic data with $N_A = N_B = 500$. For this number of samples, the critical accuracy is given by 0.527 ($\gamma_{0.05} = 0.054$). In both of the experiments, the ten features were independently generated based on zero-mean Gaussians.

For the *first* experiment, the data sets X_A and X_B were designed so that P_A and P_B had a significant difference. In X_A , the standard deviations (denoted by σ) were set to be 1.0 except for *Attr1* (the first feature), where σ was set to be 4.0. On the other hand, in X_B , all the σ s were 1.0 except for *Attr2* (the second feature), where σ was set to be 4.0. Figure 3(a) shows the marginal distribution of this data set in the *Attr1-Attr2* space. Our goal is to pick up *Attr1* and *Attr2* as features that are responsible for the difference.

We conducted change analysis for this data set with $\text{minNumObj} = 10$ for DT. The estimated prediction accuracy p computed by 10-fold CV was 0.797 (DT), which far exceeds the critical accuracy. This means that the two data sets were correctly judged as being different. Figure 3(b) represents the DT as the VC, where the labels of the ellipses and the edges show the split variables and the decision rules, respectively. The shaded boxes enclose the class labels as well as (1) the number of instances fallen into the node and (2) the number of misclassified instances in the form of (1)/(2). The latter is to be omitted when zero. The decision boundaries found by the DT are shown by the lines in Fig. 3(a). Clearly, the model learned the intended nonlinear change between *Attr1* and *Attr2*. Note that, when LR was used as L, 10-fold CV gave only $p = 0.505$, which is below the critical accuracy. This result clearly shows the crucial role of nonlinear decision boundaries.

For the *second* experiment, P_A and P_B were designed to be the same. In both X_A and X_B , all the σ s were set to be 1.0 except for *Attr2*, where $\sigma = 4.0$. Figure 4 shows the marginal distribution corresponding to Fig. 3(a). In contrast to the first experiment, the DT model naturally showed a low p of 0.500, indicating that the differences were not statistically significant. This result shows that our approach using DT generates a valid classifier with statistical significance only when the data set contains a difference between classes.

4.2 Spambase Data

Spambase is a public domain data set in UCI Machine Learning Repository [10]. While the original data contains spam and non-spam email collections, we used only the 1,813 instances belonging to the spam email set. The features consist of fifty-five continuous values of word and symbol statistics. We divided the spam set into halves, X_A and X_B , keeping the original order of the instances unchanged. In this setting, the critical accuracy is 0.520 ($\gamma_{0.05} = 0.039$). We performed change analysis for X_A and X_B to see if there was any hidden shift in the data. We used $\text{minNumObj} = 2$ for DT.

Interestingly, the 10-fold CV produced a rather high prediction accuracy of $p = 0.548$ (DT), which is higher than the critical accuracy. According to the VC, the major features were the frequencies of the words ‘edu’, ‘85’, and ‘hp’, although space limitation does not permit showing the output DT. Considering the additional fact that LR produced just $p = 0.515$, we conclude that the spam class in Spambase has some

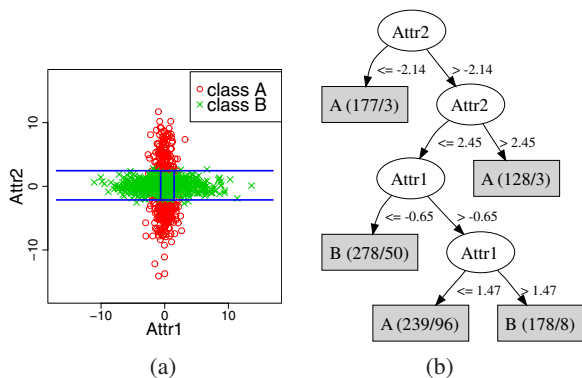


Fig. 3. (a) Distribution over *Attr1* and *Attr2* in the first synthetic data, and (b) the resulting virtual classifier

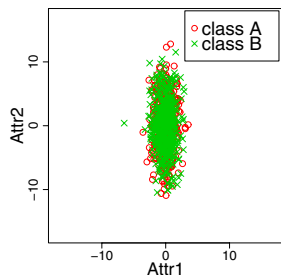


Fig. 4. Distribution over *Attr1* and *Attr2* in the second synthetic data

nonlinear changes on the word frequencies, which are difficult to find using a linear model like LR. This result is of particular practical importance, since it suggests that learning algorithms that depend on the order of the training samples might tend to have considerable biases.

4.3 Enron Email Data

The Enron email data set is an archive of real email at the now defunct Enron Corporation, and no class labels are available [11]. We used the year 2001 subset that contains 272,823 email messages in a bag-of-words representation [12]. We separated the data into the first (1H) and the second (2H) halves of this year, and generated feature vectors by including the 100 and 150 most frequent words in each period. Meaningless zero vectors including none of the selected feature words were omitted. Each half was further divided into halves to allow comparison on quarterly basis. We conducted change analysis within either 1H or 2H with $numMinObj = 1,000$. For example, in the analysis of 2H, X_A and X_B roughly correspond to the data in the third (3Q) and fourth (4Q) quarters, respectively.

Table 1 shows the estimated prediction accuracies. We see that both LR and DT mark accuracies much higher than the critical accuracies. To explore the details of the differences, we picked the 2H data, and did change analysis to obtain the DT in Fig. 5, where top 5 nodes from the root have been selected, comparing between 100- and 150-word models. The notation of the trees are the same as Section 4.1, although the rank of each feature has been added here (‘access’ is 44th frequent, etc.). The threshold values represent the occurrence numbers of feature words in each email. Since we followed the simple frequency-based feature generation strategy, the 150-word tree tends to include such words that bear particular meanings.

We see that ‘position’ is at the root node in the 150-word model in spite of its less frequency (144th rank). Enron went bankrupt at the end of 2001. If we imagine what had been talked about by the employees who were doomed to lose their job position,

Table 1. Prediction accuracies on Enron

Data set		Algorithm	
Period	Words	LR	DT
2001-1H	100	64.3%	67.4%
2001-1H	150	65.4%	68.4%
2001-2H	100	60.9%	62.8%
2001-2H	150	62.3%	64.1%

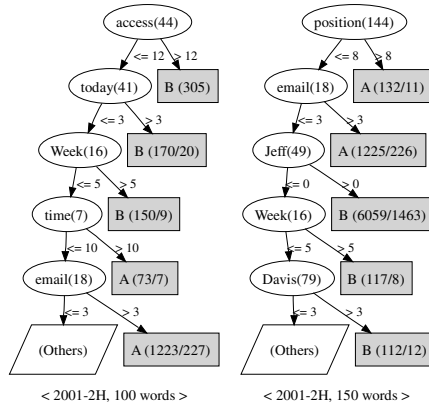


Fig. 5. VCs on the Enron 2001-2H data set

this result is quite suggestive. In addition, we see that ‘Jeff’ and ‘Davis’ are dominant features to characterize the 4Q data. Interestingly, the name of CEO of Enron in 2001 was Jeffrey Skilling, who unexpectedly resigned from this position on August 2001 after selling all his stock options. Many employees must have said something to him at the moment of the collapse. For Davis, there was a key person named Gray Davis, who was California’s Governor in the course of the California electricity crisis in the same year. It may result from his response to the investigation of Enron in 4Q. Note that the VC has discovered these key persons without any newspaper information, demonstrating the utility in studying the dynamics of complex systems such as Enron.

4.4 Academic Activities Data

As an example of application to labeled and categorical data, we performed change analysis for “academic activities” data collected in a research laboratory. This data set consists of 4,683 records over five years in the form of $(\mathbf{x}^{(s)}, y^{(s)})$, where s is the time index and $y^{(s)}$ represents a binary label of either ‘Y’ (meaning important) or ‘N’ (unimportant). Each of the vectors $\mathbf{x}^{(s)}$ includes three categorical features, *title*, *group*, and *category*, whose values are shown in Table 2.

Since the labels are manually attached to $\mathbf{x}^{(s)}$ s by evaluating each activity, it greatly depends on subjective decision-making of the database administrator. For example, some administrators might think of PAKDD papers as very important, while other might not. Triggered by such events as job rotations of the administrator and revisions of evaluation guidelines, the trend of decision-making is expected to change over time. The purpose on this analysis is to investigate when and what changes have occurred in the decision criteria to select importance labels.

We created 14 data subsets by dividing the data on quarterly basis, denoted by D_1, D_2, \dots, D_{14} . First, to see whether or not distinct concept drifts exist over time, we computed the inconsistency score ρ (see Eq. (4)) between neighboring quarters. Specifically, we think of D_A and D_B as D_t and D_{t+1} for $t = 1, 2, \dots, 13$. For M , we employed decision trees. Figure 6 shows the inconsistency score ρ for all the pairs. We see that

Table 2. Three features and their values in the academic activity data

<i>category</i>	<i>title</i>	<i>group</i>
GOVERNANCE, EDITOR, ORGANIZATION, PROFESSIONAL ACTIVITY	COMMITTEE, MEMBER, AWARD, OTHERS	UNIVERSITY, DOMESTIC, STANDARD, PUBLISHER, SOCIETY1, OTHERGROUPS

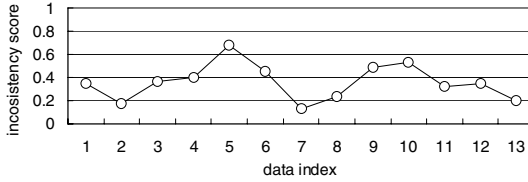


Fig. 6. Inconsistency scores ρ between D_t and D_{t+1} . The largest score can be seen where $t = 5$.

two peaks appear around $t = 5$ and $t = 10$, showing clear concept drifts at those periods. Interestingly, these peaks correspond to when the administrator changed off in reality, suggesting the fact that the handover process did not work well.

Next, to study what happened around $t = 5$, we picked D_5 and D_6 for change analysis. Following the procedure in Section 3.3, we obtained the VC as shown in Fig. 7. Here, we used a three-class DT based on three sets X_{A1} , X_{A2} and X_B , where X_{A1} includes samples whose predicted labels make a transition of $Y \rightarrow N$. The set X_{A2} includes samples of $N \rightarrow Y$, while X_B includes consistent samples of $Y \rightarrow Y$. If we focus on the leaves of ‘NY’ in Fig. 7 representing the transition from N to Y, we find interesting changes between D_5 and D_6 : The new administrator at $t = 6$ tended to put more importance on such academic activities as program and executive committees as well as journal editors.

One might think that there can be a simpler approach that two decision trees M_5 and M_6 are directly compared, where M_5 and M_6 are decision trees trained only within D_5

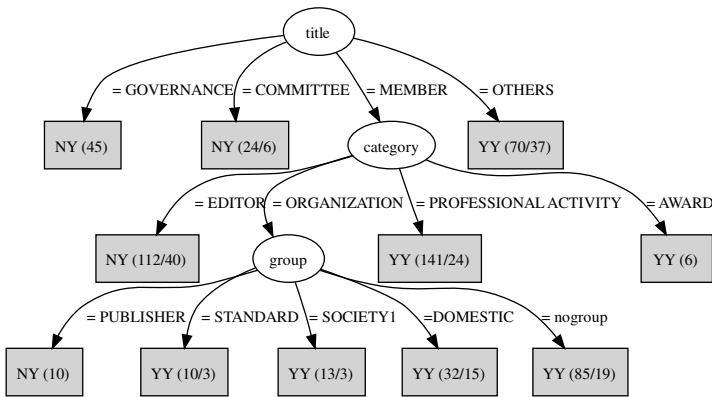


Fig. 7. Virtual classifier for D_5 and D_6

and D_6 , respectively. However, considering complex tree structures of decision trees, we see that direct comparison between different decision trees is generally difficult. Our VC approach provides us a direct means of viewing the difference between the classifiers, and is in contrast to such a naive approach.

5 Related Work

The relationship between supervised classifiers and the change detection problem had been implicitly suggested in the 80's [2], where a nearest-neighbor test was used to solve the two-sample problem. However, it did not address the problem of change analysis. In addition, the nearest-neighbor classifier was not capable of explaining changes, since it did not construct any explicit classification model. FOCUS is another framework for quantifying the deviation between the two data sets [13]. In the case of supervised learning, it constructs two decision trees (dt-models) on each data set, then expands them further until both trees converge to the same structure. The differences between the numbers of the instances which fall into the same region (leaf) indicate the deviation between the original data sets. In high-dimensional settings, however, the models should become ineffective since the size of the converged tree increases exponentially therefore the method requires substantial computational cost and massive instances.

Graphical models such as Bayesian networks [14] are often used in the context of root cause analysis. By adding a variable indicating one of the two data sets, in principle Bayesian networks allow us to handle change analysis. However, a graphical modeling approach inevitably requires a lot of training data and involves extensively time-consuming steps for graph structure learning. Our VC approach allows us to directly explain the data set labels. This is in contrast to graphical model approaches, which basically aim at modeling the joint distribution over all variables.

In stream mining settings, handling concept drift is one of the essential research issues. While much work has been done in this area [5,6,7], little of that addresses the problem of change analysis. One of the exceptions is KBS-stream [15] that quantifies the amount of concept drift, and also provides a difference model. The difference model of KBS-Stream tries to correctly discriminate the positive examples from the negative examples in the misclassified examples under the current hypothesis. On the other hand, our VC tries to correctly discriminate the misclassified examples by the current hypothesis against the correctly classified examples. Both models are of use to analyze concept drift, but the points of view are slightly different.

Other studies such as ensemble averaging [16] and fast decision trees [17] tackle problems which are seemingly similar to but essentially different from change analysis.

6 Conclusion

We have proposed a new framework for the change analysis problem. The key of our approach is to use a virtual classifier, based on the idea that it should be able to tell the two data apart if they came from two different data sources. The resulting classifier is a model explaining the differences between the two data sets, and analyzing this model allows us to obtain insights about the differences. In addition, we showed

that the significance of the changes can be statistically evaluated using the binomial test. The experimental results demonstrated that our approach is capable of discovering interesting knowledge about the difference.

For future work, although we have used only decision trees and logistic regression for the virtual classifier, other algorithms also should be examined. Extending our method to allow on-line change analysis and regression models would also be interesting research issues.

References

1. Friedman, J., Rafsky, L.: Multivariate generalizations of the Wolfowitz and Smirnov two-sample tests. *Annals of Statistics* 7, 697–717 (1979)
2. Henze, Z.: A multivariate two-sample test based on the number of nearest neighbor type coincidences. *Annals of Statistics* 16, 772–783 (1988)
3. Gretton, A., Borgwardt, K., Rasch, M., Schölkopf, B., Smola, A.J.: A kernel method for the two-sample-problem. In: *Advances in Neural Information Processing Systems* 19, pp. 513–520. MIT Press, Cambridge (2007)
4. Stuart, A., Ord, J.K.: *Kendall's Advanced Theory of Statistics*, 6th edn., vol. 1. Arnold Publishers Inc. (1998)
5. Fan, W.: Streamminer: A classifier ensemble-based engine to mine concept-drifting data streams. In: *Proc. the 30th Intl. Conf. Very Large Data Bases*, pp. 1257–1260 (2004)
6. Wang, H., Yin, J., Pei, J., Yu, P.S., Yu, J.X.: Suppressing model overfitting in mining concept-drifting data streams. In: *Proc. the 12th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pp. 20–23 (2006)
7. Yang, Y., Wu, X., Zhu, X.: Combining proactive and reactive predictions for data streams. In: *Proc. the 11th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pp. 710–715 (2005)
8. Zech, G., Aslan, B.: A multivariate two-sample test based on the concept of minimum energy. In: *Proceedings of Statistical Problems in Particle Physics, Astrophysics, and Cosmology*, pp. 8–11 (2003)
9. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools*. Morgan Kaufmann, San Francisco (2005)
10. Newman, D., Hettich, S., Blake, C., Merz, C.: *UCI repository of machine learning databases* (1998)
11. Klimt, B., Yang, Y.: The Enron Corpus: A New Dataset for Email Classification Research. In: Boulicaut, J.-F., Esposito, F., Giannotti, F., Pedreschi, D. (eds.) *ECML 2004. LNCS (LNAI)*, vol. 3201, pp. 217–226. Springer, Heidelberg (2004)
12. Other forms of the Enron data:
<http://www.cs.queensu.ca/~skill/otherforms.html>
13. Ganti, V., Gehrke, J.E., Ramakrishnan, R., Loh, W.: A framework for measuring changes in data characteristics. *Journal of Computer and System Sciences* 64(3), 542–578 (2002)
14. Pearl, J.: *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, San Francisco (1988)
15. Scholz, M., Klinkenberg, R.: Boosting classifiers for drifting concepts. *Intelligent Data Analysis Journal* 11(1), 3–28 (2007)
16. Street, W.N., Kim, Y.: A streaming ensemble algorithm (SEA) for large-scale classification. In: *Proc. the 7th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pp. 377–382 (2001)
17. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: *Proc. the 7th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pp. 97–106 (2001)

ANEMI: An Adaptive Neighborhood Expectation-Maximization Algorithm with Spatial Augmented Initialization

Tianming Hu^{1,2}, Hui Xiong³, Xueqing Gong¹, and Sam Yuan Sung⁴

¹ East China Normal University

tmhu@ieee.org, xqgong@sei.ecnu.edu.cn

² Dongguan University of Technology

³ Rutgers, the State University of New Jersey

hxiong@andromeda.rutgers.edu

⁴ South Texas College

sysung@southtexascollege.edu

Abstract. The Neighborhood Expectation-Maximization (NEM) algorithm is an iterative EM-style method for clustering spatial data. Unlike the traditional EM algorithm, NEM has the spatial penalty term incorporated in the objective function. The clustering performance of NEM depends mainly on two factors: the choice of the spatial coefficient, which is used to weigh the penalty term; and the initial state of cluster separation, to which the resultant clustering is sensitive. Existing NEM algorithms usually assign an equal spatial coefficient to every site, regardless of whether this site is in the class interior or on the class border. However, when estimating posterior probabilities, sites in the class interior should receive stronger influence from its neighbors than those on the border. In addition, initialization methods deployed for EM-based clustering algorithms generally do not account for the unique properties of spatial data, such as spatial autocorrelation. As a result, they often fail to provide a proper initialization for NEM to find a good solution in practice. To that end, this paper presents a variant of NEM, called ANEMI, which exploits an adaptive spatial coefficient determined by the correlation of explanatory attributes inside the neighborhood. Also, ANEMI runs from the initial state returned by the spatial augmented initialization method. Finally, the experimental results on both synthetic and real-world datasets validated the effectiveness of ANEMI.

1 Introduction

Spatial data have traditional numeric and categorical attributes as well as spatial attributes that describe the spatial information of the objects, such as location and shape. The assumption of independent and identical distribution (IID) is no longer valid for spatial data. In the spatial domain, everything is related to everything else but nearby objects are more related than distant objects [1]. For example, houses in nearby neighborhoods tend to have similar prices which are

affected by one another. In remote sensing images, close pixels usually belong to the same landcover type: soil, forest, etc.

Traditional model based clustering algorithms, such as the Expectation-Maximization (EM) algorithm [2], do not take spatial information into consideration. To this end, Ambroise et al. [3] proposed the Neighborhood Expectation-Maximization (NEM) algorithm, which extends EM by adding a spatial penalty term into the objective function. Such a spatial penalty favors those solutions where neighboring sites are assigned to the same class. The performance of NEM depends mainly on two factors. One is the choice of the spatial coefficient, which is used to weigh the penalty term in the objective function and specifies the degree of spatial smoothness in the clustering solution. Another one is the initial state of cluster separation, from which NEM starts iterative refinement.

For the choice of the spatial coefficient, NEM employs a fixed coefficient that has to be determined a priori and is often set empirically in practice. However, it may not be appropriate to assign a fixed coefficient to every site, regardless of whether it is in the class interior or on the class border. When estimating posterior probabilities, sites in the class interior should receive stronger influence from its neighbors than those on the border. In addition, for initialization, it is usually impossible for NEM to achieve the global optimization, which has been shown to be NP-hard [4]. The clustering performance of NEM is very sensitive to the initial state of cluster separation. As a result, a proper initialization is of great value for the success of finding a better sub-optimal solution in practice. Nevertheless, existing initialization methods for NEM and other EM-style clustering algorithms do not account for such spatial information.

To address the above mentioned challenges, in this paper, we propose a variant of NEM: Adaptive Neighborhood EM with spatial augmented Initialization (ANEMI) for spatial clustering. ANEMI exploits a site-sensitive coefficient, which is determined by the correlation of explanatory attributes inside the neighborhood. In addition, the refinement process of ANEMI starts from the initial state returned by the spatial augmented initialization method. Indeed, by pushing spatial information further into the whole clustering process, our experimental results on both synthetic and real datasets show that ANEMI generally leads to better clustering performance than traditional NEM.

Overview. The remainder of this paper is organized as follows. Section 2 introduces the problem background and related work. In Section 3, after reviewing the basic concepts of NEM, we present the ANEMI algorithm. Experimental results are reported in Section 4, where both the augmented initialization and the adaptive coefficient assignment of ANEMI are evaluated thoroughly. Finally, in Section 5, we draw conclusions and suggest future work.

2 Background

In this section, we first introduce the background by formulating the problem. Then, we briefly review related work.

2.1 Problem Formulation

The goal of spatial clustering is to partition data into groups so that pairwise dissimilarity, in both non-spatial space and spatial space, between those assigned to the same cluster tend to be smaller than those in different clusters. In detail, we are given a spatial framework of n sites $S = \{s_i\}_{i=1}^n$, which are described with an observable set $X = \{\mathbf{x}_i \equiv \mathbf{x}(s_i)\}_{i=1}^n$ of random variables. Note that we overload notation and use X to refer to both the given dataset and their corresponding random variables. Often it is enough to know the neighborhood information, which can be represented by a contiguity matrix W with $W_{ij} = 1$ if s_i and s_j are neighbors and $W_{ij} = 0$ otherwise. We need to infer the unobservable (hidden) set $Y = \{y_i \in \{1, 2, \dots, K\}\}_{i=1}^n$ of random variables, corresponding to the cluster label of \mathbf{x}_i . Due to the spatial constraint, the resulting random field defined over Y is a Markov Random Field (MRF), where $P(y_i | Y - \{y_i\}) = P(y_i | \{y_j : W_{ij} = 1\})$. Hence it is more appropriate to model the posterior distribution of y_i as $P(y_i | \mathbf{x}_i, \{y_j : W_{ij} = 1\})$ instead of $P(y_i | \mathbf{x}_i)$.

2.2 Related Work

There are roughly two categories of work that are related to the main theme of this paper: spatial clustering and the cluster initialization methods for iterative refinement clustering.

Most conventional clustering methods in the literature treat each object as a point in the high dimensional space and do not distinguish spatial attributes from non-spatial attributes. These clustering methods can be divided into the following groups: distance-based [5], density-based [6], hierarchy-based [7], etc.

In the field of spatial clustering, some methods only handle 2-dimensional spatial attributes [8] and deal with problems like obstacles which are unique in clustering geo-spatial data [9]. To incorporate spatial constraints, the simplest method is to directly add spatial information, e.g., spatial coordinates, into datasets [10]. Others achieve this goal by modifying existing algorithms, e.g., allowing an object assigned to a class if this class already contains its neighbor [11]. Another class, where our algorithm falls, selects a model that encompasses spatial information. This can be achieved by modifying a criterion function that includes spatial constraints [12], which mainly comes from image analysis where MRF and EM-style algorithms were intensively used [13, 14].

Clustering using mixture models with conventional EM does not account for spatial information. NEM extends EM by adding a weighted spatial penalty term in the objective function. The clustering performance of NEM depends largely on the global fixed coefficient, the weight of the penalty. If further information about structure is available, spatially varying coefficient models can be employed, which has been mainly investigated for regression problems [15].

In practice, the subsequent cluster refinement in NEM is only a sub-task of the whole clustering, which succeeds the execution of a certain initialization method. With the initialization methods returning a set of seed centers, the data are assigned to the closest center and thus an initial clustering is obtained for NEM to refine. Roughly speaking, the cluster initialization methods fall into three

major families: random sampling, distance optimization and density estimation. We will examine three representative methods in more details later.

3 The ANEMI Algorithm

In this section, we first introduce the basics of NEM from the MRF perspective. Then we present the cluster refinement part of the ANEMI algorithm, which exploits an adaptive scheme of coefficient assignment. Finally, we discuss the initialization methods for the ANEMI algorithm.

3.1 The MRF Framework

By the Hammersley-Clifford theorem [16], the prior probability of a cluster label configuration $Y = \{y_i\}_{i=1}^n$ (a realization of the MRF) can be expressed as a Gibbs distribution [13], $P(Y) = \frac{1}{Z_1} \exp(-V(Y))$, where Z_1 is a normalizing constant, and $V(Y)$ is the overall label configuration potential function. In the clustering framework, the conditional probability of X given Y has the form $P(X|Y) = f(X|Y, \Phi)$, a density function parameterized with Φ . The posterior probability becomes $P(Y|X) = \frac{1}{Z_2} P(Y)P(X|Y)$, where $Z_2 = Z_1 P(X)$ is the normalizing constant. Hence finding the maximum a-posteriori (MAP) configuration of the hidden MRF is equivalent to maximizing the logarithm of $P(Y|X)$ (scaled by a constant)

$$U = \ln(f(X|Y, \Phi)) - V(Y) \tag{1}$$

3.2 Neighborhood EM (NEM)

For the potential function $V(Y)$, NEM employs a soft version of the pairwise Potts model, $V(Y) = -\sum_{i,j} W_{ij} I(y_i = y_j)$, where $I(\cdot)$ is the indicator function with $I(\text{true}) = 1$ and $I(\text{false}) = 0$. In detail, let \bar{P} denote a set of distributions $\{\bar{P}_{ik} \equiv \bar{P}(y_i = k)\}$ governing $\{y_i\}$. Termed ‘‘spatial penalty’’, the potential function used in NEM is $G(\bar{P}) = -\frac{1}{2} \sum_{i,j} W_{ij} \sum_{k=1}^K \bar{P}_{ik} \bar{P}_{jk}$. One can see it becomes the Potts model if we require \bar{P}_{ik} be binary (a hard distribution). Such a model favors spatially regular partitions, which is appropriate in the case of spatial positive autocorrelation.

In NEM, the conditional density $f(\mathbf{x}|\Phi)$ takes the form of a mixture model of K components $f(\mathbf{x}|\Phi) = \sum_{k=1}^K \pi_k f_k(\mathbf{x}|\theta_k)$, where $f_k(\mathbf{x}|\theta_k)$ is k -th component’s density function and $p(\mathbf{x}|y = k) = f_k(\mathbf{x}|\theta_k)$. Following [17], NEM’s soft counterpart of $\ln(f(X|Y, \Phi))$ in Eq. (1) can be written as

$$F(\bar{P}, \Phi) = \sum_{i=1}^n \sum_{k=1}^K \bar{P}_{ik} \ln(\pi_k f_k(\mathbf{x}|\theta_k)) - \sum_{i=1}^n \sum_{k=1}^K \bar{P}_{ik} \ln \bar{P}_{ik} \tag{2}$$

Note that maximizing F is also equivalent to maximizing the log-likelihood criterion function in the conventional mixture model [18]. Then, the new objective function in NEM becomes $U(\bar{P}, \Phi) = F(\bar{P}, \Phi) + \beta G(\bar{P})$, where β is a fixed positive

coefficient to weigh the spatial penalty and controls the desired smoothness of output clustering. U can be maximized via the EM procedure, starting from an initial \bar{P}^0 .

1. M-step: With \bar{P}^t fixed, set $\Phi^t = \operatorname{argmax}_{\Phi} U(\bar{P}^t, \Phi)$, which is exactly the same as the M-step in the standard EM for mixture models, for G dose not depend on Φ .
2. E-step: With Φ^t fixed, set $\bar{P}^{t+1} = \operatorname{argmax}_{\bar{P}} U(\bar{P}, \Phi^t)$.

It can be shown that in the E-step, U will be maximized at \bar{P}^* that satisfies Eq. (3), which can be organized as $\bar{P}^* = O(\bar{P}^*)$. It was proven in [3] that under certain conditions, the sequence produced by $\bar{P}^m = O(\bar{P}^{m-1})$ will converge to that solution to maximize U . Hence \bar{P}_{ik}^* can be regarded as dot product between the estimation from its own \mathbf{x} and the estimation from its neighbors.

$$\bar{P}_{ik}^* = \frac{\pi_k f_k(\mathbf{x}_i|\theta_k)\exp\left(\beta \sum_{j=1}^n W_{ij} \bar{P}_{jk}^*\right)}{\sum_{l=1}^K \pi_l f_l(\mathbf{x}_i|\theta_l)\exp\left(\beta \sum_{j=1}^n W_{ij} \bar{P}_{jl}^*\right)} \tag{3}$$

3.3 NEM with Adaptive Coefficient Assignment

EM for the conventional mixture model is not appropriate for spatial clustering since it does not account for spatial information. In contrast, NEM adds in the criterion a spatial penalty weighted by a fixed coefficient β . However, it may not be appropriate to assign a constant coefficient to every site. For those in the class interior, the whole neighborhood is from the same class and hence the site should receive more influence from its neighbors, especially when their posterior estimates are accurate. For those on the class border, because their neighbors are from different classes, its own class membership should be determined mainly by its own explanatory attributes.

Along this line, ANEMI employs a site-sensitive spatial coefficient for the spatial penalty term. In detail, besides the original β that determines the global smoothness in the solution clustering, every site s_i has another coefficient β_i of its own that determines the local smoothness. Then the new penalty becomes $G(\bar{P}) = \frac{1}{2} \sum_{i=1}^n \beta_i \sum_{j=1}^n W_{ij} \sum_{k=1}^K \bar{P}_{ik} \bar{P}_{jk}$. The original G in NEM can be regarded as a special case with $\beta_i = 1$ for all sites. Let U' denote the Lagrangian of U : $U' = U + \sum_{i=1}^n \lambda_i (\sum_{k=1}^K \bar{P}_{ik} - 1)$, which takes into account the constraints on \bar{P}_{ik} . Based on the necessary optimality Kuhn-Tucker conditions, solving $\partial U' / \partial \bar{P}_{ik} = 0$ for \bar{P}_{ik} yields

$$\bar{P}_{ik}^* = \frac{\pi_k f_k(\mathbf{x}_i|\theta_k)\exp\left(\beta \sum_{j=1}^n W_{ij} \frac{\beta_i + \beta_j}{2} \bar{P}_{jk}^*\right)}{\sum_{l=1}^K \pi_l f_l(\mathbf{x}_i|\theta_l)\exp\left(\beta \sum_{j=1}^n W_{ij} \frac{\beta_i + \beta_j}{2} \bar{P}_{jl}^*\right)} \tag{4}$$

Then the estimation is the same as that in NEM, except that we apply Eq. (4) in the E-step.

What remains is to determine β_i . In our implementation, we employ the local Moran's I measure, which evaluates the local spatial autocorrelation at site s_i based on the explanatory attributes inside the neighborhood [19]. Let z_{ip} denote the normalized p -th attribute of site s_i , i.e., $z_{ip} = x_{ip} - \bar{x}_p$, where \bar{x}_p is the global mean of the p -th attribute. Let σ_p denote the global standard deviation of the p -th attribute. Then, for the p -th attribute at site s_i , the local I is defined as $I_{ip} = \frac{z_{ip}}{\sigma_p} \sum_j W_{ij} z_{jp}$, where W is a row-normalized (sum to 1) version of the original binary W . A high I (e.g., $I > 1$) implies a high local spatial autocorrelation at site s_i , which is likely to occur in the class interior. The reverse happens on the border. In ANEMI, β_i is obtained by first averaging I_{ip} over all attributes and then normalizing to $[0, 1]$, i.e., $I_i = \text{mean}_p(I_{ip})$, $\beta_i = \frac{I_i - \min_i\{I_i\}}{\max_i\{I_i\} - \min_i\{I_i\}}$.

3.4 Spatial Augmented Initialization

Like other EM-based algorithms, ANEMI's clustering solution is sensitive to the initial state and hence the study of proper initialization is another focus of this paper. In this paper, we examined three representative methods for clustering initialization: random sampling, K-Means and KKZ. The random sampling method returns K seed centers by uniformly selecting K input instances. For within-cluster scatter minimization, the K-Means algorithm [20] can be regarded as a simplified hard version of EM on Gaussian mixture. While many clustering methods essentially minimize the within-cluster scatter, KKZ [21] is a greedy search method to optimize the complementary between-cluster scatter.

We can see that all initialization methods above only consider normal attributes without accounting for spatial information. If the positive autocorrelation is the major trend within data, then most sites would be surrounded by neighbors from the same class. Based on this observation, we propose to augment feature vector \mathbf{x}_i of site s_i with \mathbf{x}_{Ni} , the average of its neighbors. That is, the augmented vector becomes $\mathbf{x}'_i = [\mathbf{x}_i, \alpha \mathbf{x}_{Ni}]$, where $\alpha > 0$ is a coefficient to weigh the impact of the neighbors, and $\mathbf{x}_{Ni} = \sum_{j=1}^n W_{ij} \mathbf{x}_j / \sum_{j=1}^n W_{ij}$. Then the initialization methods can be run on the augmented $\{\mathbf{x}'_i\}$.

4 Experimental Evaluation

In this section, we first introduce the datasets and the clustering comparison methodology used in our experiments. Then we report comparative results.

4.1 Experimental Datasets

We evaluate ANEMI on five datasets, two synthetic and three real. Some data characteristics are listed in Table 1. The last row gives the spatial smoothness of the target variable y measured with contiguity ratio [19].

The synthetic image datasets are generated in the following way: First, a partition in four classes is simulated from a Potts MRF model with four-neighbor context on a 20×20 rectangular grid. Then, the observations are simulated from this partition based on four Gaussian densities. Fig. 1 shows two sample

Table 1. Some data characteristics

Data	Im1	Im2	Satimage	House	Election
size	400	400	4416	506	3107
#attribute	1	1	4	12	3
#class	4	4	6	n/a	n/a
ratio	0.78	0.84	0.96	0.58	0.61

partitions Im1 and Im2 of different smoothness, together with their observations. The observations for both partitions are drawn from four Gaussian densities: $N(0, 0.5^2)$, $N(1, 0.5^2)$, $N(2, 0.8^2)$, $N(4, 0.8^2)$.

Satimage is a real landcover dataset available at the UCI repository [22]. It consists of the four multi-spectral values of pixels in a satellite image together with the class label from a six soil type set. Because the dataset is given in random order, we synthesize their spatial coordinates and allocate them in a 64×69 grid to yield a high contiguity ratio of 0.96 with four-neighbor context. Fig. 2 illustrates the original partition and a sample obtained partition.

The House dataset records house prices and their environment indices of 506 towns in Boston area [23]. The 12 explanatory variables, such as nitric oxides concentration and crime rate, are used to predict the median value of houses, which is expected to have a small spread in each cluster of a reasonable partition. Fig. 3(a) and (b) show the true house values of 506 towns and their histogram. After normalizing the data to zero mean and unit variance, we fit two Gaussian mixtures, one with two components, the other with four components.

The Election dataset [23] records 1980 US presidential election results of 3107 counties. Originally the three attributes, fraction of population with college degree, fraction of population with homeownership and income, are used to predict voting rate. Here voting rate is used to evaluate clustering performance. Fig. 4(a) and (b) show the voting rates and their histogram. Again, we normalize the data and test two Gaussian mixtures with two and four components respectively.

4.2 Comparison Methodology

We evaluate the clustering quality via two external validation measures. Let C, Y denote the true class label and the derived cluster label, respectively. The conditional entropy $H(C|Y)$ is defined as $H(C|Y) = -\sum_{k=1}^K P_Y(k)H(C|Y = k)$, where probabilities are computed as sample frequencies. Analogously, for the continuous target variable C , we calculate the weighted standard deviation defined as $S(C|Y) = \sum_{k=1}^K P_Y(k)\text{std}(C|Y = k)$, where $\text{std}(\cdot)$ denotes the standard deviation operator and $(C|Y = k)$ denotes the C 's values in cluster $Y = k$. Both measures are minimized to zero in the ideal case.

During experimentation, we concentrate on whether spatial augmented initialization and adaptive coefficient assignment bring any gain in the final clustering quality. For fair comparison, we first compute the augmented version of vectors and then randomly draw K vectors. They are treated as the initial centers returned by the random sampling method on the augmented data. The first half

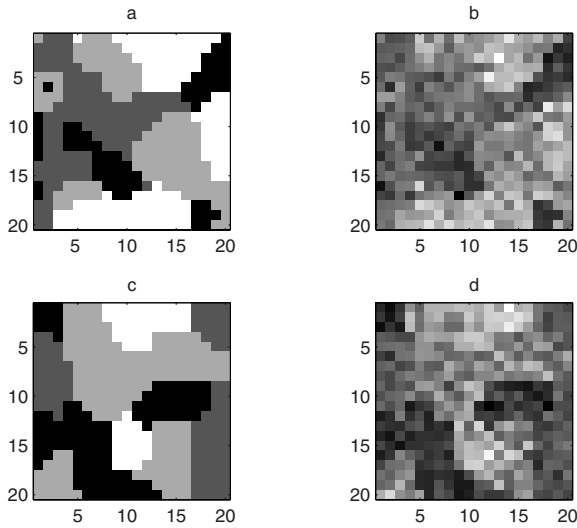


Fig. 1. (a) and (b) show Im1’s true partition and observations, respectively. The counterparts of Im2 are shown in (c) and (d).

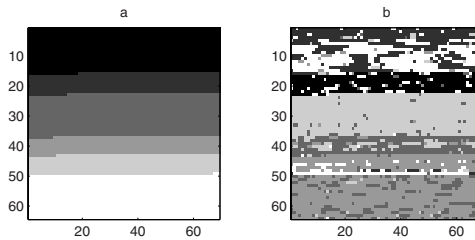


Fig. 2. (a) and (b) show Satimage’s true partitions and a sample clustering

of these vectors are treated as those on original data. These vectors also play the role of initial centers for K-Means that runs 10 iterations. Euclidean distance is used in K-Means and KKZ. In the augmented vector $\mathbf{x}'_i = [\mathbf{x}_i, \alpha \mathbf{x}_{N_i}]$, often $\alpha = 1$ led to the best results, so we only report results with $\alpha = 1$.

The cluster refinement part of ANEMI is built upon an implementation of NEM [24]. Specifically, Gaussian mixture is employed as the model. The global coefficient β is tuned empirically in NEM for each dataset and the obtained value is also used in ANEMI. The number of internal iterations of E-step is set to 10. The outer iteration is stopped when $|(U^t - U^{t-1})/U^t| < 0.0001$. Finally, for each dataset, we report average results of 20 runs.

4.3 Results and Discussions

Initialization. First, we run conventional NEM initialized in both original and augmented spaces. The clustering results are given in Table 2, where the best

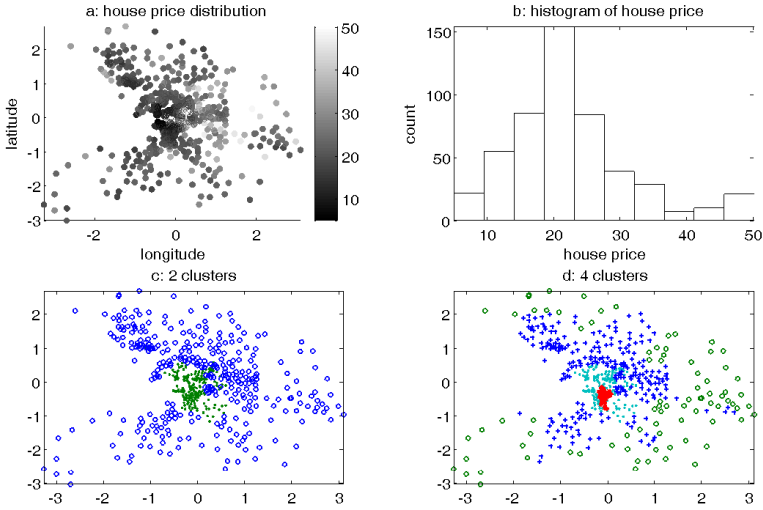


Fig. 3. (a) shows house price distribution in 506 towns in Boston. The corresponding histogram is plotted in (b). Two sample clustering results are shown in (c) and (d).

Table 2. Comparison with different initializations

Data	Sup	Rand	A-Rand	KMeans	A-KMeans	KKZ	A-KKZ
Im1	0.8252	0.9333 ± 0.0556	0.9027 ± 0.0264	0.8306 ± 0.0034	0.8145 ± 0.0117	0.9578	0.8947
Im2	0.8045	0.9900 ± 0.0949	0.9558 ± 0.0874	0.7405 ± 0.0014	0.7540 ± 0.0088	0.9362	0.7834
Satimage	0.6278	0.5970 ± 0.0453	0.6243 ± 0.0516	0.5170 ± 0.0344	0.5142 ± 0.0273	0.8468	0.8630
House:2	n/a	8.0918 ± 0.0230	8.0011 ± 0.0282	8.0633 ± 0.0001	8.0624 ± 0.0001	8.0632	8.0642
House:4	n/a	7.7830 ± 0.1427	7.7768 ± 0.2683	7.8401 ± 0.0806	7.8170 ± 0.0496	7.8145	7.8145
Election:2	n/a	0.1088 ± 0.0023	0.1011 ± 0.0024	0.0968 ± 0.0004	0.0965 ± 0.0001	0.1077	0.1077
Election:4	n/a	0.0992 ± 0.0018	0.0953 ± 0.0023	0.0927 ± 0.0005	0.0919 ± 0.0004	0.1007	0.0962

results are in boldface. “A-X” means initialization method “X” on the augmented data. For instance, the 3rd and 4th columns show the results with random initialization and augmented random initialization. For the three datasets with discrete target variables, we also list in the column “Sup” the results under supervised mode where each component’s parameters are estimated with all data from a single true class. One can see that initialization using augmented data generally brings improvement. The only exception is Satimage with random sampling and KKZ, possibly because its contiguity ratio is so high that almost every site is surrounded by sites from the same class with very similar observations. Thus using augmented data does not make much a difference to the initialization results. Among the three initialization methods, augmented K-Means always leads to the best or sub-optimal results, though the improvement of augmented versions is often more obvious with random sampling and KKZ.

Coefficient Assignment. Since K-Means generally provides the best initialization, we use it to initialize the mixture model for the subsequent comparison of spatial coefficient assignments. Fig. 5 presents the results corresponding to

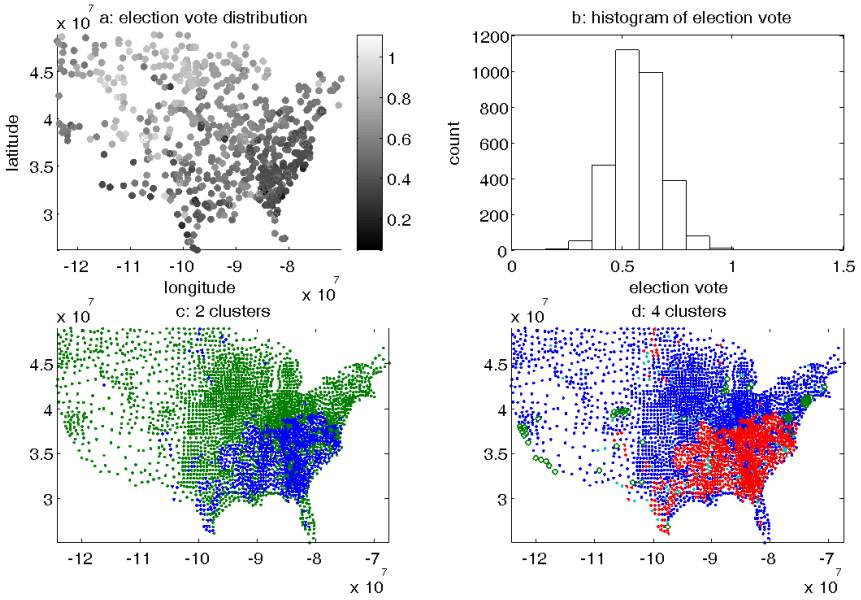


Fig. 4. (a) shows election voting rate distribution in 3107 counties. The corresponding histogram is plotted in (b). Two sample clustering results are shown in (c) and (d).

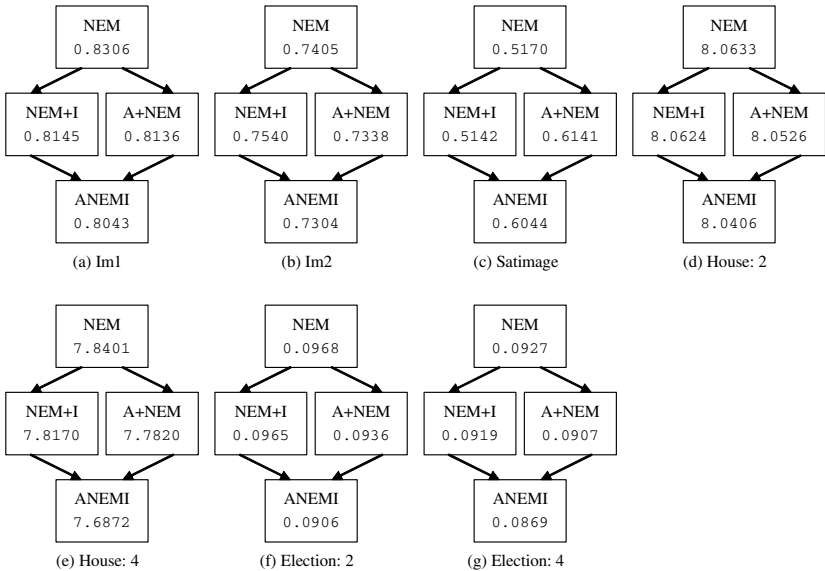


Fig. 5. Impact of spatial augmented initialization and adaptive coefficient assignment

different component combinations: “NEM” denotes conventional NEM with K-Means initialization in the original space; “NEM+I” denotes conventional NEM

with K-Means initialization in the augmented space; “A+NEM” denotes NEM with adaptive coefficient and K-Means initialization in the original space; “ANEMI” denotes NEM with adaptive coefficient and K-Means initialization in the augmented space. One can see that compared to conventional NEM, using site-sensitive coefficients generally yields better results. The only exception is Satimage again. The reasons may be that Satimage’s contiguity ratio is so high that almost every site is surrounded by sites from the same class. Thus it may be always beneficial to assign sites in the neighborhood to the same class. Compared to spatial augmented initialization, the adaptive coefficient assignment has a greater impact on the final clustering quality. The best results are always achieved by combining them two together.

5 Conclusions

In this paper, we introduced an Adaptive Neighborhood Expectation–Maximization with spatial augmented Initialization (ANEMI) algorithm for spatial clustering. ANEMI is an extension of the NEM algorithm, which is built on top of the EM algorithm by incorporating a spatial penalty term into the criterion function. This penalty term is weighed by a spatial coefficient that determines the global smoothness of the solution clustering. Unlike NEM, which assigns an equal weight to every site, ANEMI exploits an adaptive site-sensitive weight assignment scheme, which is determined by the local smoothness inside the neighborhood for each site. In addition, to provide a good initial state for clustering, we proposed to push spatial information early into the initialization methods. Along this line, we also examined three representative initialization methods in the spatial augmented space. Finally, we evaluated the impact of spatial augmented initialization and adaptive coefficient assignment in ANEMI against NEM on both synthetic and real-world datasets. Empirical results showed that with adaptive coefficient assignment, ANEMI using augmented K-Means initialization generally leads to better clustering results than NEM. The gain is most obvious when they are run on datasets with low contiguity ratio.

As for future work, we plan to investigate stochastic versions of NEM to reduce dependence on the algorithm initialization. Also, other optimization techniques, such as genetic algorithms [25], are worth trying to speed up the convergence rate and to improve the final clustering quality further.

Acknowledgments. This work was partially supported by SRF for ROCS, Sci. Tech. Plan Foundation of Guangdong (No. 20070328005), and Sci. Tech. Plan Foundation of Dongguan (No. 2007108101022).

References

1. Tobler, W.R.: Cellular Geography, Philosophy in Geography. In: Gale, W.R., Olson, W.R. (eds.) Reidel, The Netherlands (1979)
2. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Statistical Society B*(39), 1–38 (1977)

3. Ambroise, C., Govaert, G.: Convergence of an EM-type algorithm for spatial clustering. *Pattern Recognition Letters* 19(10), 919–927 (1998)
4. Garey, M.R., Johnson, D.S., Witsenhausen, H.S.: The complexity of the generalized lloyd-max problem. *TOIT* 28(2), 255–256 (1980)
5. Ng, R., Han, J.: CLARANS: A method for clustering objects for spatial data mining. *TKDE* 14(5), 1003–1016 (2002)
6. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: *KDD*, pp. 226–231 (1996)
7. Karypis, G., Han, E.H., Kumar, V.: CHAMELEON: A hierarchical clustering algorithm using dynamic modeling. *Computer* 32(8), 68–75 (1999)
8. Estivill-Castro, V., Lee, I.: Fast spatial clustering with different metrics and in the presence of obstacles. In: *ACM GIS*. (2001) 142 – 147
9. Tung, A.K.H., Hou, J., Han, J.: Spatial clustering in the presence of obstacles. In: *ICDE*, pp. 359–367 (2001)
10. Guo, D., Peuquet, D., Gahegan, M.: Opening the black box: Interactive hierarchical clustering for multivariate spatial patterns. In: *ACM GIS*, pp. 131–136 (2002)
11. Legendre, P.: Constrained clustering. In: Legendre, P., Legendre, L. (eds.) *Developments in Numerical Ecology*. NATO ASI Series G 14, pp. 289–307 (1987)
12. Rasson, J.P., Granville, V.: Multivariate discriminant analysis and maximum penalized likelihood density estimation. *J. Royal Statistical Society B(57)*, 501–517 (1995)
13. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *TPAMI* 6, 721–741 (1984)
14. Solberg, A.H., Taxt, T., Jain, A.K.: A markov random field model for classification of multisource satellite imagery. *IEEE Trans. Geoscience and Remote Sensing* 34(1), 100–113 (1996)
15. Congdon, P.: A model for non-parametric spatially varying regression effects. *Computational Statistics & Data Analysis* 50(2), 422–445 (2006)
16. Hammersley, J.M., Clifford, P.: Markov fields on finite graphs and lattices, unpublished manuscript (1971)
17. Neal, R., Hinton, G.: A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan, M. (ed.) *Learning in Graphical Models*, pp. 355–368. Kluwer Academic Publishers, Dordrecht (1998)
18. Hathaway, R.J.: Another interpretation of the EM algorithm for mixture distributions. *Statistics and Probability Letters* 4, 53–56 (1986)
19. Shekhar, S., Chawla, S.: *Spatial Databases: A Tour*. Prentice-Hall, Englewood Cliffs (2002)
20. Tou, J.T., Gonzalez, R.C.: *Pattern Recognition Principles*. Addison-Wesley, Reading (1974)
21. Katsavounidis, I., Kuo, C., Zhang, Z.: A new initialization technique for generalized lloyd iteration. *IEEE Signal Processing Letters* 1(10), 144–146 (1994)
22. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: *UCI repository of machine learning databases* (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
23. LeSage, J.P.: *MATLAB Toolbox for Spatial Econometrics* (1999), <http://www.spatial-econometrics.com>
24. Dang, V.M.: (1998), <http://www.hds.utc.fr/~mdang/Progs/prognem.html>
25. Pernkopf, F., Bouchaffra, D.: Genetic-based EM algorithm for learning gaussian mixture models. *TPAMI* 27(8), 1344–1348 (2005)

Minimum Variance Associations — Discovering Relationships in Numerical Data

Szymon Jaroszewicz

National Institute of Telecommunications
Warsaw, Poland
s.jaroszewicz@itl.waw.pl

Abstract. The paper presents minimum variance patterns: a new class of itemsets and rules for numerical data, which capture arbitrary continuous relationships between numerical attributes without the need for discretization. The approach is based on finding polynomials over sets of attributes whose variance, in a given dataset, is close to zero. Sets of attributes for which such functions exist are considered interesting. Further, two types of rules are introduced, which help extract understandable relationships from such itemsets. Efficient algorithms for mining minimum variance patterns are presented and verified experimentally.

1 Introduction and Related Research

Mining association patterns has a long tradition in data-mining. Most methods, however, are designed for binary or categorical attributes. The usual approach to numerical data is discretization [22]. Discretization however leads to information loss and problems such as rules being split over several intervals. Approaches allowing numerical attributes in rule consequent have been proposed, such as [3,25], but they do not allow undiscretized numerical attributes in rule antecedent.

Recently, progress has been reported in this area, with a number of papers presenting extensions of the definition of support not requiring discretization [23,14,7]. Other papers provide alternative approaches which also do not require discretization [20,12,19,11,5].

This work extends those methods further, allowing for the discovery of complex nonlinear relationships between sets of numerical attributes without the need for discretization. The work is set in the spirit of association rule mining. First, a concept of minimum variance itemsets is introduced. Those itemsets describe functions which are always close to zero on a given dataset, and thus represent equations describing relationships in data. Based on those itemsets, rules can be derived showing relationships between disjoint sets of attributes. An Apriori style mining algorithm is also presented.

Let us now review the related work. The approach presented in [16] allows for combining attributes using arithmetic operations, but after combining them discretization is applied. Also, since only addition and subtraction are allowed, nonlinear relationships cannot be represented.

In [20,12,19] a method for finding rules of the form “if a linear combination of some attributes is above a given threshold, then a linear combination of another set of attributes is above some other threshold” is described. Rules of this type are mined using standard optimization algorithms. While the approach could be extended to nonlinear case, the method presented here is more efficient since it requires solving eigenvalue problems of limited size instead of using general optimization methods on the full dataset. Furthermore, since binary thresholds are used, the method from [20] cannot represent continuous relationships between groups of attributes. Our work is more in the standard association rule spirit providing both itemsets and rules, as well as an Apriori style mining algorithm.

In [1], an interesting method is presented for deriving equations describing clusters of numerical data. The authors first use a clustering algorithm to find correlation clusters in data, and then derive equations describing the linear space approximating each cluster’s data points based on the cluster’s principal components computed using eigenvectors of the correlation matrix of data in the cluster. While the use of eigenvectors to discover equations may suggest similarities, the approach presented here is quite different. We are not trying to describe previously discovered clusters, but give method of pattern discovery (defining itemsets and rules) in the spirit of association rule mining. Further we allow for arbitrarily complex nonlinear relationships to be discovered, while [1] essentially describes a cluster as a linear subspace. Third, by adding an extra constraint to the optimization, we guarantee that patterns discovered will not involve statistically independent attributes.

There is some similarity between our approach and equation discovery [9,18]. Equation discovery algorithms are in principle capable of discovering minimum variance patterns we propose. However the discovery methodology, is quite different in both cases. In fact our approach was more than an order of magnitude more efficient than Lagrange [9], an equation discovery system. Combining the two approaches, such as using equation discovery to give explicit formulas for minimum variance patterns is an interesting topic for future research.

2 Minimum Variance Itemsets

Let us begin by introducing the notation and some preliminary concepts.

We assume that we are dealing with a dataset D whose attributes are all numeric. Non-numerical attributes can be trivially converted to $\{0, 1\}$ attributes. To avoid overflow problems while computing powers, we also assume that the attributes are scaled to the range $[-1, 1]$.

Attributes of D will be denoted with letters X with appropriate subscripts, and sets of attributes with letters I, J, K . If $t \in D$ is a record of D , let $t.X$ denote the value of attribute X in t , and $t[I]$ the projection of t on a set of attributes I . Following [15,8] we now define support of arbitrary functions. Let f be a function of an attribute set I . Support of f in D is defined as

$$\text{supp}_D(f) = \sum_{t \in D} f(t[I]).$$

We are now ready to describe minimum variance itemsets, the key concept of this work. Our goal is to discover arbitrary relationships between the attributes of D . The patterns we are looking for have the general form

$$f(I) = f(X_1, X_2, \dots, X_r),$$

where we expect the function f to somehow capture the relationship among the variables of $I = \{X_1, X_2, \dots, X_r\}$.

Let us look at two examples. Suppose we have two attributes x and y , such that $x = y$. The equality between them can be represented by an equation

$$f(x, y) = x - y = 0,$$

so one possible function f for this case is $x - y$. Suppose now that x, y represent random points on a circle of radius 1. The function f could now be $f(x, y) = x^2 + y^2 - 1$ since the relationship can be described by an equation $x^2 + y^2 - 1 = 0$. Of course if noise was present the equalities would be satisfied only approximately.

The common pattern of the two above cases is, that the function f was identically equal to zero for all points (records) in the data. It is thus natural, for a given itemset I , to look for a function $f(I)$ which minimizes

$$\sum_{t \in D} [f(t[I])]^2 = \text{supp}_D(f^2).$$

We will call this quantity the *variance* of f on I , or briefly *var*, and a function minimizing it, a *minimum variance function*. This concept should not be confused with statistical notion of variance, which would be around the function's mean (we consciously abuse the terminology).

This formulation has a problem. The function $f(I) \equiv 0$ minimizes variance but does not carry any information. Also $\frac{1}{2}f$ necessarily has lower variance than f , although it does not carry any more information. To avoid such situations, we add a normalizing condition guaranteeing that the function f is of appropriate magnitude. Several such normalizations will be presented below.

2.1 Formal Problem Statement

The above discussion was in terms of arbitrary functions. In practice we have to restrict the family of functions considered. Here we choose to approximate the functions using polynomials, such that the degree of every variable does not exceed a predefined value d . Let $I = \{X_1, \dots, X_r\}$ be a set of attributes. Then any function f of interest to us can be represented by

$$f_{\mathbf{c}}(I) = f(X_1, \dots, X_r) = \sum_{\alpha_1=0}^d \dots \sum_{\alpha_r=0}^d c_{(\alpha_1, \dots, \alpha_r)} X_1^{\alpha_1} \dots X_r^{\alpha_r},$$

where $c_{(\alpha_1, \dots, \alpha_r)}$ are the coefficients of the polynomial. We will organize all coefficients and monomials involved in two column vectors (using the lexicographic ordering of exponents):

$$\begin{aligned}\mathbf{c} &= [c_{(0,\dots,0)}, c_{(0,\dots,1)}, \dots, c_{(d,\dots,d)}]^T, \\ \mathbf{x} &= [X_1^0 \dots X_r^0, X_1^0 \dots X_r^1, \dots, X_1^d \dots X_r^d]^T.\end{aligned}$$

We now have $f_{\mathbf{c}} = \mathbf{c}^T \mathbf{x} = \mathbf{x}^T \mathbf{c}$, and $f_{\mathbf{c}}^2 = \mathbf{c}^T (\mathbf{x}\mathbf{x}^T) \mathbf{c}$. Notice that $\mathbf{x}\mathbf{x}^T$ is a $(d+1)^r \times (d+1)^r$ matrix, whose entries are monomials with each variable raised to power at most $2d$. So the entry in row corresponding to $(\alpha_1, \dots, \alpha_r)$ and column corresponding to $(\beta_1, \dots, \beta_r)$ is $X_1^{\alpha_1+\beta_1} \dots X_r^{\alpha_r+\beta_r}$.

We now use the trick from [15] in order to compute support of $f_{\mathbf{c}}^2$ for various values of \mathbf{c} without accessing the data. Let $t[\mathbf{x}]$ denote the \mathbf{x} vector for a given record t , $t[\mathbf{x}] = [t.X_1^0 \dots t.X_r^0, t.X_1^0 \dots t.X_r^1, \dots, t.X_1^d \dots t.X_r^d]^T$. Now

$$\text{supp}_D(f_{\mathbf{c}}^2) = \sum_{t \in D} \mathbf{c}^T (t.\mathbf{x} \cdot t.\mathbf{x}^T) \mathbf{c} = \mathbf{c}^T \left(\sum_{t \in D} t.\mathbf{x} \cdot t.\mathbf{x}^T \right) \mathbf{c} = \mathbf{c}^T \mathbf{S}_D \mathbf{c}, \quad (1)$$

where \mathbf{S}_D is a $(d+1)^r \times (d+1)^r$ matrix, whose entry in row corresponding to $(\alpha_1, \dots, \alpha_r)$ and column corresponding to $(\beta_1, \dots, \beta_r)$ contains the value of $\text{supp}_D(X_1^{\alpha_1+\beta_1} \dots X_r^{\alpha_r+\beta_r})$. It thus suffices to compute supports of all necessary monomials, after which support of $f_{\mathbf{c}}^2$ for any coefficient vector \mathbf{c} can be computed without accessing the data, using the quadratic form (1).

We now go back to the problem of normalizing $f_{\mathbf{c}}$ such that the trivial solution $f_{\mathbf{c}} \equiv 0$ is avoided. We tried various normalizations:

- require that the vector \mathbf{c} be of unit length, $\|\mathbf{c}\| = 1$,
- require that weighted length of \mathbf{c} be 1, $\sum_{\alpha} w_{\alpha} c_{\alpha}^2 = 1$, this allows for penalizing high degree coefficients.
- require that support of $f_{\mathbf{c}}^2(I)$ be equal to one, under the assumption that all variables in I are distributed uniformly.
- require that support of $f_{\mathbf{c}}^2(I)$ be equal to one, under the assumption that all variables in I are distributed as in D , but are independent.

When no outliers were present, all of those approaches worked reasonably well. However in the presence of outliers only approach (d) was useful. Other methods picked $f_{\mathbf{c}}$ such that it was close to zero everywhere except for the few outlier points. Also, this approach guarantees that patterns involving statistically independent attributes will have high minimum variance.

We thus limit further discussion to normalization based on the requirement (d). Imagine a hypothetical database D_I in which each attribute is distributed as in D but all attributes are independent. The support of $f_{\mathbf{c}}^2$ under such an independence assumption can be computed analogously to (1) as $\text{supp}_{D_I}(f_{\mathbf{c}}^2) = \mathbf{c}^T \mathbf{S}_I \mathbf{c}$, where an element of \mathbf{S}_I in row corresponding to $(\alpha_1, \dots, \alpha_r)$ and column corresponding to $(\beta_1, \dots, \beta_r)$ is given by

$$\text{supp}_{D_I}(X_1^{\alpha_1+\beta_1} \dots X_r^{\alpha_r+\beta_r}) = \text{supp}_D(X_1^{\alpha_1+\beta_1}) \dots \text{supp}_D(X_r^{\alpha_r+\beta_r}),$$

since variables X_1, \dots, X_r are assumed to be independent.

We are now ready to formally define a minimum variance itemset for a given set attributes I :

Definition 1. Let f be a function over a dataset D . The variance of f is defined as $\text{var}(f) = \text{supp}_D(f^2)$.

The minimum variance itemset I is defined as $f^*(I) = f_{\mathbf{c}^*}(I)$, where $\mathbf{c}^T \mathbf{S}_D \mathbf{c} = 1$ and $\mathbf{c}^T \mathbf{S}_I \mathbf{c} = 1$.

2.2 Finding the Minimum Variance Itemset for a Set of Attributes

To find a minimum variance itemset for a given I we use the method of Lagrange multipliers [11]. The Lagrangian is $L(\mathbf{c}, \lambda) = \mathbf{c}^T \mathbf{S}_D \mathbf{c} - \lambda (\mathbf{c}^T \mathbf{S}_I \mathbf{c} - 1)$. Using elementary matrix differential calculus [24,13] we get $\frac{\partial L}{\partial \mathbf{c}} = 2\mathbf{S}_D \mathbf{c} - 2\lambda \mathbf{S}_I \mathbf{c}$, and after equating to zero we get the necessary condition for the minimum:

$$\mathbf{S}_D \mathbf{c} = \lambda \mathbf{S}_I \mathbf{c}. \tag{2}$$

This is the generalized eigenvalue problem [10,24,13], well studied in computational linear algebra. Routines for solving this problem are available for example in LAPACK [10]. If (\mathbf{c}, λ) is a solution to (2), a candidate solution \mathbf{c}' to our optimization problem is obtained by scaling \mathbf{c} to satisfy the optimization constraint: $\mathbf{c}' = \frac{\mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{S}_I \mathbf{c}}}$. Variance of this solution (using substitution and Equation 2) is

$$\text{var}(f_{\mathbf{c}'}) = \text{supp}_D(f_{\mathbf{c}'}^2) = \mathbf{c}'^T \mathbf{S}_D \mathbf{c}' = \frac{\mathbf{c}^T}{\sqrt{\mathbf{c}^T \mathbf{S}_I \mathbf{c}}} \cdot \frac{\mathbf{S}_D \mathbf{c}}{\sqrt{\mathbf{c}^T \mathbf{S}_I \mathbf{c}}} = \frac{\lambda \mathbf{c}^T \mathbf{S}_I \mathbf{c}}{\mathbf{c}^T \mathbf{S}_I \mathbf{c}} = \lambda.$$

The variance of \mathbf{c}' is thus equal to the corresponding eigenvalue, so the final solution \mathbf{c}^* is the (scaled) eigenvector corresponding to the smallest eigenvalue.

The above property can be used to speed up computations, since finding only the smallest eigenvalue can be done faster than finding all eigenvalues (routines for finding a subset of eigenvalues are also available in LAPACK).

Another important observation is that matrices \mathbf{S}_D and \mathbf{S}_I are symmetric (follows directly from their definition) and positive semi-definite (support of a square of a function cannot be negative). This again allows for more efficient computations, see [10,24] for details.

2.3 Example Calculation

We will now show an example calculation on a toy example of a dataset $D = \{(1, -2), (-2, 4), (-1, 2)\}$ over attributes x, y , for $d = 1$. $\mathbf{x} = [1, x, y, xy]^T$, and $\mathbf{c} = [c_{(0,0)}, c_{(1,0)}, c_{(0,1)}, c_{(1,1)}]^T$. Now, $\text{supp}_D(1) = 3$, $\text{supp}_D(x) = -2$, $\text{supp}_D(y) = 4$, $\text{supp}_D(xy) = -12$, $\text{supp}_D(x^2) = 6$, $\text{supp}_D(y^2) = 24$, $\text{supp}_D(x^2y) = 16$, $\text{supp}_D(xy^2) = -32$, $\text{supp}_D(x^2y^2) = 72$. Supports under independence assumption are $\text{supp}_I(y) = \text{supp}_D(x^0) \cdot \text{supp}_D(y) = 12$, $\text{supp}_I(x^2y) = \text{supp}_D(x^2) \cdot \text{supp}_D(y) = 24$, etc. The \mathbf{S}_D and \mathbf{S}_I matrices are

$$\mathbf{S}_D = \begin{bmatrix} 3 & -2 & 4 & -12 \\ -2 & 6 & -12 & 16 \\ 4 & -12 & 24 & -32 \\ -12 & 16 & -32 & 72 \end{bmatrix}, \quad \mathbf{S}_I = \begin{bmatrix} 9 & -6 & 12 & -8 \\ -6 & 6 & -8 & 24 \\ 12 & -8 & 24 & -48 \\ -8 & 24 & -48 & 144 \end{bmatrix}.$$

After solving the generalized eigenvalue problem and rescaling we get $\mathbf{c}^* = [0, -0.5, -0.25, 0]$. The correct relationship $-2x - y = 0$ has been discovered.

Let us now discuss closure properties of minimum variance itemsets.

Theorem 1. $I \subseteq J \implies \text{var}(f^*(I)) \geq \text{var}(g^*(J))$ and $\text{var}(g^*) \leq \text{var}(f^*)$

In other words variance is upward closed, adding attributes reduces the variance. The proof is a trivial consequence of the fact that a function of I is also a function of J (constant in variables in $J \setminus I$), so the lowest variance attainable for J is at least as low as the variance attainable for I , and may be better.

The problem is that we are interested in itemsets with low variance, so if one is found, all its supersets are potentially interesting too. The solution is to set a minimum threshold for variance, and then find smallest (in the sense of set inclusion) sets of attributes for which the variance (of the minimum variance itemset or the itemset’s best equality or regression rule) is less than the specified threshold. Similar approach has been used in [6]. The algorithm is a simple adaptation of the Apriori algorithm [2], and is omitted due to lack of space.

3 From Itemsets to Rules

In order to facilitate the interpretation of minimum variance itemsets two types of rules are introduced. The first kind are what we call

Definition 2. equality rule, $I \cap J = \emptyset$, $I \cup J = D$, $\text{var}(g(I) = h(J)) = \text{supp}_D((g - h)^2)$

Thus equality rules capture relationships between disjoint groups of attributes which are usually easier to understand than the itemsets defined above.

A minimum variance equality rule $g^*(I) = h^*(J)$ is defined, similarly to the minimum variance itemset case above, as a pair of functions for which $\text{var}(g^*(I) = h^*(J))$ is minimum subject to a constraint that the support of $(g - h)^2$ is equal to one, under the independence assumption. Finding minimum variance equality rules for given I and J can be achieved using the same approach as finding minimum variance itemsets. If we approximate both g and h with polynomials, $I = \{X_1, \dots, X_r\}$ and $J = \{X_{r+1}, \dots, X_{r+s}\}$, and denote

$$\begin{aligned} \mathbf{c}_g &= [c_{(0,\dots,0)}, c_{(0,\dots,1)}, \dots, c_{(d,\dots,d)}]^T, \\ \mathbf{x}_g &= [X_1^0 \cdots X_r^0, X_1^1 \cdots X_r^1, \dots, X_1^d \cdots X_r^d]^T, \\ \mathbf{c}_h &= [d_{(0,\dots,1)}, d_{(0,\dots,2)}, \dots, d_{(d,\dots,d)}]^T, \\ \mathbf{x}_h &= [X_{r+1}^0 \cdots X_{r+s}^1, X_{r+1}^0 \cdots X_{r+s}^2, \dots, X_{r+1}^d \cdots X_{r+s}^d]^T, \end{aligned}$$

we get $g = \mathbf{c}_g^T \mathbf{x}_g$, $h = \mathbf{c}_h^T \mathbf{x}_h$, and $g + h = [\mathbf{c}_g | \mathbf{c}_h]^T \cdot [\mathbf{x}_g | \mathbf{x}_h]$. Note that the constant term is omitted from \mathbf{c}_h and \mathbf{x}_h , since it is included in \mathbf{c}_g and \mathbf{x}_g .

From that point on, the derivation proceeds exactly as in the case of minimum variance itemsets in order to find the vector $[c_g|c_h]^*$ which minimizes $\text{supp}_D((g+h)^2)$ subject to $\text{supp}_I((g+h)^2) = 1$. After finding the solution, signs of coefficients in c_h are reversed to get from a minimum variance for $g+h$ to the desired minimum variance for $g-h$.

Finding a minimum variance equality rule on I and J is analogous to finding a minimum variance itemset f on $I \cup J$ subject to an additional constraint that f be a difference of functions on I and J . Thus, the minimum variance of an itemset on $I \cup J$ is less than or equal to the minimum variance of an equality rule on I and J . If an itemset has high minimum variance, we don't need to check rules which can be generated from it, since their variance is necessarily high too.

Another kind of rules are what we call regression rules.

Definition 3. A regression rule is an equality rule $X = g(I)$ where $X \notin I$ and I contains only one attribute.

It is easy to see that regression rules are equality rules with additional constraint that one side of the rule must contain a single attribute in the first power only. It is thus clear that minimum variance of a regression rule cannot be lower than minimum variance of a corresponding equality rule. Also, the definition of variance of a regression rule as well as discovery of minimum variance regression rules are analogous to the case of equality rules and are thus omitted.

Minimum variance regression rules correspond to standard least-squares polynomial regression with X being the dependent variable. Therefore minimum variance equality rules can be seen as a generalization of standard polynomial regression to allow functions of dependent variables, and minimum variance itemset as a further generalization allowing for discovering patterns not involving equality.

4 Illustrative Examples

In this section we show some illustrative examples of patterns discovered, and give some suggestions on how to elicit understandable knowledge from them.

We first apply the method to a small artificial dataset. The dataset has three attributes x, y, z , and is generated as follows: (x, y) are randomly chosen points on a unit circle and z is set equal to x . The relationships among the attributes are therefore $z = x$, $x^2 + y^2 = 1$, and $z^2 + y^2 = 1$.

We applied the algorithm with $d = 2$ without any minimum variance threshold. Only pairs of attributes were considered. Generated patterns are given in the table below (terms with negligibly small coefficients are omitted)

attrs.	min. variance	equation
$\{x, y\}$	$6.62 \cdot 10^{-15}$	$-1.99 + 1.99x^2 + 1.99y^2$
$\{y, z\}$	$6.62 \cdot 10^{-15}$	$-1.99 + 1.99y^2 + 1.99z^2$
$\{x, z\}$	$1.24 \cdot 10^{-17}$	$-0.663x^2 + 1.325xz - 0.663z^2 = -0.663(x-z)^2$

The minimum variance itemsets for $\{x, y\}$ and $\{y, z\}$ do not require any comment. They clearly capture the correct relationship $x^2 + y^2 = 1$.

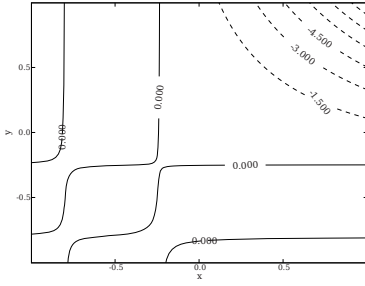


Fig. 1. Contour of the minimum variance itemset for random points satisfying the condition $x < 0 \vee y < 0$

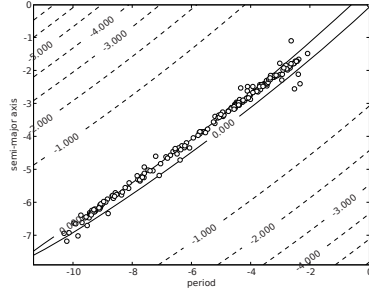


Fig. 2. Extrasolar planet data: relationship between logarithms of planet’s period and semi-major axis, together with contours of the minimum variance itemset. Solid line is the zero contour.

The case for $\{x, z\}$ is more interesting. Instead of the expected $x - z = 0$ we obtained an equivalent, but more complicated expression $(x - z)^2 = 0$. The reason is that the degree of the approximating polynomial exceeds that of the true relationship. As a result, two of the eigenvalues are equal to zero, and any linear combination of their corresponding eigenvectors is also a minimum variance solution. To avoid such situations we recommend decreasing the value of d until a minimum value is found at which the relationship still occurs. In the currently analyzed case lowering d to 1 gives the expected $-0.997x + 0.997z = 0$. Another approach, to use regression rules, which also helps in this case.

It should be noted that the best regression rules for $\{x, y\}$ and $\{y, z\}$ have variance of about 1, so the relationship would not have been discovered by standard regression analysis (indeed the correlation coefficient is about $8 \cdot 10^{-3}$).

Let us look at another example which shows that minimum variance itemsets are able to represent patterns much richer than those usually described using algebraic equations. Consider an artificial dataset which has two attributes $x, y \in [-1, 1]$ and contains points randomly generated on the set where the condition $x < 0 \vee y < 0$ is true. Thus no points are present in the $[0, 1] \times [0, 1]$ square. The correlation coefficient is -0.359 , thus not very high. The minimum variance itemset on xy however, has small values everywhere except for the $[0, 1] \times [0, 1]$ square and the minimum variance of $\{x, y\}$ is 0.024. The representation is of course not perfect, but tends to approximate the data quite well (Figure 1). We will see a similar pattern occurring in real life datasets (sonar) below.

This section shows more examples of minimum variance patterns. The dataset used is about currently known extrasolar planets, and can be downloaded from [21]. Six attributes were chosen and 197 planets selected for which all those attributes were defined. The attributes are described in the table below:

attribute	description
pl. mass	mass of the planet
period	orbital period around star
semi-major axis	distance of the planet from star
ang. distance	angular distance of planet from star (as seen from Earth)
star distance	distance of planet's star from Earth
star mass	mass of the star

Attributes were scaled to $[0, 1]$ range, so units are omitted. Afterwards, logarithmic transform was applied. The advantage of the data is that there are some well established relationships which should be discovered if the method works correctly. This experiment is similar to that from [18], but uses more data and involves additional relationships.

First, semi-major axis divided by the distance of the star from Earth is equal to the tangent of the angular distance of the star from the planet. Second, by Kepler's law, the square of orbital period of a planet is proportional to the cube of the semi-major axis of its orbit. If planet and star masses are known, the proportionality constant can also be determined [17]. It is possible that further relationships exist, but due to the author's lack of astronomical knowledge they will not be discussed. We begin by looking at pairs of attributes. The value $d = 2$ was used, with no minimum variance requirement.

The strongest relationship was discovered between planet's **period** and its **semi-major axis** with minimum variance of $6.83 \cdot 10^{-5}$. The relationship is shown in Figure 2. The data points are marked with circles. Contour plot of the minimum variance itemset is also shown. According to Kepler's law there is a linear relationship between logarithms of the two values. The minimum variance itemset is not linear (due to overfitting and ignoring the star mass) but captures the relationship well. Decreasing the degree or examining rules, reveals the linear nature. The clarity of the relationship is surprising since, planet and star masses also play a role. It turned out, that masses of most stars in the data are very close to each other, and planets' masses are too small to distort the relationship.

To explore the relationship further we examined patterns of size 3 and 4 containing attributes **period** and **semi-major axis**. As expected, the most interesting pattern of length three added the **star mass** attribute (minimum variance $6.85 \cdot 10^{-7}$), and by adding **pl. mass**, a four attribute set was obtained with variance $8.33 \cdot 10^{-10}$ — an almost perfect match.

The triple of attributes which had the lowest variance of $9.72 \cdot 10^{-8}$ was **semi-major axis**, **ang. distance** and **star distance**. This is expected due to the deterministic relationship among them described above. All equality rules involving those attributes had very low variance too. Variance of regression rules was higher (in the range of 10^{-4}).

An interesting subset of the above triple, is the pair **semi-major axis** and **ang. distance**. Its minimum variance is 0.027, but the variance of all rules between those attributes is much higher, about 0.15 in all cases. This is another example of a low variance itemset which cannot be captured by equality rules. The situation is depicted graphically in Figure 3, where data points and the

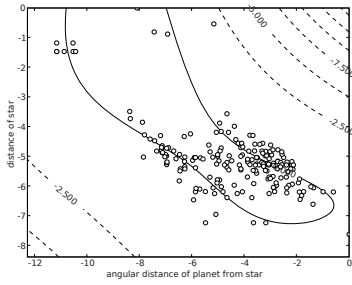


Fig. 3. Relationship between **semi-major axis** and **ang. distance** for the extrasolar planet data, and the corresponding minimum variance itemset. Solid line is the zero contour.

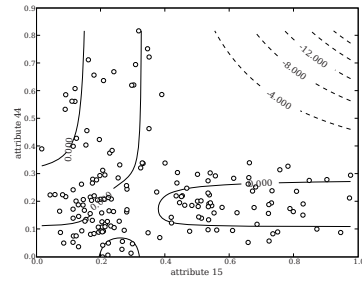


Fig. 4. Data point and contour of the minimum variance itemset for attributes 15 and 44 of the **sonar** dataset

contours of the itemset are shown. It can be seen that there is a clear relationship between the attributes, high values of **semi-major axis** correspond to low values of **ang. distance** and vice versa. But the relationship is not functional, and is not well described by rules. Nevertheless, the minimum variance itemset has values close to zero in the areas where there is a lot of data points. Minimum variance patterns are thus capable of discovering, and describing groupings of data points which are otherwise hard to define.

sonar We now turn our attention to the well known **sonar** dataset. Since our method is somewhat sensitive to outliers, we removed every record which contained a value more than 3 standard deviations from the mean for some attribute. An interesting pattern has been found between attributes 15 and 44, see Figure 4. We can see that high values of both attributes never occur together. The actual relationship is reminiscent of the second artificial dataset presented above. The correlation coefficient is only -0.114 ; based on it, the pattern would have most probably been missed by traditional correlation analysis. This situation is similar to ‘holes in data’ analyzed in [4] which are well approximated in our framework.

5 Performance Analysis

We now present performance evaluation of the minimum variance itemset mining algorithm. The default parameters were $d = 2$ and maximum of $r = 3$ attributes per itemset. We found this combination to be flexible enough to discover complex patterns, which are still reasonably easy to interpret.

We used three datasets for testing: the extrasolar planet and sonar datasets described above, and a large Physics dataset from the KDD Cup 2004 competition with 80 attributes and 50000 records.

The algorithm has been implemented in C. Figure 5 (left) shows the influence of the parameter d on computation time for various minimum variance

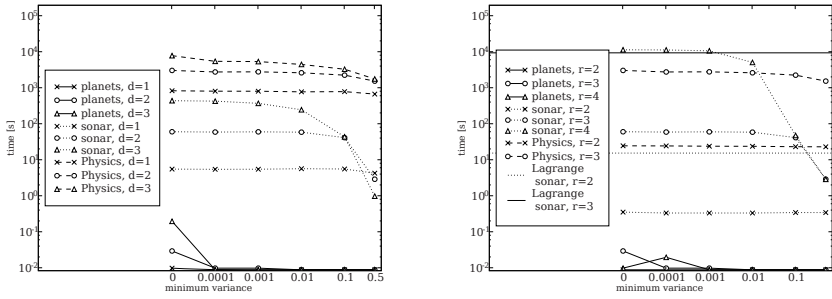


Fig. 5. Computation time vs. degree d (left) and max. itemset size r (right) for various minimum variance thresholds. Horizontal lines: Lagrange algorithm on sonar data.

thresholds. The parameter r is kept equal to the default value of 3. Figure 5 (right) shows the influence of the r parameter (d is kept equal to 2). Note that charts for $d = 2$ (left) and for $r = 3$ (right) in Figure 5 are identical since they correspond to the same parameter values. While performance of the algorithm is worse than for association rules in case of binary attributes (this is to be expected due to a much richer structure of the data), the algorithm is practically applicable even for large datasets. It is interesting to see that, below a certain threshold, the minimum variance parameter has little influence on computation time.

We have also compared our approach with an equation discoverer Lagrange 9 (horizontal lines in Figure 5 (right)). The parameters were set such that it would discover polynomials of degree at most 2 involving at most 2 or 3 variables. Our approach was more than an order of magnitude faster than Lagrange. This is not surprising, as for every set of attributes Lagrange conducts an exhaustive search compared to a single relatively efficient eigenvalue computation in our case.

6 Conclusions and Future Research

A method for discovering arbitrarily complex relationships among numerical attributes has been presented. Its application yields itemsets and rules in the spirit of associations discovery. It has been shown experimentally that the approach does indeed produce interesting patterns, which capture various types of complex relationships present among the attributes. It is capable of finding patterns which would have been missed by standard polynomial regression analysis.

Future work is planned on increasing performance, ... by using bounds for eigenvalues to prune itemsets early.

References

1. Achtert, E., Böhm, C., Kriegel, H.-P., Kröger, P., Zimek, A.: Deriving quantitative models for correlation clusters. In: KDD, Philadelphia, PA, pp. 4–13 (2006)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining association rules between sets of items in large databases. In: SIGMOD, pp. 207–216 (1993)

3. Aumann, Y., Lindell, Y.: A statistical theory for quantitative association rules. In: Proc. of ACM-SIGKDD 1999, San Diego, CA, pp. 261–270 (1999)
4. Ku, L.-P., Liu, B., Hsu, W.: Discovering interesting holes in data. In: International Joint Conference on Artificial Intelligence (IJCAI 1997), pp. 930–935 (1997)
5. Besson, J., Robardet, C., De Raedt, L., Boulicaut, J.-F.: Mining bi-sets in numerical data. In: Dzeroski, S., Struyf, J. (eds.) KDID 2006. LNCS, vol. 4747, pp. 9–19. Springer, Heidelberg (2007)
6. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: Generalizing association rules to correlations. In: SIGMOD, pp. 265–276 (1997)
7. Calders, T., Goethals, B., Jaroszewicz, S.: Mining rank-correlated sets of numerical attributes. In: KDD, pp. 96–105 (2006)
8. Calders, T., Jaroszewicz, S.: Efficient auc optimization for classification. In: Kok, J.N., Koronacki, J., Lopez de Mantaras, R., Matwin, S., Mladenič, D., Skowron, A. (eds.) PKDD 2007. LNCS (LNAI), vol. 4702, pp. 42–53. Springer, Heidelberg (2007)
9. Dzeroski, S., Todorovski, L.: Discovering dynamics: From inductive logic programming to machine discovery. *J. of Intelligent Inform. Systems* 4, 89–108 (1995)
10. Anderson, E., et al.: LAPACK Users' Guide. SIAM, Philadelphia (1999)
11. Fletcher, R.: Practical Methods of Optimization. Wiley, Chichester (2000)
12. Georgii, E., Richter, L., Rückert, U., Kramer, S.: Analyzing microarray data using quantitative association rules. *Bioinformatics* 21(2), ii1–ii8 (2005)
13. Healy, M.: Matrices for Statistics. Oxford University Press, Oxford (2000)
14. Jaroszewicz, S.: Polynomial association rules with applications to logistic regression. In: KDD, pp. 586–591 (2006)
15. Jaroszewicz, S., Korzeń, M.: Approximating representations for continuous data. In: SIAM'DM, pp. 521–526 (2007)
16. Karel, F.: Quantitative and ordinal association rules mining (qar mining). In: Gabrys, B., Howlett, R.J., Jain, L.C. (eds.) KES 2006. LNCS (LNAI), vol. 4251, pp. 195–202. Springer, Heidelberg (2006)
17. Wikipedia: Kepler's laws of planetary motion (retrieved February 26, 2007), http://en.wikipedia.org/wiki/Kepler's_laws_of_planetary_motion
18. Langley, P., Simon, H., Bradshaw, G., Zytkow, J.: Scientific Discovery. Computational Exploration of the Creative Process. MIT Press, Cambridge (1987)
19. Rückert, U., Kramer, S.: A statistical approach to rule learning. In: International Conference on Machine Learning (ICML 2006), Pittsburgh, PA, June 2006, pp. 785–792 (2006)
20. Rückert, U., Richter, L., Kramer, S.: Quantitative association rules based on half-spaces: An optimization approach. In: ICDM, pp. 507–510 (2004)
21. Schneider, J.: The extrasolar planets encyclopaedia, <http://exoplanet.eu>
22. Srikant, R., Agrawal, R.: Mining quantitative association rules in large relational tables. In: ACM SIGMOD Conf. on Management of Data, pp. 1–12 (1996)
23. Steinbach, M., Tan, P.-N., Xiong, H., Kumar, V.: Generalizing the notion of support. In: KDD 2004, Seattle, WA, August 2004, pp. 689–694 (2004)
24. van Loan, C.F., Golub, G.H.: Matrix Computations. Johns Hopkins University Press (1996)
25. Zhang, H., Padmanabhan, B., Tuzhilin, A.: On the discovery of significant statistical quantitative rules. In: KDD, pp. 374–383 (2004)

An Efficient Unordered Tree Kernel and Its Application to Glycan Classification*

Tetsuji Kuboyama¹, Kouichi Hirata², and Kiyoko F. Aoki-Kinoshita³

¹ Center for Collaborative Research, University of Tokyo
4-6-1 Komaba, Meguro, Tokyo 153-8505, Japan
kuboyama@ccr.u-tokyo.ac.jp

² Department of Artificial Intelligence, Kyushu Institute of Technology
680-4 Kawazu, Iizuka 820-8502, Japan
hirata@ai.kyutech.ac.jp

³ Faculty of Engineering, Soka University
1-236 Tangi-cho, Hachioji, Tokyo 192-8577, Japan
kkiyoko@t.soka.ac.jp

Abstract. The problem of computing *unordered tree kernels* based on exhaustive counts of subtrees has known to be $\#P$ -complete. In this paper, we develop an efficient and general unordered tree kernel based on *bifoliate q -grams* that are unordered trees with at most two leaves and just q nodes. First, we introduce a *bifoliate q -gram profile* as a sequence of the frequencies of all bifoliate q -grams embedded into a given tree. Then, we formulate a *bifoliate tree kernel* as an inner product of bifoliate q -gram profiles of two trees. Next, we design an efficient algorithm for computing the bifoliate tree kernel. Finally, we apply the bifoliate tree kernel to classifying glycan structures.

1 Introduction

A *tree structure* is a fairly general data structure that models a wide variety of hierarchical data including parse trees for natural language texts, semi-structured data such as HTML/XML, and biological data such as RNA secondary structures and glycans.

In this paper, we concentrate on a binary classification problem based on kernel methods with support vector machines (SVMs). Let X be the input space (e.g. a set of rooted labeled unordered trees in this paper), and $Y = \{+1, -1\}$ be the output domain. A training set is a finite set of training data, denoted by $D = \{(x_1, y_1), \dots, (x_m, y_m)\} \subsetneq X \times Y$. The purpose of the learning procedure in SVMs is to give a decision function $f_d(\cdot)$ from a training set D . The learning procedure outputs a decision function $f_d : X \rightarrow Y$ so that $y_i = f_d(x_i)$ approximates the probabilistic relation between inputs and outputs.

A number of tree structure classification problems have been successfully addressed by kernel methods with SVMs in the past decade. In order to apply kernel

* This work is partly supported by Grant-in-Aid for Scientific Research No. 17700138 from the Ministry of Education, Culture, Sports, Science and Technology, Japan.

methods to a specific domain, the most important task is to design similarity functions, so called $k(T, T')$, between two objects. One of the earliest work on tree kernels was by Collins and Duffy [4], who presented a $k(T, T')$ as a counting function of common subtrees between two parse trees. Inspired by the parse tree kernel, Kashima and Koyanagi [9] extended it to general rooted labeled ordered trees and proposed a quadratic-time algorithm. These kernels employ the convolution kernel [5] as their design framework by counting all the common subtrees between two trees.

On the other hand, in our previous work [12,13,14,15,16], we introduced an $k_q(T, T')$ as a rooted ordered labeled tree isomorphic to a path. We further proposed a $k_q(T, T')$ [14] and a $k_q(T, T')$ [13] based on the frequencies of all common q -grams embedded in a given tree, which are more efficient and representative than the tree kernels by Kashima and Koyanagi [9].

In contrast to ordered trees, Kashima, Sakamoto, and Koyanagi [10] recently showed that their approach to design kernel functions inherently for $k(T, T')$, in which the order of sibling nodes is arbitrary, leads to #P-completeness. It is also known that the problem of computing the similarity of trees based on $k(T, T')$ [20] and $k(T, T')$ [7] is intractable. On the other hand, Vishwanathan first presented a fast kernel for unordered trees [17] based on a string kernel using suffix trees. Kailing [8] also proposed a tractable algorithm for computing the structural dissimilarity between unordered trees [8]. The effectiveness of these methods, however, has yet to be proven.

In this paper, we aim at developing an expressive and efficient kernel for $k_q(T, T')$ by circumventing the issues in the previous work. In fact, our kernel counts all the common subtrees with q nodes and at most two leaves, as an extension of all the common paths with q nodes (q -grams) [12,13,14,15,16] and restricting to all the common subtrees between two trees [9]. We call such a subtree a q -gram.

Our contributions are as follows: (1) we introduce a $k_q(T, T')$ as a sequence of the frequencies of all bifoliate q -grams embedded in a given tree; (2) we design an efficient algorithm for computing a $k_q(T, T')$ as an inner product of the bifoliate q -gram profiles of two trees; (3) we apply the bifoliate tree kernel to classifying glycan structures in bioinformatics and compare the performance of the bifoliate tree kernel with the kernel based on the structural similarity of unordered trees proposed by Kailing [8].

This paper is organized as follows: in Section 2, we introduce a $k_q(T, T')$ and a $k_q(T, T')$. We also formulate the $k_q(T, T')$ of two given trees as an inner product of their bifoliate q -gram profiles. In Section 3, we design an efficient algorithm $k_q(T, T')$ to compute a bifoliate q -gram profile of a given tree, which runs correctly in $O(qd \min(q, d) \ln n)$ time, where n , d and l are the number of nodes, the depth, and the number of leaves, respectively. This implies that we can also compute bifoliate tree kernels efficiently. In Section 4, we apply the bifoliate tree kernel to classifying glycan structures. Our experimental results illustrate the effectiveness of our kernel. Section 5 concludes the paper by summarizing our contributions.

2 Bifoliate Tree Kernel

We first introduce the basic notions used in this paper. A tree is a connected graph without cycles. For a tree $T = (V, E)$, we sometimes denote $v \in T$ instead of $v \in V$, and $|T|$ instead of $|V|$. A rooted tree is a tree with one node r chosen as its root . For each node v and u in T , let $\text{path}_r(v, u)$ be the unique path from v to u in T .

For a root r of T , we call the number of edges in $\text{path}_r(v)$ the depth of v (in T) and denote it by $\text{depth}_r(v)$. In particular, since $\text{path}_r(r)$ has no edges, we set $\text{depth}_r(r) = 0$. For a tree T , we call $\max\{\text{depth}_r(v) \mid v \in T\}$ the depth of T and denote it by $\text{depth}(T)$.

The parent of $v (\neq r)$ is the node adjacent to v on the path $\text{path}_r(v)$. We say that u is a child of v if v is the parent of u . A leaf is a node having no children, and a bifoliate node is a node having just two children. We denote the number of all leaves in T by $\text{leaves}(T)$.

A rooted tree is $\text{left-to-right ordered}$ if a left-to-right order for the children of each node is given, and it is $\text{right-to-left ordered}$ otherwise. A rooted tree $T = (V, E)$ is ordered (by an alphabet Σ of labels) if there exists an onto function $l : V \rightarrow \Sigma$ such that $l(v) = a$ ($v \in V, a \in \Sigma$). In the remainder of this paper, we simply call a rooted unordered labeled tree and a rooted ordered labeled tree a rooted tree and an ordered tree , respectively.

Let T be an ordered tree with the root v and the children v_1, \dots, v_m of v . The depth sequence (depth sequence , for short) of T is obtained by visiting v_i ($1 \leq i \leq m$) in order, recursively, and then visiting v .

Let T be an ordered tree with n nodes and suppose that the sequence $v_1 \cdots v_n$ is the postorder of T . Also let $p(v_i)$ be the index j such that v_j is a parent of v_i for every $1 \leq i \leq n - 1$. Then, we formulate the depth sequence $D(T)$, the label sequence $L(T)$ and the parent sequence $p(T)$ of T as follows.

$$D(T) = \text{depth}_r(v_1) \cdots \text{depth}_r(v_n), L(T) = l(v_1) \cdots l(v_n), p(T) = p(v_1) \cdots p(v_{n-1}).$$

For the depth sequence D of T , we denote $\max\{d \mid d \in D\}$ by $\max D$. It is obvious that $\text{depth}(T) = \max D$.

Consider the tree T in shown at the top of Figure 1. The depth sequence $D(T)$, the label sequence $L(T)$, and the parent sequence $p(T)$ of T are given below the tree in the figure.

In this paper, as an extension of tree q -grams [12,13,14,15,16], we introduce the concept of q -grams. Note that we are here only concerned with their structures. Thus, their labels are omitted.

Definition 1. A q -gram is a tree with at most two leaves and exactly q nodes, denoted by $P_{k,b}^q$ for $\lfloor q/2 \rfloor \leq k \leq q - 2$ and $0 \leq b \leq k - 1$ and $P_{q-1,0}^q$, where k is the depth of a leaf located relatively far from the root (hereafter called a deeper leaf) and b is the depth of a branch.

Note that the range of the depth b of the branch varies depending on the depth k of the deeper leaf.

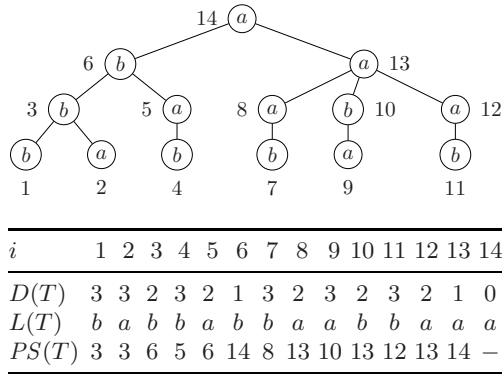


Fig. 1. The tree T and its corresponding depth, label, and parent sequences

Proposition 1. The number of bifoliate q -grams is $\lfloor q/2 \rfloor (q - \lfloor q/2 \rfloor - 1) + 1$.

Let $p = \lfloor q/2 \rfloor$. Note that the depth k of a deeper leaf varies from p to $q - 2$. If $k = q - i$ ($2 \leq i \leq q - p$), then the number of bifoliate q -grams is $q - 2(i - 1)$. Since $i = q - k$, the number of bifoliate q -grams for k ($p \leq k \leq q - 2$) is $2k + 2 - q$. Hence, the number of bifoliate q -grams is $\sum_{k=p}^{q-2} (2k + 2 - q) + 1$. \square

We denote the number $\lfloor q/2 \rfloor (q - \lfloor q/2 \rfloor - 1) + 1$ in Proposition 1 by \tilde{q} .

Proposition 2. Let \succeq be a total order on the set of bifoliate q -grams. Then, for any $1 \leq j \leq \tilde{q}$, the j -th element of \succeq is $P_{k,b}^q$, where $k = q - i$ and $i = q - 2(j - 1) - b + 1$.

It is obvious that the first element of a bifoliate q -gram under \succeq is $P_{q-1,0}^q$. Let j be an integer such that $2 \leq j \leq \tilde{q}$. By Proposition 1, in the case that $k = q - i$, there exist $q - 2(i - 1)$ bifoliate q -grams. Since $P_{k,b}^q$ is the $(k - b - (i - 2))$ -th element from the first element $P_{k,k-i+1}^q$ under \succeq for $k = q - i$, $P_{k,b}^q$ is the

$1 + \left\{ \sum_{l=2}^{i-1} (q - 2(l - 1)) + k - b - (i - 2) \right\}$ -th element from $P_{q-1,0}^q$ under \succeq . By replacing i with $q - k$, we obtain the statement in Proposition 2. \square

Hence, we also denote the j -th bifoliate q -gram under \succeq by Q_j^q ($1 \leq j \leq \tilde{q}$).

All of the bifoliate 5-grams with their depth sequences are described in Figure 2. Here, the deeper leaf is set to the left.

For labeled trees, we denote a bifoliate q -gram by a pair $(Q_j^q, L(Q_j^q))$, where Q_j^q is an ordered tree and $L(Q_j^q)$ is its label sequence. It is obvious that $L(Q_j^q) \in \Sigma^q$.

Definition 2 (Zhang & Shasha [19]). Let T and P be trees. Then, we say that $P \preceq T$ if there exists a bijection f from the nodes of P into the nodes of T satisfying the following conditions.

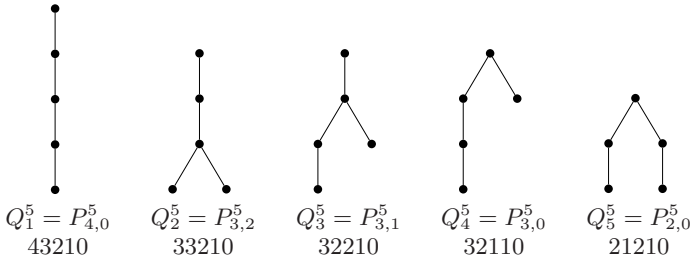


Fig. 2. All of the bifoliate 5-grams

1. f maps the root of P to v .
2. Suppose that f maps x to y and x has children x_1, \dots, x_l . Then, y has children y_1, \dots, y_m such that $m \geq l$ and there exists an injection $g : \{1, \dots, l\} \rightarrow \{1, \dots, m\}$ such that $f(x_i) = y_{g(i)}$.
3. $l(x) = l(f(x))$ for each $x \in P$.

Definition 3. Let T be a tree and (Q_j^q, w) be a bifoliate q -gram for $1 \leq j \leq \tilde{q}$ and $w \in \Sigma^q$. Then, we say that $(Q_j^q, w) \blacktriangleright T$ if there exists a node v in T such that (Q_j^q, w) matches T at v . Furthermore, we denote the number of (Q_j^q, w) embedded into T by $L_q(T)[Q_j^q, w]$.

We order all of the strings in Σ^q by $w_1, \dots, w_{|\Sigma|^q}$. For $1 \leq j \leq \tilde{q}$, we denote the sequence $(L_q(T)[Q_j^q, w_1], \dots, L_q(T)[Q_j^q, w_{|\Sigma|^q}])$ by $L_q(T)[Q_j^q]$.

Definition 4. For a tree T , the following sequence $\mathcal{L}_q(T)$ of the number of every embedded bifoliate q -gram into T is a $|\tilde{q}|$ -tuple of T .

$$\mathcal{L}_q(T) = (L_q(T)[Q_1^q], \dots, L_q(T)[Q_{\tilde{q}}^q]).$$

We are now ready to formulate the inner product of two trees T_1 and T_2 as an inner product of their bifoliate q -gram profiles as follows.

Definition 5 (Bifoliate Tree Kernel). For rooted labeled unordered trees T_1 and T_2 and a fixed integer $q \geq 2$, the $\mathbf{K}_q(T_1, T_2)$ is defined as

$$\mathbf{K}_q(T_1, T_2) = \langle \mathcal{L}_q(T_1), \mathcal{L}_q(T_2) \rangle.$$

For $q = 1$, we assume that $\mathbf{K}_q(T_1, T_2)$ denotes the inner product of the label frequency vectors of T_1 and T_2 .

3 Computing a Bifoliate q -Gram Profile

In this section, we design the algorithm to compute a bifoliate q -gram profile. First, we prepare subroutines as given in Algorithm 1, where D , L and π denote the depth sequence, the label sequence and the parent sequence, respectively, of an ordered tree.

```

procedure pseq( $D$ )
    /*  $D$ : a depth sequence */
    1  $T[0] \leftarrow |D|$ ;
    2 for  $i = |D| - 1$  downto 1 do
    3      $PS[i] \leftarrow T[D[i] - 1]$ ;  $T[D[i]] \leftarrow i$ ;
    4 return  $PS$ ;

procedure labels( $i, k, PS, L$ )
    /*  $PS$ : a parent sequence,  $L$ : a label sequence */
    5  $w \leftarrow \varepsilon$ ;  $pt \leftarrow i$ ;
    6 for  $m = 1$  to  $k$  do
    7      $w \leftarrow w \cdot L[pt]$ ;  $pt \leftarrow PS[pt]$ ;
    8 return ( $w, pt$ );

procedure shift_table( $q, D$ )
    /*  $shift$  is assumed to be an empty array */
    9 for  $d = \max D - 1$  downto 1 do
    10     for  $k = 1$  to  $q - 1$  do
    11         if  $d + k \leq \max D$  then
    12              $shift[d] \leftarrow shift[d] \cup \{(d + k, k)\}$ ;
    13 return  $shift$ ;
    
```

Algorithm 1. Subroutines for computing a bifoliate q -gram profile

The algorithm $pseq(D)$ constructs the parent sequence from a given depth sequence D . The algorithm $labels(i, k, PS, L)$ concatenates the labels from the node indexed by i with length k by selecting nodes and labels according to a parent sequence PS and a label sequence L . “ \cdot ” and ε denote the concatenation of two strings and an empty string, respectively. The algorithm $shift_table(q, D)$ constructs the table $shift(q, D)$, [12,13,14,15,16].

Using these subroutines, we can design the algorithm $bf_profile(q, T, f_k)$ to compute a bifoliate q -gram profile of a given tree described as in Algorithm 2. Here, we use an ordered tree with q nodes isomorphic to a path whose depth of the left leaf is k , and we denote it by P_k^q . We also denote \prec as a lexicographic order on Σ^q . Furthermore, the algorithm adopts the table $indices[j][k]$ in order to store the indices of the left leaf of P_k^p for some $p < q$. We will show below that $p = D[i] + 2k + 1 - j$ for a current depth $D[i]$.

Consider the tree T in Example 1 (Figure 1) and let q be 5. Note first that the result applying the algorithm $bf_profile(q, T, f_k)$ to the depth sequence $D(T)$ is given in Figure 3.

Figure 4 describes the transition of the table $indices$ in the algorithm $bf_profile(q, T, f_k)$. Here, the first and second lines are the depth sequence $D(T)$ of T and index i , respectively. The numbers in bold in the i -th column satisfies the condition of line 7 at the $(i + 1)$ -th iteration of the main loop.

```

procedure Bifoliate_Profile( $q, D, L$ )
  /*  $D$ : a depth sequence,  $L$ : a label sequence */
  /* initialize: Every  $P[k][b][w]$  is assumed to be zero. */
  /* initialize: Every  $id[k][j]$  is assumed to be empty. */
  1   $PS \leftarrow pseq(D)$ ;  $shift \leftarrow shift\_table(q, D)$ ;
  2  for  $i = 1$  to  $|D|$  do
  3    for  $j = \max D$  downto 1 do
  4      for  $k = 1$  to  $\min(j, q - 1)$  do
  5         $p \leftarrow D[i] + 2k + 1 - j$ ;
  6         $s \leftarrow j - k$ ; /*  $s$ : the depth of the root of  $P_k^p$  */
  7        if  $2 \leq p \leq q$  and  $q - p \leq s$  then
  8          foreach  $c \in id[j][k]$  do
  9             $(w_1, \_)$   $\leftarrow labels(c, k, PS, L)$ ;
 10             $(w_2, pt) \leftarrow labels(i, p - k - 1, PS, L)$ ;
 11            if  $(|w_1| < |w_2|)$  or  $(|w_1| = |w_2|$  and  $w_1 \prec w_2)$  then
 12               $(w_1, w_2) \leftarrow (w_2, w_1)$ ;
 13               $w_r \leftarrow L[pt]$ ;  $pt \leftarrow PS[pt]$ ;  $w \leftarrow w_1 \cdot w_2 \cdot w_r$ ;
 14              /*  $pt$ : the index of the root of  $P_k^p$  */
 15              if  $j \neq D[i] + k$  then
 16                /* not  $P_{p-1}^p$  */
 17                 $(w_3, \_)$   $\leftarrow labels(pt, q - p, PS, L)$ ;  $w \leftarrow w \cdot w_3$ ;
 18                 $P[|w_1| + q - p][q - p][w]++$ ;
 19              else if  $p = q$  then  $P[q - 1][0][w]++$ ;
 20          if  $D[i] < \max D$  then
 21            foreach  $(j, k) \in shift[D[i]]$  do
 22               $id[j][k + 1] \leftarrow id[j][k + 1] \cup id[j][k]$ ;
 23               $id[j][k] \leftarrow \emptyset$ ;
 24           $id[D[i]][1] \leftarrow id[D[i]][1] \cup \{i\}$ ;
 25  return  $P$ ;

```

Algorithm 2. $\mathcal{A}_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}^{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$, $f_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$.

Consider the indices 1, 2 and 3 in the third column for index 4. They denote the left leaves of ordered 5-grams whose index of the right leaf is 4.

For index 1 $\in \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}$, the algorithm $\mathcal{A}_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}^{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$, $f_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$ constructs the label sequences $w_1 = l(v_1)l(v_3) = \dots$ and $w_2 = l(v_4)l(v_5) = \dots$. Since $|w_1| = |w_2|$, $w_2 \prec w_1$ and $w_r = l(v_4) = b$, w is set to \dots . Furthermore, it holds that $3 \neq D[4] + 2 = 5$. Since $p = 3 + 2 \cdot 2 + 1 - 3 = 5 = q$, the algorithm $\mathcal{A}_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}^{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$, $f_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$ constructs $w_3 = \varepsilon$ and increments the frequency of bifoliate 5-gram $(P_{2,0}^5, \dots)$, where $|w_1| = 2$.

Moreover, for index 2 $\in \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}$, the algorithm $\mathcal{A}_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}^{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$, $f_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$ constructs the label sequences $w_1 = l(v_2)l(v_3) = \dots$ and $w_2 = \dots$. Since $|w_1| = |w_2|$, $w_1 \prec w_2$ and $w_r = l(v_4) = b$, w is set to \dots by replacing w_1 with w_2 . Similarly to the case for index 1, the algorithm $\mathcal{A}_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}^{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$, $f_{\mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}, \mathcal{V}}$ increments the frequency of bifoliate 5-gram $(P_{2,0}^5, \dots)$.

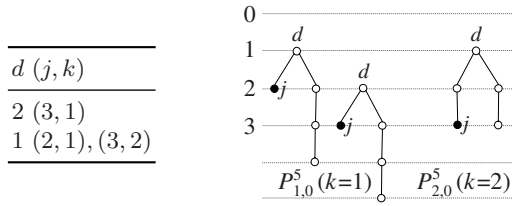


Fig. 3. The table *shift* for $q = 5$ and $\max D = 3$

<i>id</i>	3	3	2	3	2	1	3	2	3	2	3	2	1	0
<i>j</i>	1	2	3	4	5	6	7	8	9	10	11	12	13	14
3	1	1, 2		4			7		9		11			
3	2		1, 2	1, 2	1, 2, 4			7	7	7, 9	7, 9	7, 9, 11		
3	3					1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4	1, 2, 4, 7, 9, 11	
2	1		3	3	3, 5			8	8	8, 10	8, 10	8, 10, 12		
2	2				3, 5	3, 5	3, 5	3, 5	3, 5	3, 5	3, 5	3, 5	3, 5, 8, 10, 12	
1	1					6	6	6	6	6	6	6	6, 13	

Fig. 4. The transition of the table *id* in the algorithm *Bifoliate_Profile*

On the other hand, for index $3 \in \mathcal{V}_2[2][1]$, the algorithm $\mathcal{A}_{\mathcal{V}_2[2][1]}^{f_1}$ constructs the label sequences $w_1 = l(v_3) = b$ and $w_2 = \dots$. Since $|w_1| < |w_2|$ and $w_r = l(v_4) = b$, w is set to \dots by replacing w_1 with w_2 . Furthermore, it holds that $3 \neq D[4] + 2 = 5$. Since $p = 3 + 2 \cdot 1 + 2 - 3 = 4$ and $q - p = 1$, the algorithm $\mathcal{A}_{\mathcal{V}_2[2][1]}^{f_1}$ constructs $w_3 = l(v_6) = a$ and increments the frequency of bifoliate 5-gram $(P_{3,1}^5, \dots)$, where $|w_1| + q - p = 2 + 5 - 4 = 3$.

As a result, we obtain the frequencies of bifoliate 5-grams in T that are non-negative for every $P_{k,b}^q$ as in Figure 5.

$P_{k,b}^5$	$(w, \text{frequency})$
$P_{2,0}^5$	$(babaa, 1)$ $(abaaa, 2)$ $(bbbaa, 1)$ $(bbbab, 1)$ $(baaba, 3)$ $(bbaaa, 2)$ $(baabb, 1)$
$P_{3,0}^5$	$(babaa, 1)$ $(bbbaa, 1)$ $(ababa, 1)$ $(baaba, 2)$ $(abbaa, 1)$
$P_{3,1}^5$	$(bbaba, 1)$ $(babaa, 2)$ $(abaaa, 2)$ $(babba, 1)$ $(ababa, 1)$ $(baaaa, 2)$
$P_{3,2}^5$	$(babba, 1)$

Fig. 5. The frequencies of bifoliate 5-grams in T that are non-negative

Theorem 1. Let T be a tree with $D = D(T)$, $L = L(T)$, $n = |T|$, $d = \max_{v \in T} (|l(v)| + |r(v)|)$, $l = \max_{v \in T} |l(v)|$, and $r = \max_{v \in T} |r(v)|$. Then, the algorithm $\mathcal{A}_{\mathcal{V}_2[2][1]}^{f_1}$ runs in time $O(qd \min(q, d)ln)$.

First, we discuss the correctness of the algorithm $\mathcal{A}_{\mathcal{V}_2[2][1]}^{f_1}$. Consider an ordered p -gram P_k^p with left leaf v and right leaf u , where $j = |l(v)|$ and $d = |l(u)|$. Also let s be $j - k$. Then, it holds that $p = d + 2k + 1 - j$,

and s is the depth of root r of P_k^p (Figure 6(a)). This corresponds to lines 5–6 in the algorithm $\text{find_label_sequences}$.

Let k' be the depth of a deeper leaf of P_k^p . If $q - p > s$, then there exists no bifoliate q -gram $P_{k'+q-p, q-p}^q$. Otherwise, if $q - p \leq s$ (line 7), then the algorithm $\text{find_label_sequences}$ finds the label sequences w_1 on the path from v to the child of r on $\text{root}(v)$ and w_2 on the path from u to the child of r on $\text{root}(u)$ (lines 9–10) using the subroutine find_paths . By comparing the length of w_1 with that of w_2 , the algorithm $\text{find_label_sequences}$ determines which of v and u is a deeper leaf, and it then constructs the label sequence $w_{12r} = w_1 \cdot w_2 \cdot w_r$ (where $w_r = l(r)$) of a bifoliate q -gram $P_{|w_1|, 0}^p$ by setting v (corresponding to w_1) to a deeper leaf (lines 11–12).

Furthermore, if $j \neq d + k$, then it holds that $p \neq q$, so the algorithm $\text{find_label_sequences}$ finds a path from r to r' in Figure 6(b) that is the root of a given tree with length $q - p$ and its label sequence w_3 (line 15). Hence, $w_{12r3} = w_1 \cdot w_2 \cdot w_r \cdot w_3$ is the label sequence of a bifoliate q -gram $P_{|w_1|+q-p, q-p}^q$, and the algorithm $\text{find_label_sequences}$ increments the bifoliate q -gram ($P_{|w_1|+q-p, q-p}^q, w_{12r3}$) in the table P (line 16). Otherwise, if $j = d + k$ and $p = q$, that is, u is the root of an ordered q -gram P_{q-1}^q , then the algorithm $\text{find_label_sequences}$ increments the bifoliate q -gram ($P_{q-1, 0}^q, w_{12r}$) in table P (line 17).

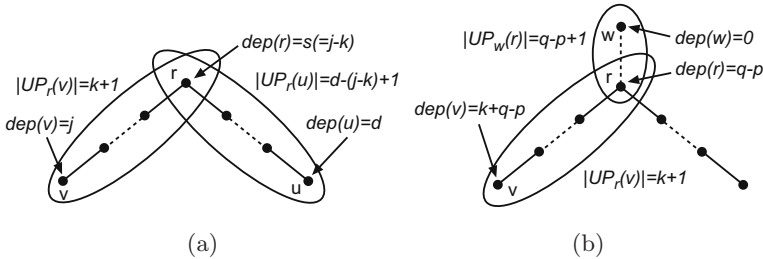


Fig. 6. The relationship of the parameters in P_k^p (left) and $P_{k,b}^q$ (right)

The algorithm $\text{find_label_sequences}$ maintains the indices already searched in a table $\text{visited}[j][k]$. Note that $l \in \text{visited}[j][k]$ means that l is the left leaf of P_k^p . For an index i , the algorithm $\text{find_label_sequences}$ first stores it in $D[i][1]$ (line 22). Next, for every $(j, k) \in \text{visited}[D[i]]$, the algorithm $\text{find_label_sequences}$ shifts the indices in $\text{visited}[j][k]$ to $\text{visited}[j][k+1]$ (lines 19–21), because $D[i]$ and j are the depths of the root and the left leaf of P_k^q , respectively. In this case, the algorithm $\text{find_label_sequences}$ finishes searching for P_k^q and begins searching for P_{k+1}^q .

Next, we consider the running time of the algorithm $\text{find_label_sequences}$. Since $|\text{visited}[j][k]| \leq l$ and $\text{find_paths}(i, k, v, D)$ runs in $O(k)$ time, the running time of the routine from lines 8 to 17 is $O(ql)$. Here, in lines 16 and 17, we use the hash function to increment the element of $P[\cdot][\cdot][w]$ (by encoding a string w as a numeral), so the running time is assumed to be constant. Furthermore, the algorithms $\text{find_label_sequences}(D)$ and $\text{find_label_sequences}(q, D)$ (line 1) run in $O(n)$ and $O(qd)$ time, respectively. Since $|\text{visited}[D[i]]| \leq q$ for every i , the algorithm $\text{find_label_sequences}$ runs in $O(n + (d \times \min(q, d) \times ql + qln)) = O(qd \min(q, d)ln)$ time. \square

Table 1. Summary of the glycan data used in experiments

data set	# of data	avg.# of nodes	avg.height
leukemic cells	192	16.1	6.0
other blood components	294	10.4	5.4
colon cancer	93	7.8	4.2
other colon-related	46	9.7	4.5

4 Experimental Results

In this section, we evaluate the effectiveness of our kernel by empirically comparing its predictive performance in glycan structure classification problems with two other kernels for unordered trees. Glycans are defined as the third major class of biomolecules next to DNA and proteins and play important roles in various fundamental biological processes such as cell-cell interactions. Glycan structures are modeled as either ordered or unordered trees according to its context since the level of appropriate abstractions in modeling the structures depend on the problem to be addressed (cf. [1]). In this paper, we focus on unordered tree modeling of glycans.

We consider the following two competitors to the bifoliate tree kernel. One is the tree kernel by Vishwanathan [17] based on a string kernel, and the other, denoted by $K_H(T_1, T_2)$, is defined based on three simple vectors used in the dissimilarity measure proposed by Kailing et al. [8], which are the vectors of the degree histogram $V_d(T)$, the height histogram $V_h(T)$, and the label histogram $V_l(T)$ for an unordered tree T . We define the kernel $K_H(T_1, T_2)$ for two trees T_1 and T_2 as the sum of the inner products of each pair of vectors.

$$K_H(T_1, T_2) = \langle V_d(T_1), V_d(T_2) \rangle + \langle V_h(T_1), V_h(T_2) \rangle + \langle V_l(T_1), V_l(T_2) \rangle.$$

These kernels were implemented in Ruby and executed on a Windows XP machine with a Pentium M processor running at 1.50 GHz and 750 MB of memory. We used LIBSVM [3] as the SVM implementation, and we computed the area under the ROC curve (AUC) for measuring performance. AUC is the prevailing performance measure for a decision function with a kernel that separates positive examples from negative ones. The AUC values range from 0.5 to 1.0, where the value 0.5 indicates a random separation and the value 1.0 indicates a perfect separation.

The glycan data that we used in the first experiment basically follows Hizukuri et al. [6]; we retrieved the glycan structures from the KEGG/GLYCAN database [11] and used the annotations from the CarbBank/CCSD database [2]. Based on these annotations, we extracted those structures annotated with blood components, labeled as *leukemic cells*, and other non-leukemic blood components (*other blood components*, and *colon cancer*). *Colon cancer* is a cancer of the blood induced by an abnormal proliferation of blood components (usually white blood cells). In

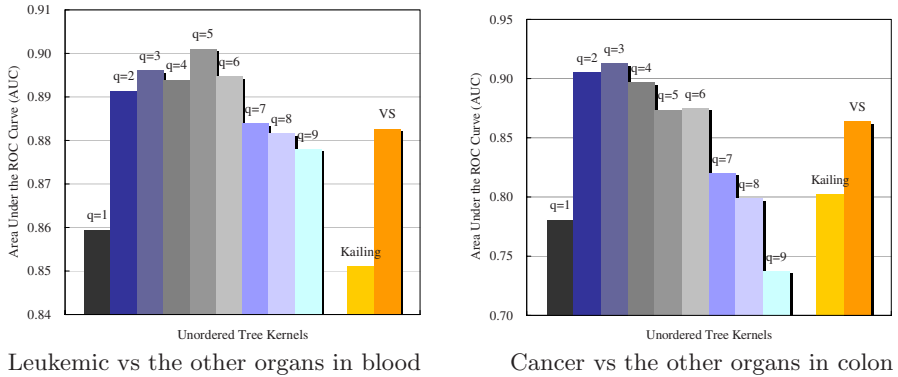


Fig. 7. The areas under the ROC curves for two experiments

the second experiment, we employ two data sets from colon, i.e. glycans related to colon cancer, and others not related to cancer but related to the colon. We retrieved 29 distinct node labels. We have summarized the data used in our experiments in Table 1.

Figure 7 shows the comparison of the results by the proposed method while varying the parameter q . The kernel by Vishwanathan [17] is indicated by “VS”, and the kernel based on dissimilarity proposed by Kailing [8] is indicated by “Kailing.” All of the performance measures were calculated with 5-fold cross validation.

Our tree kernel achieves the best performances at $q = 5$ and $q = 3$ for the leukemia and colon data sets, respectively. The tree kernel due to Vishwanathan also achieves relatively good performance in spite of its restricted expressive power. Since the nodes near the leaves tend to determine the functionalities of glycans, this data set seems to be well-suited to this tree kernel.

Also, it is interesting to see that the value of q achieving the highest predictive performance varies between the two experiments, which indicates that the q size of the most characteristic features varies according to the data set. This corresponds with previous knowledge that structure of glycan biomarkers are varied depending on the cell population being studied.

5 Conclusion

We have presented a novel kernel function for rooted labeled unordered trees. Given two trees, our tree kernel counts the number of common subtrees of size q between them, which are trees with at most two leaves and a fixed number of nodes q . We conducted comparative experiments to illustrate the efficiency of our kernel by applying it to the classification problem of glycan structures. Our kernel outperformed the existing kernels for unordered trees in its predictive performance. The experiments also suggested that the performance depends on the fixed number q , and the optimal value q to give the best performance depends on the data set.

In the future, we plan to design a new tree kernel based on the bifoliate tree kernel so that we can select an appropriate parameter q to achieve better average performance regardless of the data set.

References

1. Aoki, K.F., Ueda, N., Yamaguchi, A., Akutsu, T., Kanehisa, M., Mamitsuka, H.: Managing and analyzing carbohydrate data. *SIGMOD Rec.* 33(2), 33–38 (2004)
2. Doubet, S., Albersheim, P.: CarbBank. *Glycobiology* 2(6), 505 (1992)
3. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001), <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>
4. Collins, M., Duffy, N.: Convolution Kernels for Natural Language. In: Proc. NIPS 2001, pp. 625–632 (2001)
5. Haussler, D.: Convolution Kernels on Discrete Structures, Technical Report UCSC-CRL 99-10 (1999)
6. Hizukuri, Y., Yamanishi, Y., Nakamura, O., Yagi, F., Goto, S., Kanehisa, M.: Extraction of leukemia specific glycan motifs in humans by computational glycomics. *Carbohydrate Research* 340, 2270–2278 (2005)
7. Jiang, T., Wang, L., Zhang, K.: Alignment of trees - an alternative to tree edit. *Theoret. Comput. Sci.* 143, 137–148 (1995)
8. Kailing, K., Kriegel, H.-P., Schönauer, S., Seidl, T.: Efficient similarity search for hierarchical data in large databases. In: Bertino, E., Christodoulakis, S., Plexousakis, D., Christophides, V., Koubarakis, M., Böhm, K., Ferrari, E. (eds.) *EDBT 2004*. LNCS, vol. 2992, pp. 676–693. Springer, Heidelberg (2004)
9. Kashima, H., Koyanagi, T.: Kernels for Semi-Structured Data. In: Proc. ICML 2002, pp. 291–298 (2002)
10. Kashima, H., Sakamoto, H., Koyanagi, T.: Tree Kernels (in Japanese). *J. JSAI* 21(1), 113–121 (2006)
11. Hashimoto, K., Goto, S., Kawano, S., Aoki-Kinoshita, K.F., Ueda, N.: KEGG as a glycome informatics resource. *Glycobiology* 16, 63R–70R (2006)
12. Kuboyama, T., Hirata, K., Ohkura, N., Harao, M.: A q -gram based distance measure for ordered labeled trees. In: Proc. LLLL 2006, pp. 77–83 (2006)
13. Kuboyama, T., Hirata, K., Aoki-Kinoshita, K.F., Kashima, H., Yasuda, H.: A gram distribution kernel applied to glycan classification and motif extraction. In: Proc. GIW 2006, pp. 25–34 (2006)
14. Kuboyama, T., Hirata, K., Aoki-Kinoshita, K.F., Kashima, H., Yasuda, H.: A spectrum tree kernel. *J. JSAI* 22(2), 140–147 (2007)
15. Ohkura, N., Hirata, K., Kuboyama, T., Harao, M.: The q -gram distance for ordered unlabeled trees. In: Hoffmann, A., Motoda, H., Scheffer, T. (eds.) *DS 2005*. LNCS (LNAI), vol. 3735, pp. 189–202. Springer, Heidelberg (2005)
16. Ohkura, N., Hirata, K., Kuboyama, T., Nakano, S., Harao, M.: The gram distribution for rooted ordered trees. In: Proc. LLLL 2006, pp. 69–76 (2006)
17. Vishwanathan, S.V.N.: Kernel Methods: Fast Algorithms and Real Life Applications, PhD thesis, Indian Institute of Science, Bangalore (2002)
18. Yang, R., Kalnis, P., Tung, A.K.H.: Similarity evaluation on tree-structured data. In: Proc. SIGMOD 2005, pp. 754–765 (2005)
19. Zhang, K., Shasha, D.: Tree pattern matching. In: Apostolico, A., Galil, Z. (eds.) *Pattern matching algorithms*, pp. 341–371 (1997)
20. Zhang, K., Statman, R., Shasha, D.: On the editing distance between unordered labeled trees. *Inform. Proc. Let.* 42, 133–139 (1992)

Generation of Globally Relevant Continuous Features for Classification

Sylvain Létourneau¹, Stan Matwin^{2,3}, and A. Fazel Famili¹

¹ Institute for Information Technology, National Research Council Canada, Ottawa
sylvain.letourneau@nrc-cnrc.gc.ca, fazel.famili@nrc-cnrc.gc.ca

² School of Information Technology and Engineering, University of Ottawa, Canada

³ Institute for Computer Science, Polish Academy of Sciences, Warsaw, Poland
stan@site.uottawa.ca

Abstract. All learning algorithms perform very well when provided with a small number of highly relevant features. This paper proposes a constructive induction method to automatically construct such features. The method, named GLOREF (GLOBally RElevant Features), exploits low-level interactions between the attributes in order to generate globally relevant features. The usefulness of the approach is demonstrated empirically through a large scale experiment involving 13 classifiers and 24 datasets. Results demonstrate the ability of the method in generating highly informative features and a strong positive effect on the accuracy of the classifiers.

Keywords: Machine Learning, Attribute Interactions, Feature Extraction.

1 Introduction

Attribute interactions may increase the complexity of a classification task by \mathcal{A} , \mathcal{B} , \mathcal{C} , \mathcal{D} , the instances that belong to the same class across the attribute space. In such cases, the initial attributes, when taken individually, appear to be only remotely related to the class attribute. To uncover the predictive power of such data, the learning systems need to analyze the interacting attributes simultaneously and then build a model that takes into account the interactions observed. As explained by several researchers, this is a complex task that surpasses the ability of many existing machine learning systems.

In particular, Rendell & Seshu [12] emphasizes the fact that current machine-learning techniques rely on the assumption of simple attribute interactions which make them sub-optimal in domains with important attribute interactions. Focusing on the attribute evaluation process, Kononenko & Hong [6] and Bloedorn & Michalski [1] explain that all learning approaches that evaluate the usefulness of each attribute individually using quality measures such as the information gain, the gini-index, the distance measure, or the j-measure are likely to generate inaccurate or too complex models whenever there are important attribute interactions. There have been numerous works on trying to improve the ability of the naive-Bayes with respect to attribute dependencies (e.g., [7]). Specific issues

such as the $\text{logit}(\sigma(\mathbf{w}_i^T \mathbf{x}))$ and the $\text{logit}(\sigma(\mathbf{w}_i^T \mathbf{x}))$ problems with decision trees are also directly related to the lack of capacity of current techniques to deal with attribute interactions [13]. From an applied perspective, it has been argued that attribute interactions are becoming the norm in KDD applications and failing to address this problem adequately has important consequences on the performance obtained [2]. All of these observations call for novel practical techniques that can facilitate learning in domains with important attribute interactions.

This paper proposes such a technique. It is a constructive induction technique that augments the initial representation with new features which make explicit the important information hidden in the interactions among the initial attributes. The new features are self-contained globally relevant features that are suitable for learning algorithms assuming independence. As it will be shown experimentally, the new features can also increase the performance of more complex learning algorithms.

After presenting motivation and related work, the paper introduces the method to derive the new globally relevant features. Sect. 5 offers a large-scale experiment illustrating the usefulness of the approach and the last section concludes the paper.

2 Motivation

In this research, the concept of $\text{relevance}(A)$ designates the usefulness of a given attribute to predict the values of the class attribute. We assume that relevance is computed through a univariate measure such as the gain ratio [11]. Moreover, we use the term $\text{globally relevant}(A)$ to designate an attribute that is relevant over the full training set.

To illustrate the potential effects of attribute interactions on relevance and the usefulness of globally relevant features, let us consider a simple binary classification task with three attributes X_1, X_2 , and X_3 that follow a multivariate normal distribution defined by the following class-conditioned mean vectors and variance-covariance matrix (same for both class values):

$$\mathbf{u}_0 = \begin{bmatrix} 70 \\ 70 \\ 40 \end{bmatrix} \quad \mathbf{u}_1 = \begin{bmatrix} 70 \\ 70 \\ 55 \end{bmatrix} \quad \Sigma = \begin{bmatrix} 650.0 & 0 & -160 \\ 0 & 50 & -115 \\ -160 & -115 & 350 \end{bmatrix}$$

From the mean vectors (\mathbf{u}_0 and \mathbf{u}_1), we conclude that X_3 is the only relevant attribute for this task while Σ indicates that X_1 and X_2 interact with X_3 . Fig. 1(a) shows a simple dataset generated from the above distribution. As seen from the scatter plots of X_3 versus X_1 and X_3 versus X_2 , it is difficult to separate the positive from the negative instances. This difficulty is further illustrated by the class-conditional density curves for X_3 ; the great overlap between the two curves clearly indicates that any decisions based on X_3 will be highly error-prone. The null gain ratio and χ^2 values confirm that, from a univariate global perspective, X_3 appears powerless in predicting the class values.

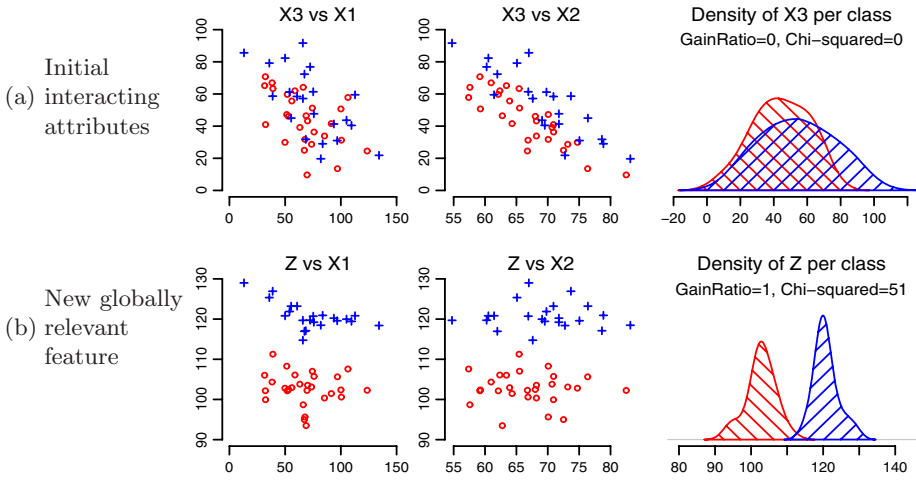


Fig. 1. The effects of interactions on relevance and a globally relevant feature

To uncover the power of the data, we propose a constructive induction method capable of generating a new Z that cancels the negative effects of X_1 and X_2 on X_3 . The new feature is shown in Fig. 1(b). We observe that the transformation removed a great proportion of the initial dispersion since the instances of the same class are now grouped together. As illustrated by the class-conditional density curves for Z , the new feature is highly relevant and its power is observable across the full dataset independently of the other attributes.

The data transformation approach proposed in this paper can automatically generate globally relevant features from complex interactions between any continuous attribute and an arbitrary large number of influencing attributes of possibly different types (continuous, nominal, binary). No information about the underlying distribution of the data or the nature of the interactions is required.

3 Related Work

Related research has been conducted in constructive induction and statistics. A large proportion of the constructive induction techniques are designed to be integrated with existing learning approaches and are not producing a new representation (e.g.:FRINGE [10], AQ17-DCI [1], and OCI [9]). With these systems, the focus is on the improvement of the accuracy of existing methods by opposition to be on the assessment and removal of the negative effects of attribute interactions. Hu [4] noticed the lack of general data pre-processing methods that are independent of specific learning algorithms. Their solution was to propose the GALA systems. These systems generate highly comprehensible features but the types of interactions that it can handle are limited to either prototypical relationships or boolean expressions. The GALA systems do not directly assess the

interactions observed in the data and do not produce a model that describes the effects of these interactions. Recent work by Jakulin & Bratko [5] introduced the notion of \mathcal{V}_i to analyse attribute interactions along with visualization methods. They proposed an experiment showing the benefits of Cartesian product as an approach to resolve the most important interactions.

The topic of interactions has been extensively studied in statistics. PCA, ICA, and contextual normalization methods (e.g., [8]) are examples of methods that have been used in machine learning to help assess the structure of the interactions and produce new features that keep the most important information (according to some criteria). On the other hand, these methods do not rely on the class information, which limit their usefulness for classification tasks [3]. We also notice that most of them can only handle continuous attributes.

In summary, we observe a lack of paradigm-independent supervised constructive induction techniques that directly address the issues of attribute interactions while being capable of handling both continuous and discrete attributes. The GLOREF approach we propose in this paper is an attempt to fulfill this need.

4 The GLOREF Approach

We now describe the GLOREF (GLOBally RElevant Features) approach which we propose for the construction of globally relevant features that account for the initial interactions among the attributes. GLOREF works as a pre-processor and can be used with any standard learning algorithm. The input is a training dataset which contains at least one numerical attribute. The GLOREF approach has two phases: the analysis of relevance and the generation of globally relevant features. The analysis phase computes information to characterize the interactions among the attributes along with their impact on learning. The results of this analysis are stored in data structures named \mathcal{R}_i . The feature generation phase uses the relevance matrices to search for transformation models. Finally, these transformation models are applied to augment the initial data representation and the learning can proceed as usual with the augmented data representation. The following subsections describe the analysis of relevance, the automatic generation of globally relevant features, and application considerations.

4.1 Analysis of Relevance

The analysis of relevance takes as input the training dataset and, optionally, two lists defining the \mathcal{E} and the \mathcal{R} attributes. If these lists are not provided, we simply generate default lists of explanatory and response attributes containing all initial attributes and all initial continuous attributes, respectively¹. As output, the analysis of relevance returns a set of relevance

¹ The use of the terms *response* and *explanatory* attributes follows statistical nomenclature for the analysis of interactions. On the other hand, it is important to notice that the end objective of the proposed method is not to generate new features that approximate the response attributes but instead generate new features that have higher global relevance than any of the initial attributes.

matrices. These matrices provide information on the relevance of the response attributes over partitions based on the explanatory attributes.

The analysis starts by creating a partition of the training dataset $S = \{s_1, s_2, \dots, s_N\}$ for each explanatory attribute. For example, a partition based on a nominal explanatory attribute X with m possible values, noted $\{X(1), X(2), \dots, X(m)\}$ ², generates m subsets S_1, S_2, \dots, S_m where each $S_i = \{s \in S \mid \text{val}_X(s) = X(i)\}$ for $i \in \{1, 2, \dots, m\}$. If the explanatory attribute is continuous, we first discretize it and then partition based on the discretized values instead of the original ones. Since the discretized attributes produced are not going to be used for classification, there is no need to use a supervised discretization technique in this step. A simple unsupervised method such as equal-width or equal-frequency is more appropriate. By default, we use three intervals for discretization. As shown by the experimental results in Sect. 5, this seems to be an adequate choice across a variety of domains although it is likely that even better performance could be obtained by increasing the number of intervals.

The next step computes the relevance information. This step considers one explanatory and one response attributes at a time. To evaluate the effect of the explanatory attribute, we evaluate the relevance of the response attribute in each of the subsets (S_i) and in the full training dataset (S). Following the standard approach to characterize relevance of continuous attributes in decision tree building, we first sort the instances along the response attribute. We then define a split for each observed value of the response attribute in the given set and compute how many examples of each class would fall on each side of the split. Using these numbers, we compute the gain ratio for each possible split. Finally, we define two additional values noted λ_1 and λ_2 that identify the majority class on each side of the split. We name these two values λ_1 and λ_2 since they will be used to determine whether the subsets of the partitions interact in a compatible manner or not (i.e., if they reduce the global relevance or not). All information computed during this step is stored in a set of relevance matrices noted RM_1, RM_2, \dots, RM_m , and RM , where RM_i contains the information computed using subset S_i , and RM the information from S .

To illustrate, let us consider the analysis of the effects of X_2 on the relevance of X_3 for the domain introduced above. First, the partitioning step needs to discretize X_2 . Let us suppose that this discretization did lead to a new attribute $X_{2_discretized}$ with 5 possible values (0, 1, 2, 3, and 4). In this case, 6 relevance matrices would be generated (one for each subset and one for the global dataset). The table on the left hand side in Fig. 2 shows part of the relevance matrix for the subset S_1 , which includes all instances s such that $\text{val}_{X_{2_discretized}}(s) = 0$. There are 18 entries in this relevance matrix which corresponds to the number of distinct values observed for the response attribute X_3 in the given subset. For each cut point, the relevance matrix shows the threshold value, the number of instances per class in each side of the split (columns ‘Cumul.’ and ‘Bal.’), the compatibility characteristics λ_1 and λ_2 ³, and the relevance in terms of gain ratio.

² The missing value (indicated by ‘?’) is considered like any other possible values.

³ The symbol NA indicates that no class is in majority in the given side of the split.

Num	Thresh.	Cumul.	Bal.	λ_1	λ_2	Rel.
1	97.36	{7,10}	{0,1}	1	1	0
2	91.67	{7,9}	{0,2}	1	1	0
3	89.07	{7,7}	{0,4}	NA	1	.0
4	84.45	{7,6}	{0,5}	0	1	.02
		...				
11	76.32	{7,1}	{0,10}	0	1	.5 ◀
		...				
17	57.83	{2,0}	{5,11}	0	1	.0
18	43.43	{1,0}	{6,11}	0	1	.0

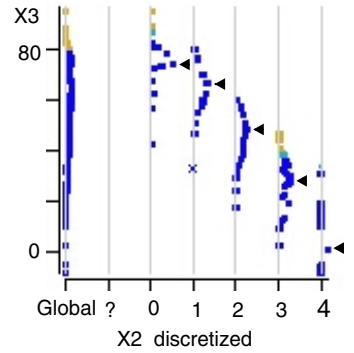


Fig. 2. A relevance matrix and the relevance graph to analyze the effects of X_2 on X_3

The best cut point for this subset (denoted by ◀) is at threshold 76.32 which splits the dataset into two subsets of 8 (7 from 1st class and 1 from 2nd class) and 10 (all from 2st class) instances, respectively.

Visualizing Relevance Matrices and Detecting Harmful Interactions.

The information contained in the relevance matrices for a given pair of attributes can be efficiently visualized through a relevance graph. For example, let us consider the graph in Fig. 2 which shows the effects of X_2 on the relevance of X_3 for the same example. This relevance graph is composed of 6 curves, one for each relevance matrix. The one on the left (named R_{Global}) accounts for the global relevance matrix (i.e., RM) while the following ones (named $R_{0}, R_{1}, R_{2}, R_{3}, R_{4}$) are for the relevance matrices corresponding to the subsets of the partition based on X_2 discretized (i.e., RM_1, RM_2, \dots, RM_5). In particular, the first local relevance curve (labeled '0') corresponds to the relevance matrix shown on the left side. Each point on a given curve represents one entry in the corresponding relevance matrix. The threshold values for the response attribute are shown along the vertical axis. The color (or gray scale) and symbol (e.g., square, cross, plus) of each point designate the compatibility characteristics λ_1 and λ_2 , respectively. There is one color (symbol) for each possible value of λ_1 (λ_2). The relevance of a given point is shown by the horizontal distance that separates it from the vertical reference line located on the left side of each curve. The larger the distance; the better is the cut point in producing pure partitions.

The effect of a given interaction on the global relevance is directly assessed by comparing the relevance of the best cut points (the ones that are the farthest away from their vertical reference line) in the local relevance curves with the relevance of the best cut point in the global relevance curve. If one or more best cut points in local curves are more relevant than the best global cut point, then the interaction has a negative effect on the global relevance of the response attribute. The relevance graph in Fig. 2 illustrates this situation since several of the most relevant cut points in the local relevance curves (indicated on the graph by ◀) are more relevant than the best global cut point.

4.2 Automatic Generation of Globally Relevant Features

A key idea behind the GLOREF approach comes from the observation that the global relevance of the response attribute can be modified by altering the alignment of the local relevance matrices. Such a re-alignment can be accomplished by modifying the values of the response attribute within each local relevance matrix by a value ω_i for $i = 1, \dots, m$. The result is a new feature Z defined as

$$Z = \Gamma(X, Y) = \begin{cases} Y + \omega_1 & \text{if } X = X(1) \\ Y + \omega_2 & \text{if } X = X(2) \\ \vdots & \\ Y + \omega_m & \text{if } X = X(m) \end{cases} \quad (1)$$

where Y is the response attribute, $X(1), X(2), \dots, X(m)$ are the distinct values for the explanatory attribute X , and $\{\omega_1, \omega_2, \dots, \omega_m\}$ are the parameter values of the model. The objective is to set the ω_i values in a way that maximizes the global relevance of the new feature. We first present the algorithm developed to resolve this optimization problem and then introduce the approach to cope with interactions involving several explanatory attributes.

Univariate Transformations. A brute force solution to select the parameter values $\{\omega_1, \omega_2, \dots, \omega_m\}$ is to evaluate all possible alignments of the local relevance curves and select the alignment with the best global relevance. Recognizing that the number of possible alignments is exponential in the number of relevance curves, this solution would not be practical in most real world applications. We therefore introduce the heuristic approach described in Fig. 3.

The algorithm starts by handling a special case that happens when all the most relevant cut points in the various relevance matrices are compatible (equal values for both compatibility characteristics λ_1 and λ_2). In this case, the algorithm directly returns the optimal solution which aligns these most relevant cut points on an arbitrary threshold noted T^* ⁴. The relevance graph in Fig. 2 illustrates this situation since all maximally relevant cut points are compatible (same color and same symbol). When the most relevant cut points are not all compatible, the algorithm proceeds with a greedy search. This search gradually builds the complete solution by combining local solutions. It starts by finding the best alignment between the first two relevance matrices and store the result into a temporary relevance matrix noted RM_{cum} . In the following iteration, it combines RM_{cum} with the third relevance matrix and so on until all local relevance matrices have been processed. There are three steps in each iteration of the search procedure: reduction of the two relevance matrices to be considered (**ReduceRM**), search for the best local alignment (**UnivExhaustiveSearch**), and update of the current solution (**ComputeGlobalRM**). In the first step, **ReduceRM** removes many of the cut points from the two relevance matrices considered in order to reduce

⁴ By default, the algorithm sets the ω_i values such that the best global cut point will be at threshold value 0.

Algorithm UnivGLOREF

Input: The set of relevance matrices RM_1, RM_2, \dots, RM_m for pair of attributes

Output: A set of values $\{\omega_1, \omega_2, \dots, \omega_m\}$ maximizing the relevance of a new feature.

if best cut points from all subsets S_i have identical λ_1 and λ_2

$\{T_1^*, T_2^*, \dots, T_m^*\} \leftarrow$ best cut point thresholds in $\{RM_1, RM_2, \dots, RM_m\}$

$\Omega^* \leftarrow \{-T_1^*, -T_2^*, \dots, -T_m^*\}$

else

$\{\omega_1, \omega_2, \dots, \omega_m\} \leftarrow \{0, 0, \dots, 0\}$, $RM_{cum} \leftarrow RM_1$

For $i = 2$ to m

/* Simplify current relevance matrices */

$RM_{cum} \leftarrow$ ReduceRM(RM_{cum}), $RM_i \leftarrow$ ReduceRM(RM_i)

/* Find current best solution and update previous solution */

$\{\omega, \omega_i\} \leftarrow$ UnivExhaustiveSearch(RM_{cum} , RM_i)

For $j = 1$ to $i - 1$ $\omega_j \leftarrow \omega_j + \omega$

/* Compute global relevance info for current partial solution */

$RM_{cum} \leftarrow$ ComputeGlobalRM($\{RM'_{i-1}, RM'_i\}$, $\{\omega_{i-1}, \omega_i\}$)

$\Omega^* \leftarrow \{\omega_1, \omega_2, \dots, \omega_m\}$

return Ω^*

Fig. 3. Heuristic to efficiently generate univariate GLOREF features

the number of potential alignments to evaluate. Precisely, it removes all entries except the most relevant cut point for each observed combination of λ_1 and λ_2 and the two points with minimal and maximal thresholds. In the second step, `UnivExhaustiveSearch` evaluates all potential alignments of the two reduced relevance matrices and returns the two ω values that maximize the global relevance of a new feature that would be created by combining the subsets considered. Finally, `ComputeGlobalRM` updates the current solution by adding the new ω values to the previous global solution. Once all relevance matrices have been considered, the heuristic returns the set of parameter values $\{\omega_1, \omega_2, \dots, \omega_m\}$ selected for the generation of a new globally relevant feature (Eq. 11).

Multivariate Transformations. The direct extension of the univariate solution to handle the multivariate case would require a multivariate partitioning of the initial dataset along with the analysis of the resulting combinatorial number of subsets. Efficiency concerns and the risk of having to proceed with insufficient data in the various subsets call for an alternative method. Accordingly, we propose an inductive process where each phase has two steps: $\dots \rightarrow \dots, \omega_{i-1}, \omega_i$ and $\dots \rightarrow \dots, \omega_i, \omega_{i+1}$.

The generation step constructs features in progressive order of complexity by combining pairs of features from the previous phase. In the first phase, it

uses the univariate transformations to create multivariate features with two explanatory attributes. For instance, if there are two univariate transformations $Z_1 = \Gamma(X_1, Y)$ and $Z_2 = \Gamma(X_2, Y)$, then the generation step in the initial phase would create a new multivariate feature $Z = \Gamma(\{X_1, X_2\}, Y)$. In the second phase, the generation step uses the selected features from the first phase to create features involving either three or four explanatory attributes, and so forth. Each multivariate feature is constructed through an iterative optimization process. Precisely, to construct a multivariate feature Z involving l explanatory attributes X_1, \dots, X_l and a response attribute Y we repeat the following steps

1. Using the univariate procedure described above, compute for each X_i a set of parameter values noted Ω_i^t that optimizes the relevance of $\Gamma(X_i, Z^{t-1})$.
2. Update the values of the new feature using

$$Z^t = \sum_{i=1}^l \Gamma(X_i, Z^{t-1}; \Omega_i^t) - (l-1) * Y \quad (2)$$

where $t > 1$, $Z^0 = Y$, and $\Gamma(X_i, Z^{t-1}; \Omega_i^t)$ is equivalent to **(II)** with the parameter values specified by the set Ω_i^t . The repeated summation allows us to jointly realign the univariate relevance curves in a way that maximize the relevance of the new feature. The process stops when there is no significant improvements in the global relevance of Z between two iterations or when a maximal number of iterations has been performed. In practice, only a few iterations are required to converge (between two and five in most cases). This process ensures that the number of parameters to estimate grows linearly with the number of explanatory attributes and avoids the multivariate partitioning issues mentioned above. The reuse of the efficient univariate heuristic presented above further improve the performance of the approach.

The feature selection determines which features are allowed to proceed to the next phase of the inductive process. To be selected, a new multivariate feature must have a higher global relevance than any of the attributes involved in its creation. To control the risk of overfitting, we use only 70% of the training data during the creation of the features and keep the remaining part for the feature selection step. The inductive process stops when less than two new features are selected for the following iteration. Finally, all univariate transformation models and all selected multivariate ones are applied to augment the initial representation with globally relevant features. We notice that the overall computational complexity of the approach is polynomial in the number of features provided as input to each iteration. By applying feature selection prior to each iteration, we ensure that the approach stays practical regardless of the number of initial attributes.

4.3 Application Issues and Smoothing of Transformations

When computing the values for the new features, two issues may arise: missing values and unseen values. Missing values might be observed for one or more of

Table 1. Global relevance of the best initial attribute and GLOREF feature

Dataset	Initial GR	GLOREF				Dataset	Initial GR	GLOREF			
		Type	GR	Diff (%)	Type			GR	Diff (%)		
autos	.55	Mul	.92	.37	(69 %)	N1F1	.29	Mul	.63	.34	(117 %)
balance-scale	.17	Mul	.67	.49	(286 %)	N1MN	.15	Mul	.29	.14	(91 %)
breast-w	.55	Mul	.86	.31	(57 %)	N2F1	.59	Mul	.79	.20	(34 %)
cars	.44	Mul	.54	.10	(23 %)	N2MN	.17	Uni	.32	.15	(88 %)
colic	.28	Mul	.38	.10	(36 %)	N3F1	.56	Mul	.90	.34	(60 %)
credit-a	.42	Mul	.47	.05	(12 %)	N3MN	.17	Mul	.41	.24	(145 %)
diabetes	.18	Mul	.30	.12	(64 %)	N4F1	.31	Mul	.84	.53	(169 %)
glass	.80	Mul	.98	.17	(21 %)	N4MN	.17	Mul	.98	.81	(488 %)
heart-statlog	.35	Mul	.59	.24	(68 %)	N5F1	.38	Mul	.58	.20	(53 %)
hepatitis	.33	Mul	.55	.22	(65 %)	N5MN	.19	Mul	1.0	.81	(435 %)
ionosphere	.50	Mul	.73	.22	(44 %)	N6F1	.27	Mul	.53	.26	(97 %)
liver	.05	Mul	.31	.26	(482 %)	N6MN	.16	Mul	.41	.25	(163 %)

the explanatory attributes or for the response attribute. The former case does not cause any problem as our implementation treats this situation explicitly by including the missing value as one of the potential values for all explanatory attributes. However, if the response attribute has a missing value then the new feature would also need to have a missing value. The problem of unseen values arises when the model tries to process an instance for which the observed explanatory attribute value has not been seen during the generation of the transformation model. Since the given value was not part of the training dataset, the models do not include an entry for this value and therefore there is no corresponding ω parameter value. In this case, the value of the new feature equals the value of the response attribute (i.e., no transformation).

The discretization of continuous explanatory attributes may introduce unnecessary discontinuities in the new features. We avoid this problem by smoothing the ω values when applying transformations that involve one or more continuous explanatory attributes. We use the inverse distance weighting smoothing method to adjust the ω values based on the observed values of the explanatory attribute(s).

5 Experimental Evaluation

To evaluate the feasibility of the GLOREF approach, we propose a large-scale experiment involving 24 datasets (12 artificial and 12 from the UCI repository) and 13 classifiers implemented in the WEKA package. The artificial datasets contain numerical attributes only with pre-defined univariate and simple multivariate interactions. Several of the UCI datasets contain a mix of continuous and discrete attributes. The maximal number of attributes is 35. We followed the 10-fold cross-validation methodology. In each fold, we performed the following tasks: (1) apply GLOREF on the training data to learn univariate and multivariate transformation models, (2) use these models to augment the initial representation with GLOREF features, (3) for each learning system, learn a model using only the initial attributes and another model using the augmented representation, and (4) evaluate the accuracy of the two models on test data.

Classifier	Accuracy improvement				Better (Significant)	Worse
	Avg	Std	Min	Max		
BaggingDT	1.10	3.67	-3.8	11.2	16 (6)	7 (0)
BoostingDT	1.01	3.82	-4.1	14.6	13 (5)	10 (0)
DecisionStump	9.87	9.81	-4.5	32.8	22 (13)	2 (0)
DecisionTable	5.33	7.41	-3.5	21.5	18 (12)	6 (0)
HyperPipes	24.4	17.4	-6.0	61.2	22 (22)	2 (1)
IB1	3.01	3.49	-1.9	10.5	18 (7)	4 (0)
IB5	2.70	3.10	-3.0	8.40	9 (5)	3 (0)
J48	2.87	4.89	-5.6	12.9	17 (10)	7 (0)
KernelDensity	2.00	3.97	-5.8	10.1	13 (8)	10 (0)
NaiveBayes	4.82	6.21	-3.7	18.7	19 (11)	5 (1)
OneR	10.5	10.9	-1.3	34.9	19 (16)	5 (0)
PART	3.27	3.73	-4.6	10.0	20 (8)	4 (0)
SMO	1.78	4.09	-3.0	16.3	12 (2)	6 (0)
Artificial	7.54	9.75	-1.2	47.5	135(95)	17(0)
UCI	3.79	9.54	-6.0	61.2	83(30)	54(2)
All	5.78	9.82	-6.0	61.2	218(125)	71(2)

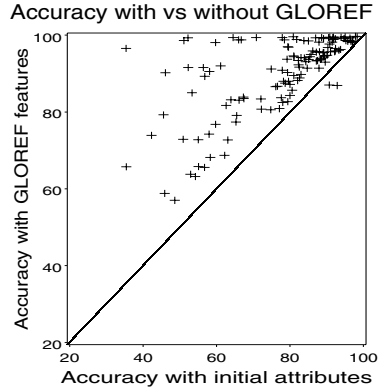


Fig. 4. The effects of GLOREF features on accuracies

We first consider the ability of GLOREF to produce new globally relevant features by comparing the expected gain-ratio of the best initial attribute and the best GLOREF feature. We compute the expected gain-ratio of the best initial (resp. GLOREF) attribute by averaging the gain-ratios of the best initial (resp. GLOREF) attribute based on test data from the various folds of the cross-validation procedure. Table 1 presents the results. The gain ratio for the best GLOREF feature is systematically higher than the one for the best initial attribute. The standard t-test to compare group means reveals that all increases are statistically significant at the 0.05 level. The relatively large variation in percentage of increase (from 12% to 488%) suggests that the datasets are not all equally affected by the problem of attribute interactions. We repeated the analysis using the χ^2 measure and obtained consistent results. Therefore, we conclude that the GLOREF approach succeeded in producing new highly globally relevant features.

The graph on the right side of Fig. 4 offers a quick view of the usefulness of the new features for learning. There is one point for each combination of learning system and dataset for which the use of GLOREF features significantly changed the accuracy. All points located above the diagonal line indicate positive results and inversely for the points located below. The table on the left side details the results by classifier. The first four columns provide the statistics on increase in accuracy due to the GLOREF features while the last two columns count the number of better and worse results with statistically significant results in parentheses (the number of datasets for which the addition of the GLOREF features did not change the results equals the difference between 24 and the sum of the ‘Better’ and ‘Worse’ columns). Out of the 312 experiments (13 classifiers * 24 datasets), 127 lead to a significant difference in accuracy and only 2 of these are on the negative side. As expected, learning systems which are powerless with respect to attribute interactions such as HyperPipes, OneR, and DecisionStump profited the most from the GLOREF features with average increase in accuracy of 24%, 10.5% and 9.8%, respectively. Focusing on statistically significant results,

we notice that all classifiers have been positively affected by the GLOREF features, with the number of statistically significant wins varying from 2 to 22 over 24. Moreover, the column ‘Max’ clearly shows that complex approaches such as bagging, boosting and support vector machine (SMO) can also greatly benefit from highly globally relevant features. The relatively important standard deviations tend to confirm the heterogeneity of the selected datasets. Finally, by analyzing the results by datasets, we observe that the levels of increase in accuracy tend to match the increase of global relevance between the best GLOREF and best initial feature. In other words, large improvements in global relevance generally result in high increases in accuracy and inversely.

6 Conclusion

This paper links the problem of attribute interactions to the concept of attribute relevance. After discussing the potential effects of interactions on relevance, we introduce the GLOREF method to model interactions and construct new globally relevant features. The autonomous solution is evaluated through a large-scale experimentation involving 24 datasets and 13 learning systems. The analysis of the relevance of the new features shows that the GLOREF system generates highly globally relevant features for all datasets, with some increases in gain ratio that are close to 500%. Adding the GLOREF features to the initial representation significantly improved the accuracy in more than 40% of the experiments, while reducing it in only less than 1%. Although these results are strongly positive, it is possible that the heuristics proposed are not optimal. Future work will investigate alternative heuristics to further improve performance.

References

- [1] Bloedorn, E., Michalski, R.S.: Data-driven constructive induction. *IEEE Intelligent Systems and their Applications* 13(2), 30–37 (1998)
- [2] Freitas, A.A.: Understanding the crucial role of attribute interaction in data mining. *Artificial Intelligence Review* 16, 177–199 (2001)
- [3] Fukunaga, K.: *Introduction to Statistical Pattern Recognition*, 2nd edn. Academic Press, NY (1990)
- [4] Hu, Y.-J.: *Representational Transformation Through Constructive Induction*. PhD thesis, University of California, Irvine (1999)
- [5] Jakulin, A., Bratko, I.: Analyzing attribute dependencies. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) *PKDD 2003*. LNCS (LNAI), vol. 2838, pp. 229–240. Springer, Heidelberg (2003)
- [6] Kononenko, I., Hong, S.J.: Attribute selection for modelling. *Future Generation Computer Systems* 13, 181–195 (1997)
- [7] Langley, P.: Induction of recursive Bayesian classifier. In: Brazdil, P.B. (ed.) *ECML 1993*. LNCS, vol. 667, pp. 152–164. Springer, Heidelberg (1993)

- [8] Létourneau, S., Famili, A.F., Matwin, S.: A normalization method for contextual data: Experience from a large-scale application. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 49–54. Springer, Heidelberg (1998)
- [9] Murthy, S.K., Kasif, S., Salzberg, S.: A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research* 2, 1–33 (1994)
- [10] Pagallo, G., Haussler, D.: Boolean feature discovery in empirical learning. *Machine Learning* 5(1), 71–99 (1990)
- [11] Quinlan, J.R.: Induction of decision trees. *Machine Learning* 1, 81–106 (1986)
- [12] Rendell, L.A., Seshu, R.: Learning hard concepts through constructive induction: Framework and rationale. *Computational Intelligence* 6(4), 247–270 (1990)
- [13] Vilata, R., Blix, G., Rendell, L.A.: Global data analysis and the fragmentation problem in decision tree induction, pp. 312–326 (1997)

Mining Bulletin Board Systems Using Community Generation

Ming Li¹, Zhongfei (Mark) Zhang², and Zhi-Hua Zhou¹

¹ National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China

² Computer Science Department, SUNY Binghamton, Binghamton, NY 13902, USA
{lim, zhouzh}@lamda.nju.edu.cn, zhongfei@cs.binghamton.edu

Abstract. Bulletin board system (BBS) is popular on the Internet. This paper attempts to identify communities of interest-sharing users on BBS. First, the paper formulates a general model for the BBS data, consisting of a collection of user IDs described by two views to their behavior actions along the timeline, i.e., the topics of the posted messages and the boards to which the messages are posted. Based on this model which contains no explicit link information between users, a uni-party data community generation algorithm called ISGI is proposed, which employs a specifically designed hierarchical similarity function to measure the correlations between two different individual users. Then, the BPUC algorithm is proposed, which uses the generated communities to predict users' behavior actions under certain conditions for situation awareness or personalized services development. For instance, the BPUC predictions may be used to answer questions such as "what will be the likely behavior user X may take if he/she logs into the BBS tomorrow?". Experiments on a large scale, real-world BBS data set demonstrate the effectiveness of the proposed model and algorithms.

1 Introduction

Bulletin board system (BBS) is an important information exchanging and sharing platform on the Internet. The analysis of useful patterns from BBS data has drawn much attention in recent years [5,6,8].

A BBS is an electronic "whiteboard" which usually consists of a number of *boards*, the discussion areas relating to some general themes (e.g. *Sports*). On each board, users read and/or post messages on different *topics*, which may be well determined by the titles of the message. In a BBS, one could easily start a discussion on a specific topic or express his/her viewpoint on an existing topic.

Since users with different backgrounds, different interests may access the same BBS, the BBS essentially serves as a mapping to the real world society, such that the relationships between the individual users may be discovered and analyzed through discovering and learning this mapping. Various relationships between users that hold sufficient interestingness to mine through the BBS data include the users with a similar interest or a similar taste, or a similar behavior action, and given what type of users, what specific behavior action may be taken if they share a similar specific interest. For example, two individuals who happen to be both basketball fans are likely to go to the same

boards under a topic related to basketballs of a BBS. Clearly, effective discovery of these relationships between users of a BBS through mining the BBS data is essential and extremely helpful in situation awareness and in the development and delivery of personalized services to users.

Community generation is an effective way to identify groups of data items satisfying certain relationship constraints in a large amount of data, where the identified groups are called *communities*. Based on the availability of link information between data items, methods could be divided into two categories [9]. One is *bi-party data community generation* (BDCG), where link information between data items is explicitly provided besides the features that describe the data items. Such link information is important and methods of this category usually generate communities by combining link analysis and clustering techniques (e.g., [11]). Successful applications include [4], [2], [3], etc. The other category, in contrast, is *uni-party data community generation* (UDCG), where the link information is *not* available and must be obtained by further exploring additional information from data items.

In this paper, the BBS data are mined to discover the interest-sharing user groups, or communities. In particular, the topics of the posted messages and the boards the messages are posted to are considered as the two attributes of a user's behavior actions to demonstrate the user's interest, and thus are subsequently considered as the two views to the user's actions. Hence, a formulated BBS data model is proposed in this paper consisting of a collection of the BBS users, whose behaviors or access patterns are described by the history of actions reflected in the two views. Under this model, a UDCG algorithm called ISGI, i.e. Interest-Sharing Group Identification, is proposed to discover the groups of the users with similar interests, where communities are generated by analyzing the correlations between users based on a specially designed hierarchical similarity function. In addition, the users' behaviors are predicted with the help of the interest-sharing groups under certain conditions, which illustrates one of many potential applications using the generated community. Experiments show that the interest-sharing user groups may be effectively discovered by ISGI, and the generated communities are helpful in predicting users' behaviors, which will be useful in situation awareness and personalized services development.

The rest of the paper is organized as follows. Section 2 formulates the BBS data model. Section 3 proposes the ISGI method. Section 4 describes how to use the generated community to predict the behavior of a given user. Section 5 reports on the experiment results. Finally, Section 6 concludes the paper.

2 A General Model for Community Generation on BBS

In general, a BBS provides more facilities (e.g., file sharing). To simplify the problem, we only consider the posted messages in a BBS in this paper. For further simplification, the message body is ignored and only the title of a message is used to fully determine the topics of the message. Key words of the titles are extracted using standard text processing techniques, and mapped to those collected topics through standard statistical analysis (histogramming).

To identify the specific interest-sharing relationships among a BBS users, we explicitly model a user's *access pattern* on BBS using information from two different views. Presumably, a BBS user tends to initiate or join in a discussion on a certain topic in which he or she is interested. Thus, the history of the topics on which the user has posted messages may reflect the interests of the user. Note that the users' interests are time-dependent because the discussions on BBS are usually closely related to the events that happen at the times when the discussions are raised. Consequently, posting messages to the same topic at different times may carry different semantics and meanings. On the other hand, a user's interest level in a specific topic may also be assessed by the *frequency* of messages which this user had posted on this topic within a certain period of time. For example, given a specific time interval, a user posting more messages on a topic presumably shows a greater interest in this topic than another user posting fewer messages on the same topic within the same time interval. Therefore, for the proposed BBS model, in the view of *Topics*, a user's access pattern is explicitly represented as a set of topics and the user access frequencies of the messages posted to different boards by different users along the timeline.

On the other hand, a user's interests may also be revealed by the boards where the messages are posted. In a typical BBS, discussion area is divided into different boards according to a set of categories. When accessing to a BBS, a user usually prefers visiting the boards that have the most interesting categories to this user. After exposing to an interesting topic in these boards, the user may decide to join the discussion on the topic being held in this board. Therefore, for the proposed BBS model, in the view of *Boards*, a user's access pattern is represented as a set of boards and the frequencies of messages posted to the boards along the timeline.

Consequently, the proposed BBS model is represented as a collection of users, each being represented by two timelines of *actions* on the Boards view and Topics view, respectively. Formally, let ID denote the set of all valid users in a BBS. Let \mathcal{T} and \mathcal{B} be the sets of the topics that have been discussed on the BBS and all the boards to which messages are posted, respectively; let T denote the set of time intervals quantified (e.g., a day) for the whole activation period of the BBS. Thus, the proposed BBS model is represented as follows:

$$BBS = \{ \langle id, A_{id}^T, A_{id}^B \rangle \mid id \in ID, A_{id}^T \subset \mathcal{A}^T, A_{id}^B \subset \mathcal{A}^B \} \quad (1)$$

$$\mathcal{A}^T = \{ \langle \tau, f_\tau, t \rangle \mid \tau \in \mathcal{T}, f_\tau \in \mathbb{N}, t \in T \} \quad (2)$$

$$\mathcal{A}^B = \{ \langle \beta, f_\beta, t \rangle \mid \beta \in \mathcal{B}, f_\beta \in \mathbb{N}, t \in T \} \quad (3)$$

where $\langle \tau, f_\tau, t \rangle$ and $\langle \beta, f_\beta, t \rangle$ are actions in each view, indicating that at time t posting messages with topic τ for f_τ times and to the board β for f_β times, respectively. Note that the timelines of both views are used together and contribute equally to the representation of the user's access pattern.

3 Interest-Sharing Group Identification

Given the BBS model presented above, we can identify the communities of users sharing similar interests. Unfortunately, many widely used methods (e.g., [3,4,7]) rely on

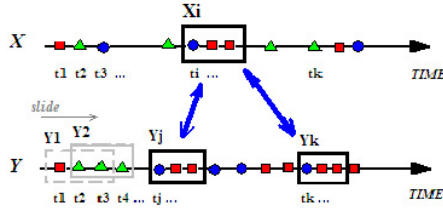


Fig. 1. An example of finding similar access patterns between the timelines of users

explicit link information to generate communities. Due to the absence of link information in our problem, we propose ISGI algorithm to identify interest-sharing groups from BBS without provided link information.

Firstly, the links between all the pairs of users are hypothesized, which induces a complete graph G_h on ID . And then, the correlation between each pair of users is measured by aggregating the overall similarities in each view of actions of the two users. we hierarchically define a similarity function to determine the correlation between two users access patterns under a given view. Such similarity is measured by combining a set of time-dependent local similarities between all pairs of access patterns in individual time slots along the timeline.

Specifically, given two timelines of actions X and Y (either in the Topic View or in the Boards View) of two users id_x and id_y , respectively, we examine similarity between every pair of time slots from different timelines by sliding a time window of size z along both the timelines, as shown in Figure 1. Let X_i and Y_j be sets of the actions in two time slots starting at time t and time s along each timelines, respectively. Note that the order information of actions within a time slot is not considered because users with similar interest may not necessarily take similar actions within a time slot in the same order. A straightforward way to define the similarity between X_i and Y_j is $|X_i \cap Y_j| / |X_i \cup Y_j|$. However, this definition ignores the frequencies of the actions; with this definition, one who takes an action (e.g., posting a message to a board) 100 times would be considered the same as another who takes the action only once. To accommodate the contributions from different action frequencies, the average frequency difference of the actions shared by both X_i and Y_j is defined as

$$fd(X_i, Y_j) = \frac{1}{|X_i \cap Y_j|} \sum_{a \in X_i \cap Y_j} |f_{X_i}(a) - f_{Y_j}(a)| \quad (4)$$

where $f_{X_i}(a)$ and $f_{Y_j}(a)$ denote the frequencies of the action a in X_i and Y_j , respectively. Then, we define *local similarity* between X_i and Y_j as

$$ls(X_i, Y_j) = \frac{1}{1 + fd(X_i, Y_j)} \cdot \frac{|X_i \cap Y_j|}{|X_i \cup Y_j|} \quad (5)$$

We then construct a global similarity between the two timelines based on the local similarities between all pairs time slots. Firstly, for any time slot X_i , we aggregate these local similarities between X_i and all $Y_j \in Y$ into a *hybrid similarity* between X_i and Y , which is defined as follows,

Table 1. Pseudo-code describing the ISGI algorithm**Algorithm:** ISGI**Input:** user set ID correlation threshold θ **Process:** Generate a complete graph $G_h(V_h, E_h)$ based on all users in ID **for each** $id_x \in ID$ **do****for each** $id_y \in ID$ **do** Compute the global similarity of id_x and id_y from the Boards view (c.f. Eq. 8) Compute the global similarity of id_x and id_y from the Topics view (c.f. Eq. 8) Generate the correlation value c on the edge (id_x, id_y) of G_h **end****end**Add all the edges whose correlation values are no less than θ to a new Edge set E Construct a new Vertex set V with id_x, id_y such that $(id_x, id_y) \in E$ **Output:** interest-sharing group $G(V, E)$

$$hs(X_i, Y) = \max_{Y_j \in Y} \{w(X_i, Y_j) ls(X_i, Y_j)\} \quad (6)$$

where

$$w(X_i, Y_j) = \exp\left(-\frac{|i-j|}{M}\right) \quad (7)$$

and M is the number of possible time slot in timeline Y .

Note that the local similarities are weighted by Eq. 7 which incorporates regularization that similar actions taken by two users with similar interests should not be too far from each other. The reason has been explained in Section 2.

Then, by using the hybrid similarities with respect to different time slots, we derive the *global similarity* between X and Y as

$$gs(X, Y) = \frac{1}{2} \left(\frac{\sum_{X_i \in X, X_i \neq \emptyset} hs(X_i, Y)}{\sum_{X_i \in X, X_i \neq \emptyset} 1} + \frac{\sum_{Y_j \in Y, Y_j \neq \emptyset} hs(Y_j, X)}{\sum_{Y_j \in Y, Y_j \neq \emptyset} 1} \right) \quad (8)$$

Note that only the hybrid similarities for the non-empty time slots are aggregated in Eq. 8. The reason is that in real world two users with similar interests may differ from each other by the log-in frequency. For instance, user id_y may login BBS everyday, while user id_x may login only once a month but does exactly what id_y does. If we use the hybrid similarities for all the empty time slots, the global similarity between the two users id_x and id_y would be very low.

Since the global similarity in each view reveals the correlation of id_x and id_y in different perspective, the overall correlation between the two users is computed by simply averaging the global similarities in both views.

After correlations between all pairs of users are obtained, all the weak links whose corresponding correlation value is less than a prest threshold θ is removed from the hypothesized graph G_h , and the induced graph is regarded as the interest-sharing groups G , where the neighbors of a user id_i , i.e., those who are connected to id_i by the links, share similar interests to id_i . The pseudo code of ISGI algorithm is shown in Table 1.

4 Predicting User Behavior Using Generated Community

In many existing work, the generated communities are only used for identifying correlated entities. Besides such a simple application, we consider another potential application which exploits the communities generated by ISGI on BBS – predicting user behavior under certain conditions.

Given a user id_i , now the task is to predict what action id_i may take in the near future, i.e., in a time slot of size z which starts at time t . A possible solution to this problem is to learn the probabilistic model directly from the BBS data. Since the actions that have been taken by id_i in current time slot may be closely related to id_i 's future actions in the same time slot, the prediction may be made according to Eq. 9, where the posterior probability is estimated by consulting the access history of id_i .

$$P(a_x | A_i^{obsv}; id_i) = \frac{\# \text{ of } a_x \text{ in a time slot with } a' \in A_i^{obsv}}{\# \text{ the time slots contain } a' \in A_i^{obsv}} \quad (9)$$

where A_i^{obsv} is the set of actions taken by id_i in the current time slot.

In reality, however, such a method fails since A_i^{obsv} is often empty. In this case, the posterior probability cannot be computed directly. This situation is common in a BBS. For instance, in order to provide a discussion recommendation, the prediction is usually required to be made as soon as the user logs in to the BBS. Fortunately, with the interest-sharing groups identified by ISGI, this problem can be resolved as follows.

Recall that a community is generated based on the similar access patterns between users. If a user is likely to take an action at a time instant, other users with the similar behavior also tend to take the action at some other time instants. Thus, when the posterior probability of action a_x for user id_i is computed, given that A_i^{obsv} is empty, we consults the neighbors of id_i in the generated community for determine the possible future actions of id_i according to the following equation,

Table 2. Pseudo-code describing the BPUC algorithm

Algorithm: BPUC

Input: user to be predicted id_i
view of action to be predicted V
generated community G
time slot TS_t starting at time t

Process: Fill the neighbor set N_i with all the neighbors of id_i in G
for each action a_x on the view V **do**
 for each id_j in N_i **do**
 Record the correlation value c_{ij} between id_i and id_j from G
 Construct A_j^{obsv} of with all the actions taken by id_j in TS_t on the both views
 Estimate the posterior probability $P(a_x | A_j^{obsv}; id_j)$ according to Eq. 9
 end
 Approximate the posterior probability using Eq. 10
end

Output: predicted user behavior $a^* \leftarrow \arg \max_{a_x} P(a_x | A_i^{obsv}; id_i)$

$$P(a_x | A_i^{obsv}; id_i) = \frac{1}{Z} \sum_{id_j \in N_i; A_j^{obsv} \neq \emptyset} c_{ij} P(a_x | A_j^{obsv}; id_j) \quad (10)$$

where c_{ij} is the correlation value between id_i and id_j , and $Z = \sum_{id_j \in N_i; A_j^{obsv} \neq \emptyset} c_{ij}$.

Note that according to Eq. 10 the estimation is done by weighting the sum of posterior probabilities of the neighbors instead of filling A_i^{obsv} with the actions in A_j^{obsv} first and then computing the posterior probability $P(a_x | A_i^{obsv}; id_i)$ directly. The reason is that the correlations between users reflect the possibilities that two users may take similar actions at a time instant; hence, the posterior probabilities of the action a_x may be “smoothly” propagated from those similar users to id_i . By contrast, propagating the events to id_i assumes that id_i should have also taken the actions that id_i 's neighbors have already taken, which is clearly inconsistent with the information conveyed by this community.

Based on Eq. 10, an algorithm called BPUC (Behavior Prediction Using Community), whose pseudo code is shown in Table 2, is proposed to generate the probabilities for user behavior prediction. BPUC may be used to predict what actions a given user may take in the near future. This is extremely useful in situation awareness in which we can foresee any potential event that is likely to happen as well as the likelihood associated with this event. Besides, it is also helpful in the development and the delivery of the personalized services such as discussion recommendation to the BBS users.

5 Experiments

5.1 Data Set

The data used for the experiments are extracted from the BBS of Nanjing University¹. Currently, this system is one of the most popular university BBS in mainland China. The daily average number of online users is usually above 5000.

In the experiments, all the messages dated from January 1st, 2003 to December 1st, 2005 on 17 most popular and frequently accessed boards are collected. For each message, all the nouns, verbs and quantities appearing in the title are extracted as a bag of key words to represent a certain topic. Some different topics discussing the same issue are merged together manually for semantic consistency. After that, the topics that have been discussed by less than 5 messages and the users who have posted less than 50 messages are removed from the data set.

After the removal, the data set contains 4512 topics of 17 boards, and there are 1109 users under consideration. For each user, data are organized into two views, i.e. the Boards view and the Topics view. In each view, the sets of actions with their frequencies are ordered along the timeline. Due to the considerations on effectiveness and efficiency, the smallest time unit used in this experiment is *Day*. Thus, there are altogether 1066 time instants along the timeline, and actions taken within a day are regarded as simultaneous events.

¹ More information could be found by accessing this BBS at <http://bbs.nju.edu.cn>

5.2 Experiments on Community Generation

In order to evaluate whether ISGI correctly identifies the interest-sharing groups, the ground truth of the data set must be available. However, since this is a real-world BBS, it is not feasible to get all the ground-truth information as this involves the users' privacy. Fortunately, 42 volunteers have joined the experiment and told us their IDs and main interests. Based on this valuable information, an evaluation set ES of 42 users is obtained. According to the main interest of the 42 users, they were roughly divided into 3 groups: 18 users are interested in modern weapons; another 12 users are fond of programming skills; and the rest of the users are fans of various computer games.

With the availability of part of the ground truth, the performance of the ISGI algorithm is evaluated by the *neighborhood accuracy* and the *component accuracy*, respectively. The neighborhood accuracy describes how accurate the neighbors of a user in the generated community share similar interests to that of the user, while the component accuracy measures how well these generated groups represent certain interests that are common to the individuals of the groups. For instance, considering a generated community shown in Fig. 2, the number of all possible links is 21 ($= \frac{7 \cdot (7-1)}{2}$). 7 links between similar users which should be kept in the graph and 10 links between dissimilar users which should be removed are correctly identified from the 21 possible links. Thus, the neighborhood accuracy is $(7 + 10)/21 = 0.810$. Since 7 pairs of similar users are grouped into the same graph component and no pairs of dissimilar users are split into different group, the component accuracy is $(7 + 0)/21 = 0.333$.

In the experiment, the size of the time slot used in ISGI is fixed to 5. Note that many well-known community generation methods (e.g., [11]) are essentially BDCG methods directly working on explicitly provided link information. They are not suitable for our baselines. Here, we only compares ISGI with another recently developed UDCG algorithm CORAL [9], which does not rely on explicit link information either. Due to the large number of users and the long timelines in both views, CORAL fails to generate a community from the experimental data set within a reasonable time interval. In order to report a manageable evaluation comparison between ISGI and CORAL, the original data set is reduced by downsizing the action points along the timelines by a factor of 10 such that each timeline comprises 107 time instants, and all the comparison evaluations with CORAL are reported based on this reduced data set. For simplicity, the original data set and the reduced data set is denoted by BBS_{big} and BBS_{small} respectively.

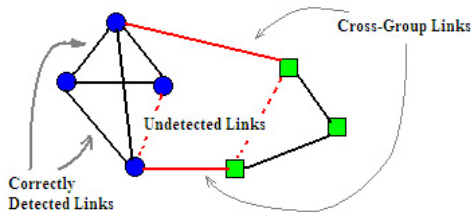


Fig. 2. An example of computing neighborhood accuracy and component accuracy

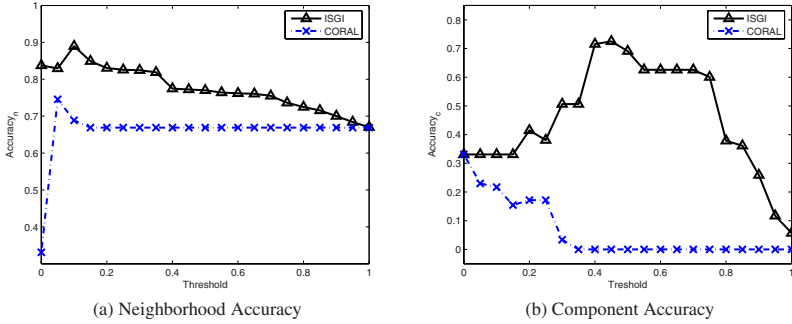


Fig. 3. Accuracies of the communities generated by ISGI and CORAL on *BBS_small*

Also, since CORAL only assumes one timeline for each individual user while in ISGI two timelines are used for the two views, respectively, another version of *BBS_small* is prepared for CORAL by collapsing the two timelines together into one to ensure a fair comparison between the two algorithms.

Recall that the structure of the community is determined by a pre-set minimum correlation threshold θ . In order to see how θ affects the community generation, in the experiments the value of θ varies from 0 to 1 with the step length 0.05. For each θ , the correlation values on all the links in communities generated by ISGI and CORAL respectively are normalized into the range $[0, 1]$, and then the accuracy of the communities on *ES* are measured respectively.

Fig. 3 reports the neighborhood accuracy and the component accuracy versus the threshold θ , respectively. It is clear to observe from the figures that the communities generated by ISGI are always better than those generated by CORAL for different θ w.r.t. both neighborhood and component accuracies.

Interestingly, when increasing θ from 0 to 0.05 to remove links from the initial community generated by CORAL, the neighborhood accuracy climbs up from 0.331 to the highest value 0.746, while the component accuracy drops at the same time. By

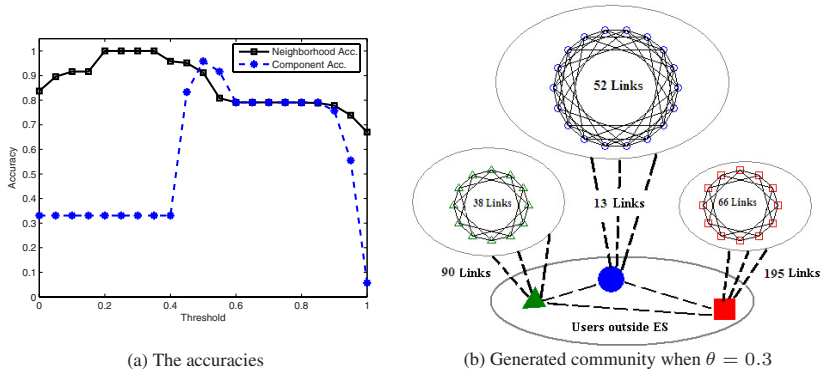


Fig. 4. Results of community generation using ISGI on *BBS_big*

Table 3. Time (in hours) taken for community generation

Data set	ISGI	CORAL
<i>BBS_small</i>	5.5	56.1
<i>BBS_big</i>	20.5	N/A

investigating the average number of the neighbors of a user and the number of the components when $\theta = 0$ and $\theta = 0.05$, it is found that the average number of a user's neighbors in a generated community drops dramatically from 953.1 to 64.1, and the number of the components in the community increases to 286. Therefore, it is concluded that most of the correlation values between similar users and between dissimilar users are both small such that it is difficult to discriminate links between similar users and those between dissimilar users by increasing θ . In CORAL, only the frequencies of actions can be used. Neither the information on the boards where the messages are posted nor the topics that the messages are addressed are used for deriving the correlations between users. Two users who post 10 messages to B_1 and B_2 respectively are regarded as similar by CORAL, while two users who post 5 messages and 20 messages to B_1 are regarded dissimilar. Therefore, all these facts suggest that CORAL is not suitable for identifying the interest-sharing user groups as ISGI does.

To further illustrate the effectiveness of ISGI on the original data set, ISGI is applied to *BBS_big* to generate communities with respect to different values of θ , and the accuracies of the generated communities are plotted in Fig. 4(a). Similarly, value of θ varies from 0 to 1 with the step length 0.05. As shown in the figure, ISGI performs even better on this large data set with respect to both the neighborhood accuracy and component accuracy. When θ ranges from 0.2 to 0.35, the neighborhood accuracy even reaches 1.0. Note that both accuracies of the communities generated by ISGI do not reach their corresponding maxima with the same value of θ . This phenomenon is due to the incomplete evaluation set *ES*. Even if the link between two dissimilar users is removed, the users may still be in the same group since they may still be connected to some other users outside the evaluation set. Moreover, Fig 4(b) gives an insight view of the generated community when $\theta = 0.3$. It is easy to find that the 3 groups of users with different interests are exactly identified by ISGI.

In addition, the evaluations are performed on workstations with 3.0 GHz Pentium 4 hyper-thread CPU. The running time ISGI and CORAL, respectively, on *BBS_small*, and the running time of ISGI on *BBS_big* is shown in Table 3. The CPU time shows that the extensibility of ISGI is better than that of CORAL in that ISGI is able to generate from large data set while CORAL fails.

5.3 Experiments on User Behavior Prediction

The community generated by ISGI in Section 6.2 is used to evaluate the BPUC algorithm described in Section 5. Here the task is to predict what actions a given user might take in the near future, i.e., within a time slot of the size z .

For each user in the experimental data set, the actions along the timeline in each view, either Boards or Topics, are split into two parts. One part which contains the

actions taken in the first 1056 days are used for training the probability model, while the actions in the last 10 days are kept aside for testing. In the experiment, the length of the time slot, within which the predicted actions may take place, is set to 5 days. Thus, there are altogether 6 different predictive time slots in the last 10 days. Predictions are made for each time slot and the errors are averaged over the 6 time slots. When predicting the most probable action that may be taken by a user within a time slot in the last 10 days, all the actions in the corresponding time slot of the user's neighbors are considered as the observed actions and are available for use.

Two algorithms, PM and Comm, are compared with BPUC. PM is a pure probabilistic model directly learned from the training data without using the generated community. Due to the characteristics of the task specified in Section 4, where a user has taken no actions in the predictive time slot observed, it is unable to compute the posterior probability according to Eq. 10. Instead, the prediction of the most probable action taken by the user is made based on the user's *prior probability* on the action to be predicted. Comm is a method that totally bases its prediction on the generated community. It considers the most frequent action taken by a user's neighbors in the community as the most probable action taken by the user, where the frequency of an action a_x is the correlation-weighted sum of the frequencies of a_x taken by the neighbors.

Leave-one-out test is used. In detail, when making prediction for a user with respect to a certain predictive time slot, the actions of the other users in the corresponding time slot are available for use. The users without neighbors in the community is skipped for prediction. Note that some neighbors of a user in the generated community may take no actions in the predictive time slots. In this case, both BPUC and Comm ignore these neighbors in making the prediction. If all the neighbors are ignored, the prediction for this user is also skipped.

Since a user may take several actions in a predictive time slot, the prediction is made correctly if the predicted most probable action appears in the given predictive time slot. Thus, the error rate with respect to a predictive time slot is computed by the ratio of the number of users whose predicted actions do not appear in the time slot over the total number of the users included in prediction. The evaluations are repeated for each of the 6 predictive time slots and the error rates are averaged to report the final error rate.

Different communities can be generated using different θ , thus, the experiment is repeated on each generated community. However, as θ increases, a user may have fewer neighbors in the community. To ensure that the neighborhood size is larger than 2, θ only ranges from 0 to 0.55 with a step length of 0.05.

For each community determined by θ , PM, Comm, and BPUC are used to predict the most probable boards a user might access. The error rates are tabulated in Table 4. It is obvious that BPUC and Comm outperforms PM. The average error rate of BPUC over different structures reaches 0.231, which improves 17.5% over PM on average. Moreover, even though Comm makes prediction only based on the generated community, it reaches lower error rates than PM. The average performance improvement of Comm over PM is 5.3%. Thus, the generated community is helpful to improve the prediction on the user behavior.

The average performance improvement of BPUC is higher than that of Comm. Although Comm achieves higher improvements for 6 different communities ($0.2 \leq \theta \leq$

Table 4. Error rates of compared algorithms based on the communities specified by θ

θ	0	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5	0.55	Avg.
PM	.307	.307	.307	.307	.307	.310	.310	.305	.277	.246	.199	.182	.280
Comm	.392	.390	.339	.261	.215	.213	.220	.233	.241	.230	.227	.214	.265
BPUC	.249	.249	.232	.225	.226	.236	.241	.260	.247	.242	.197	.174	.231

0.45), it also performs worse than BPUC for the other 6 communities. By contrast, BPUC performs stably well for different structures of the communities in the experiments. This fact indicates that BPUC benefits from the combination of probabilistic model and the generated community. BPUC is more suitable for this special task than Comm which bases its predictions only on the community.

6 Conclusions

Bulletin board system is an important platform for information exchange and sharing. This paper attempts to mine the interest-sharing groups from the BBS data and further applies the identified groups for user behavior prediction under certain condition. The contributions of this paper are as follows:

- We have formulated a general BBS data model for community generation as a collection of BBS users represented by two timelines of actions on different views. One view stands for the boards where the messages are posted, while the other represents the topics of the posted messages.
- We have designed a hierarchical similarity function to measure the relationship between different user IDs under the formulated model. This similarity function exploits time-dependent local similarities between timelines for each view and combines them for use.
- We have proposed a uni-party data community generation method called ISGI to identify the interest-sharing user groups under the formulated BBS data model. We have proposed the algorithm that combines a probabilistic model and the identified interest-sharing groups to predict the user behavior under certain conditions, which may be very useful for applications such as situation awareness and personalized services development.

Note that two users may post a message on the same topic to the same board with totally different actual contents. Consequently, besides the boards and the topics of the posted messages, the content of a message may also be used to describe a user's interest in the future work. Moreover, the user behavior prediction is just one application of the generated communities; identifying other applications using the generated communities will also be investigated in future.

Acknowledgement

Z.-H. Zhou and M. Li were partially supported by NSFC (60635030, 60721002) and 973 (2002CB312002), and Z. Zhang was supported in part by NSF (IIS-0535162), AFRL (FA8750-05-2-0284), and AFOSR (FA9550-06-1-0327).

References

1. Bhattacharya, I., Getoor, L.: Deduplication and group detection using links. In: KDD Workshop on Link Analysis and Group Detection (2004)
2. Cohen, W.W., Fan, W.: Web-collaborative filtering: recommending music by crawling the web. In: WWW 2000, pp. 685–698 (2000)
3. Culotta, A., Bekkerman, R., McCallum, A.: Extracting social networks and contact information from email and the web. In: CEAS 2004 (2004)
4. Gibson, D., Kleinberg, J., Raghavan, P.: Inferring web communities from link topology. In: Hypertext 1998, pp. 225–234 (1998)
5. Kou, Z., Zhang, C.: Reply networks on a bulletin board system. *Phys. Rev. E* 76 (2003)
6. Pena-Shaff, J.B., Nicholls, C.: Analyzing student interactions and meaning construction in computer bulletin board discussions. *Comp. & Edu.* 42, 243–265 (2004)
7. Toyoda, M., Kitsuregawa, M.: Creating a Web community chart for navigating related communities. In: Hypertext 2001, pp. 103–112 (2001)
8. Xu, J., Zhu, Y., Li, X.: An article language model for bbs search. In: Lowe, D.G., Gaedke, M. (eds.) ICWE 2005. LNCS, vol. 3579, pp. 152–160. Springer, Heidelberg (2005)
9. Zhang, Z., Salerno, J.J., Yu, P.S.: Applying data mining in investigating money laundering crimes. In: KDD 2003, pp. 747–752 (2003)

Extreme Support Vector Machine Classifier

Qiuge Liu^{1,2}, Qing He¹, and Zhongzhi Shi¹

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, P.O. Box 2704-28, Beijing 100080 China

² Graduate School of the Chinese Academy of Sciences, Beijing, China

{liuqg, heq, shizz}@ics.ict.ac.cn

Abstract. Instead of previous SVM algorithms that utilize a kernel to evaluate the dot products of data points in a feature space, here points are explicitly mapped into a feature space by a Single hidden Layer Feedforward Network (SLFN) with its input weights randomly generated. In theory this formulation, which can be interpreted as a special form of Regularization Network (RN), tends to provide better generalization performance than the algorithm for SLFNs—Extreme Learning Machine (ELM) and leads to a extremely simple and fast nonlinear SVM algorithm that requires only the inversion of a potentially small matrix with the order independent of the size of the training dataset. The experimental results show that the proposed Extreme SVM can produce better generalization performance than ELM almost all of the time and can run much faster than other nonlinear SVM algorithms with comparable accuracy.

1 Introduction

It has been shown that SLFNs with arbitrarily assigned input weights and with almost any nonzero activation function can universally approximate any continuous functions on any compact input sets ([1], [2]).

Based on these research results ELM [3] randomly chooses the input weights of an SLFN, then the output weights (linking the hidden layer to the output layer) of an SLFN is analytically determined by the minimum norm least-squares solutions of a general system of linear equations [3]. The running speed of ELM can be thousand times faster than traditional iterative implementations of SLFNs like BP, however it still tends to be overfitting and can be seen within the Empirical Risk Minimization (ERM) principle [15][16][17].

In this paper, with the focus on 2-class classification problem, a new nonlinear Support Vector Machine (SVM) formulation is proposed, in which a nonlinear map function is explicitly constructed by a SLFN with its input weights randomly generated. As can be seen later it leads to a better generalization performance than ELM most of the time and provides a stronger capacity control capability.

The new SVM classifier, which can be interpreted as a special form of regularization networks [4][5][6], classifies points by assigning them to the closest of

two parallel "approximating" planes like the way in Proximal SVM (PSVM) [7], Multisurface PSVM [8], and Least Squares SVM (LSSVM) [9] etc.. In all of the other SVMs, a nonlinear kernel is utilized to obtain a nonlinear classifier and a linear system of equations of the order of the number of data points needs to be solved, which makes it intractable when the number of training points is several thousands. In our new formulation, however, the dot products of the data points are computed explicitly by first map them into a feature space through a random SLFN and then an extremely fast and simple nonlinear SVM algorithm can be devised, which requires only the solution of a potentially small (usually less than 200) system of linear equations with the order independent of the size of the input dataset. We will call it the Extreme Support Vector Machine (ESVM) in the context of this paper.

This work is organized as follows. In Sect.2 the basic architecture of SLFNs and the ELM classifier is reviewed. Then in Sect.3 the new ESVM classifier is proposed, and we will also compare it with some other theories. Finally many numerical test results based on real world benchmarking classification problems can be found in Sect.4, which show that ESVM can produce better generalization performance than ELM most of the time and can run much faster than other nonlinear SVM algorithms with comparable testing set correctness.

A word about our notations. All vectors will be column vectors unless transposed by a superscript $'$. The scalar product of two vectors x and y in n -dimensional space R^n will be denoted by $x'y$, and the 2-norm of a vector x is denoted by $\|x\| = \sqrt{x'x}$. For a matrix $A \in R^{m \times n}$, A_i is the i th row of A which is a row vector R^n , while A_j is the j th column of A . A column vector of ones of arbitrary dimension will be denoted by e . The identity matrix of arbitrary dimension will be denoted by I .

2 SLFNs and ELM

In this section we provide some preliminaries about the architecture of SLFN (similar as the notation in [1]), and review the SLFN's ELM algorithm.

2.1 Single Hidden Layer Feedforward Networks

Assume that a training set consisting of m pairs of input vectors $\{a_j, d_j\} \ 1 \leq j \leq m$ is given, where $a_j \in R^n$, and $d_j \in R^{\tilde{n} \times 1}$ is the desired output vector. The output of the single layer neural network is described by

$$O^1 = G(A^1 W^1) \tag{1}$$

where $A^1 := [a_1^1, \dots, a_m^1]' \in R^{m \times (n+1)}$ is the matrix of input vectors, a_i^1 is the vector of inputs plus the bias of one for the last term i.e. $a_i^1 = [a_i', 1]'$; $O^1 := [o_1^1, \dots, o_m^1]' \in R^{m \times \tilde{n}}$ is the matrix of the neurons' output, $o_i^1 \in R^{\tilde{n} \times 1}, 1 \leq i \leq m$ is the vector of the neurons' output vector corresponding to a_i ; and $W^1 := [w_1^1, \dots, w_{\tilde{n}}^1] \in R^{(n+1) \times \tilde{n}}$ is the matrix of weight vectors. The notation

$G(Z)$ represents a map which takes a matrix Z with elements z_{ij} and returns another matrix of the same size with elements $g(z_{ij})$, where g is the neuron's nonlinearity.

The multilayer neural network consists of many layers of parallel neurons connected in a feedforward manner. Using the quantity $\#k$ as the number of nodes in the k th layer, the output of the k th layer is described by

$$O^k = G\left(A^k W^k\right) \quad (2)$$

Here $A^k := [a_1^k, \dots, a_m^k]' \in R^{m \times (\#(k-1)+1)}$ is the matrix of input vectors; a_i^k is the vector of inputs equal to the outputs from the previous layer plus the bias of one for the last term; $O^k := [o_1^k, \dots, o_m^k]' \in R^{m \times \#k}$ is the matrix of outputs; o_i^k is the k th layer neurons' output vector for a_i ; $W^k := [w_1^k, \dots, w_{\#k}^k] \in R^{\#(k-1)+1 \times \#k}$ is the matrix of weight vectors.

Thus for an Single hidden Layer Feedforward Network (SLFN) the expression of the output of the first hidden layer is the same as (1). And for simplicity the output of the second hidden (output) layer is described by

$$O^2 = A^2 W^2. \quad (3)$$

where

$$A^2 = [O^1, e] = [G(A^1 W^1), e]. \quad (4)$$

In this paper A^2 is called the hidden layer output matrix, and W^1, W^2 is named the input weights and the output weights of an SLFN respectively.

It has been shown that for an arbitrary training set with m training patterns, a neural network with one hidden layer and with $m - 1$ hidden layer neurons can exactly implement the training set ([12],[13],[22]). It has been further indicated that SLFNs (with N hidden neurons) with arbitrarily chosen input weights can learn N distinct observations with arbitrarily small error, which means that the input weights w^1 are not necessarily adjusted in applications ([2],[3]).

2.2 Extreme Learning Machine

Consider the 2-class classification problem of classifying m points in n -dimensional real space R^n , represented by the $m \times n$ matrix A . A diagonal matrix D with $+1$ or -1 along its diagonal specifies the membership of class $A+$ or class $A-$ of each point A_i . Note that for the 2-class classification problem the number of the output layer neurons of the SLFN is one, i.e. $\#2 = 1$.

ELM, based on what is stated in the preliminaries, randomly generates the input weights W^1 and it models the SLFN as follows:

$$\min_{W^2} F\left(W^2\right) = \|A^2 W^2 - De\|^2 \quad (5)$$

where A^2 is defined the same as in (4).

The key point of ELM is that the input weights W^1 of an SLFN need not be adjusted at all and can be arbitrarily given. It can be easily seen from (5) that to train an SLFN, in ELM, is simply equivalent to finding a least squares solution of the linear system $A^2W^2 = De$, which can be analytically determined by the expression below:

$$\widehat{W}^2 = A^{2\dagger}De \tag{6}$$

where $A^{2\dagger}$ is the generalized inverse of the hidden layer output matrix(14).

The expression (5) aims to minimize the empirical risk of the approximating function $A^2W^2 = O^2$. And, since the solution takes the minimum norm among the least-squares solutions, ELM provides weak control of the capacity of the models. Consequently the ELM algorithm can be considered within the ERM theme ([15],[16],[17]) and tends to result an overfitting model especially when the number of hidden neurons is relatively large as is shown by the numerical results in Sect.4.

Observe that the leaning process of ELM for an SLFN can be interpreted as consisting of two steps. First the input vectors are mapped to the hidden layer output vectors through the 1st hidden layer of the SLFN, with its input weights randomly generated. Second a minimum norm least squares solution of the output weights W^2 is obtained through (6).

Based on these observations a new SVM classifier — ESVM is devised in Sect.3, which first maps the input data into a feature space explicitly by the hidden layer of a random SLFN, then a linear algorithm based on regularization least squares is performed in the feature space. In theory it is derived from the SRM theory (15,16,17), and is supposed to provide better generalization performance than ELM. Moreover The experimental results in Sect.4 show that it runs much faster than other SVM algorithms with comparable accuracy.

3 Extreme Support Vector Machine Classifier

In this section we will introduce our new SVM algorithm — Extreme Support Vector Machine (ESVM). And we will also compare it with some other learning methods in theory.

3.1 The Linear Extreme Support Machine Classifier

Consider again the 2-class classification problem stated in Sect.2.

The linear Extreme Support Vector Machine (ESVM) algorithm has the same form as the linear PSVM [7], however still we present it here for the convenience of the derivation of our nonlinear formulation. For the classification problem stated above, the ESVM with a linear kernel tries to find the proximal planes: $x'w - r = \pm 1$ where w, r are the orientation and the relative location to the origin respectively. And it can be formulated by the following quadratic program with a parameter ν :

$$\begin{aligned} \min_{(w,r,y) \in R^{n+1+m}} & \frac{\nu}{2} \|y\|^2 + \frac{1}{2} (w'w + r^2) \\ \text{s.t.} & \quad D(Aw - er) + y = e \end{aligned} \tag{7}$$

which replaces the inequality constraints in standard SVM by equality. The resulting separating plane acts like below:

$$x'w - r \begin{cases} > 0, & \text{then } x \in A+, \\ < 0, & \text{then } x \in A-, \\ = 0, & \text{then } x \in A+ \text{ or } x \in A-, \end{cases} \tag{8}$$

We now introduce our new nonlinear ESVM classifier by applying the linear formulation (7) in a feature space introduced by a mapping function.

3.2 The Nonlinear Extreme Support Vector Machine Classifier

To obtain the nonlinear ESVM formulation, we devise a special nonlinear transform function: $\Phi(x)$, which maps the input vectors into the vectors in a feature space. Then the linear expression (7) is performed in the feature space to get the nonlinear classifier. To be concrete the nonlinear ESVM is formulated to be the following quadric program problem with a parameter ν .

$$\begin{aligned} \min_{(w,r,y) \in R^{\tilde{n}+1+m}} & \frac{\nu}{2} \|y\|^2 + \frac{1}{2} \left\| \begin{bmatrix} w \\ r \end{bmatrix} \right\|^2 \\ \text{s.t.} & \quad D(\Phi(A)w - er) + y = e \end{aligned} \tag{9}$$

where $\Phi(x) : R^n \rightarrow R^{\tilde{n}}$ is a map function which will be explained later. The lagrangian for (9) can be written as follow:

$$L(w, r, y, s) = \frac{\nu}{2} \|y\|^2 + \frac{1}{2} \left\| \begin{bmatrix} w \\ r \end{bmatrix} \right\|^2 - s'(D(\Phi(A)w - er) + y - e) \tag{10}$$

Here $s \in R^m$ is the lagrangian multiplier with the equality constraints of (9). Setting the gradients of this lagrangian with respect to (w, r, y, s) equal to zero gives the following KKT optimality condition:

$$\begin{aligned} w &= \Phi(A)'Ds \\ r &= -e'Ds \\ \nu y - s &= 0 \\ D(\Phi(A)w - er) + y - e &= 0 \end{aligned} \tag{11}$$

Substituting the first three expressions of (11) in the last expression gives an explicit expression for Ds in terms of the problem data A and D as follows:

$$Ds = \left(\frac{1}{\nu} I + \Phi(A)\Phi(A)' + ee' \right)^{-1} De = \left(\frac{1}{\nu} I + E_\Phi E_\Phi' \right)^{-1} De \tag{12}$$

where $E_\Phi = [\Phi(A) \quad -e] \in R^{m \times (\tilde{n}+1)}$.

To the best of our knowledge almost all previous nonlinear SVM algorithms make use of a kernel function $K(x', x)$ (e.g. RBF, Polynomial), which corresponds to the dot products of mapped vectors in the feature space, to implement the expression $\Phi(A)\Phi(A)'$ in (12). Thus the transform function Φ and many of its properties are unknown in these nonlinear SVM algorithms. However in ESVM we will construct the map function Φ explicitly by the hidden layer of a random SLFN as what is stated at the end of Sect.2. To be concrete the transform function can be formulated as follows:

$$\begin{aligned} \Phi(x) &= G(W^1 x^1) \\ &= \left(g\left(\sum_{j=1}^n W_{1j}^1 x_j + W_{1(n+1)}^1\right), \dots, g\left(\sum_{j=1}^n W_{\tilde{n}j}^1 x_j + W_{\tilde{n}(n+1)}^1\right) \right) \end{aligned} \tag{13}$$

where $x \in R^n$ is the input vector and $x^1 = [x', 1]'$, $W^1 \in R^{\tilde{n} \times (n+1)}$ is a matrix whose elements is randomly generated, and the notation $G(\cdot)$ has the same definition as in (11). Note that x^1, W^1 can be interpreted as the input vector and input weights of an SLFN respectively, and $\Phi(x)$ is the hidden layer's output vector of x .

It can be seen that the expression (12) of Ds still entails the inversion of a possibly massive matrix of order $m \times m$. To get rid of this problem we can make immediate use of the Sherman-Morrison-Woodbury (SMW) formula [21] for matrix inversion which results in the following expression:

$$Ds = \nu(I - E_\Phi \left(\frac{I}{\nu} + E_\Phi' E_\Phi\right)^{-1} E_\Phi') De \tag{14}$$

Note that if we substitute the expression (14) for Ds in (11), we can obtain the following simple expression for w and r in terms of problem data:

$$\begin{bmatrix} w \\ r \end{bmatrix} = \left(\frac{I}{\nu} + E_\Phi' E_\Phi\right)^{-1} E_\Phi' De \tag{15}$$

We comment further that the expression (15) only involves the inversion of a matrix of order $(\tilde{n} + 1) \times (\tilde{n} + 1)$, where \tilde{n} can be typically very small (usually less than 200 as is shown in Sect.4) and is independent of the number of the training points m .

Now for an unseen point x the nonlinear ESVM classifier works as follows:

$$\Phi(x)'w - r \begin{cases} > 0, & \text{then } x \in A+, \\ < 0, & \text{then } x \in A-, \\ = 0, & \text{then } x \in A+ \text{ or } x \in A-, \end{cases} \tag{16}$$

Compared to the linear classifier (8) we can see that (16) classify the point x in the feature space by maps it into $\Phi(x)$ first.

We can now give an explicit statement of our ESVM algorithm.

Algorithm 1. Extreme Support Vector Machine (ESVM) classifier

1. Input: m training points $\{x_i\}_{i=1}^m \in R^n$, target labels $\{y_i\}_{i=1}^m \in \{-1, +1\}$, $m \times n$ matrix $A = [x_1, \dots, x_m]$, bias vector $D = [d_1, \dots, d_m]$, $\nu > 0$, $\tilde{n} > 0$.

$$\begin{aligned}
 & W^1 \in R^{\tilde{n} \times (n+1)} \\
 & E_{\Phi} = [\Phi(A), -e] \quad e, \quad m \times 1 \\
 & \begin{bmatrix} w \\ r \end{bmatrix} \\
 & x
 \end{aligned}$$

In the next section we will present many experimental results which demonstrate the effectiveness of the ESVM algorithm.

3.3 What Is the Relationship between ESVM and the RN?

ESVM is a special form of Regularization Network. We can see from the expression (7) or (9) that the planes $x'w - r = \pm 1$ or $\Phi(x)'w - r = \pm 1$ are not bounding planes, like in standard SVM, anymore, but can be thought of as "proximal" planes, around which the points of each class are clustered. Thus the ESVM classifiers are constructed from an approximating function whose inputs are the training patterns and expected outputs are +1 or -1 according to the membership of input vectors in the class A+ or A- like PSVM [7] and LSSVM [9].

The problem of approximating a function from sparse data is ill-posed and a classical way to solve it is regularization theory [4, 18, 19], which formulates the approximating problem as a variational problem of finding the function f that minimizes the functional of the form

$$\min_{f \in \mathcal{H}} = \frac{1}{l} \sum_{i=1}^m V(D_{ii}, f(x_i)) + \lambda \|f\|_K^2, \tag{17}$$

where $V(\cdot, \cdot)$ is a loss function and $\|f\|_K^2$ is a norm in a Reproducing Kernel Hilbert Space \mathcal{H} defined by the positive definite function K and λ is the regularization parameter [20]. The ESVM formulation (9) can be seen as a special form of (17), in which the loss function is squares error and the positive definite kernel function K is defined by $K(x, y) = \Phi(x)' \cdot \Phi(y)$.

As depicted in [5] regularization network provides a form of capacity control and it, like SVM, can also be derived from Structural Risk Minimization (SRM) principle. Thus we can expect that ESVM can lead to a model that not only fits the training data but also with good predictive capability on new data according to Vapnik's theory [15, 16, 17].

3.4 What Is the Relationship between ESVM and Nonlinear PSVM?

As what is stated above, the linear ESVM (7) has the same formulation as the linear PSVM [7], however they have different nonlinear expression.

In [7] the proximal kernel-based Nonlinear PSVM (NPSVM) is formulated as follows:

$$\begin{aligned} \min_{(u,r,y) \in R^{n+1+m}} & \frac{\nu}{2} \|y\|^2 + \frac{1}{2}(u'u + r^2) \\ \text{s.t.} & D(K(A, A')Du - er) + y = e \end{aligned} \tag{18}$$

which, compared to (7), replace the primal variables w by its dual equivalent $w = A'Du$ and replace the linear kernel by a nonlinear kernel $K(A, A')$. Through the KKT optimality conditions of (18), we can get the explicit expression for Ds (s is the dual variables) in terms of the problem data A and D as follows:

$$Ds = \left(\frac{1}{\nu}I + KK' + ee'\right)^{-1}De \tag{19}$$

Compare (12) with (19), it can be easily seen that (12) do not require the kernel matrices' multiplication: KK' . Furthermore, as K is a square $m \times m$ matrix, the SMW formula is useless for (19) and the inversion must take place in a potentially high-dimensional R^m [7], which makes it intractable when the dataset is huge. However the resolution (14) of ESVM only requires the inversion of a matrix of order $\tilde{n} \times \tilde{n}$ where \tilde{n} is independent of m even when there are millions of data points. It is shown in the experimental results in Sect.4 that \tilde{n} can be much smaller than m with acceptable accuracy.

3.5 What Is the Relationship between ELM and ESVM?

As what is mentioned above both learning processes of ELM and ESVM can be think of consisting of two steps: first the input vector is mapped to a feature space by the hidden layer of a SLFN in ELM or by the function $\Phi(\cdot)$ in ESVM; second the algorithms are performed in the feature space. We can easily see that the transform function (13) in ESVM works in a similar way as the hidden layer of an SLFN in ELM. However the learning processes of the two algorithms are quite different.

As mentioned above, the solution of ESVM is a regularized least squares solution of $D(\Phi(A)'w - er) = e$, however the ELM obtains the minimum norm least square solution of (5) where we have the following relationship $A^2 = [\Phi(A), e]$ and $W^2 = [w', -r']'$ between ELM and ESVM.

As what is stated above the algorithm ELM tries to minimize the empirical risk of an SLFN on the training dataset and provides weak capacity control, which means, according to Vapnik's theory, it may leads to an overfitting model. However ESVM avoids this problem by regularization technique and the experimental results in Sect.4 show that it can lead to better generalization performance than ELM most of the time.

3.6 What Are the Differences between ESVM and Standard SVM?

Both ESVM and SVM [17] can be derived from Vapnik's SRM theory, however there are two main differences between ESVM and standard SVM.

First unlike standard SVM, ESVM is based on regularized least squares and can lead to an extremely fast and simple algorithm for generating a nonlinear classifier that merely requires the solution of a single system of linear equations (14). Second, instead of making use of an integral operator kernels $K(x, y)$ as in standard SVM, we construct a map function $\Phi : R^n \rightarrow R^{\tilde{n}}$ explicitly, which makes the resolution of ESVM only requires the inversion of a matrix of order $\tilde{n} \times \tilde{n}$ where \tilde{n} can be much smaller than the number of input vectors.

4 Experimental Results

In this section, the performance of the proposed ESVM learning algorithm is compared with the popular SVM algorithm, the NPSVM algorithm and the ELM algorithm on some benchmarking problems in the classification areas. Most of our computations for ESVM and ELM were performed in the environment of MATLAB 7.0 running in a machine with 2.80GHz Pentium 4 CPU and 512M memory. The C-coded SVM packages: LIBSVM is used in our simulations for SVM algorithm in the same PC. The kernel function used in SVM is radial basis function whereas the activation function used in ESVM and ELM is a simple sigmoidal function $g(x) = 1/(1 + exp(-x))$. To compare our ESVM and ELM, the dimensional \tilde{n} of the feature space in the ESVM are set to be the number of hidden neurons of the SLFN in the ELM.

The datasets used for our numerical tests are eight publicly available datasets from the UCI [24], Statlog and Delve repositories: australian, breast-cancer, diabetes, heart, ionosphere, liver-disorders, sonar, splice.

We conclude our computational results now in two groups as follows:

1. **Figure 1: Comparison of Generalization Performance between ESVM and ELM on Eight Different Publicly Available Datasets.** In this experiment we compared the generalization performance between ESVM and ELM on eight publicly available datasets. The testing accuracy of both ESVM and ELM are obtained by a ten-fold testing (10 percent of the total data points are randomly chosen as the testing datasets) and the parameter ν of ESVM is decided by cross validation. As shown in Fig.1, the generalization performance of ESVM are better than ELM most of the time especially when the number of hidden neurons is relatively large. We can observe that the testing accuracy of ELM first rises, and after arriving at the peak then falls as the number of hidden neurons increases, however the performance of ESVM is more stable.
2. **Table 1: Comparison between ESVM, Standard SVM and Nonlinear PSVM.** In this experiment we performed the ESVM, LIBSVM and Nonlinear PSVM (NPSVM) algorithms on 8 publicly available datasets. Here cross validation method is used to choose the parameter of SVM and NPSVM. Then the ten fold average training and testing accuracy and training time of these three algorithms are given. Furthermore for ESVM we also give the results with different value of \tilde{n} . The best results for different data sets is

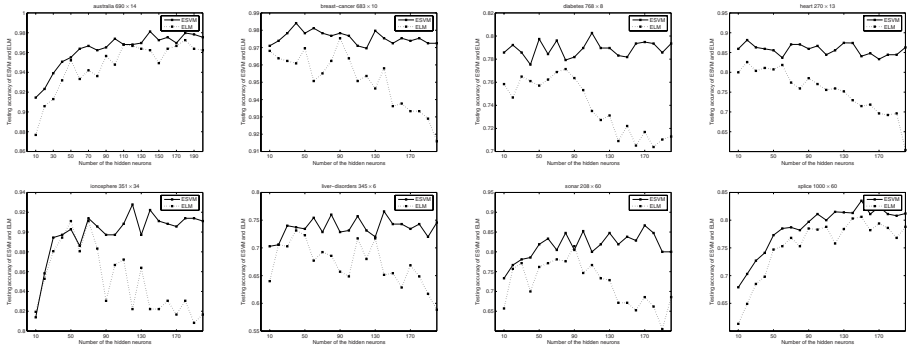


Fig. 1. Training and testing accuracy of ESVM and ELM on 8 publicly available datasets when different dimensional feature space is used

Table 1. Training and testing accuracy and training time of ESVM, SVM and NPSVM on eight publicly available datasets. The best testing accuracy is in bold face.

Datasets	ESVM						SVM Train Test Time	NPSVM Train Test Time
	20	60	100	140	180	200		
Australia 690 × 14	91.26%	96.09%	97.86%	98.42%	99.19%	99.28%	92.59%	100%
	92.32%	96.38%	97.39%	98.12%	97.97%	97.54%	83.91%	96.52%
	0.0047	0.0141	0.0219	0.0469	0.0703	0.0828	0.1703	0.3297
breast-cancer 683 × 10	97.08%	97.54%	97.77%	97.74%	97.92%	97.85%	96.73%	97.48%
	97.39%	98.12%	97.68%	97.68%	97.54%	97.25%	96.63%	97.73%
	0	0.0125	0.0281	0.0453	0.0672	0.0781	0.125	0.3281
diabetes 768 × 8	78.17%	80.46%	79.15%	80.81%	79.83%	85.33%	77.47%	79.15%
	79.22%	78.44%	78.96%	80.81%	79.83%	85.33%	75.78%	77.48%
	0.0078	0.0172	0.0313	0.0516	0.0766	0.0906	0.1689	0.4406
heart 270 × 13	85.76%	85.35%	88.72%	89.26%	90.12%	84.73%	96.75%	83.29%
	88.15%	83.70%	86.67%	87.41%	84.44%	86.30%	75.56%	82.96%
	0.0047	0.063	0.0109	0.0219	0.0313	0.0344	0.0312	0.0297
ionosphere 351 × 34	85.62%	94.19%	96.67%	96.32%	94.19%	97.46%	100%	99.37%
	85.83%	88.61%	89.72%	92.22%	91.39%	91.11%	92.02%	94.87%
	0.0031	0.0094	0.0156	0.0281	0.0344	0.0437	0.0610	0.0626
liver 345 × 6	75.13%	75.35%	77.23%	78.16%	76.32%	74.97%	80.58%	76.75%
	70.57%	75.43%	73.14%	76.57%	74.29%	74.57%	72.49%	73.34%
	0.0016	0.0063	0.0156	0.0234	0.0359	0.0453	0.05	0.0581
sonar 208 × 60	81.18%	90.43%	90.91%	99.89%	99.57%	87.49%	100%	100%
	76.67%	83.33%	85.24%	81.90%	84.76%	80%	74.04%	89.47%
	0.0016	0.0031	0.0141	0.0172	0.0313	0.0281	0.0405	0.0156
splice 1000 × 60	68.31%	80.08%	83.99%	86.63%	88.44%	86.17%	100%	-
	70.30%	78.50%	81.10%	81.30%	81.10%	81.20%	56.9%	-
	0.0063	0.0234	0.0484	0.0703	0.1	0.1141	1.25	-

emphasized in boldface. We can observe that the ESVM can achieve comparable accuracy to SVM most of the time, however the training time is shorter than SVM and NPSVM obviously. Specially for the splice dataset, the NPSVM is unapplicable as it requires too much memory.

5 Conclusions

In this paper we have proposed a new nonlinear SVM algorithm — ESVM based on regularized least squares. Instead of utilizing a kernel to compute the dot product of mapped data in the feature space, we explicitly construct a nonlinear transform function $\Phi(x) : R^n \rightarrow R^{\tilde{n}}$, which acts like the first hidden layer of an SLFN with its input weights randomly generated. The resolution of it requires nothing more sophisticated than solving a simple system of linear equations, in contrast to the more costly solution of a quadratic program in standard SVM. Our computational results demonstrate that ESVM can lead to a better predictive capability than ELM most of the time and reduce the training time of standard SVM greatly while still hold comparable accuracy.

Acknowledgements

This work is supported by the National Science Foundation of China (No. 60435010, 60675010), the 863 Project (No.2006AA01Z128), National Basic Research Priorities Programme (No. 2007CB311004) and the Nature Science Foundation of Beijing (No. 4052025).

References

1. Sartori, M.A., Antsaklis, P.J.: A simple method to derive bounds on the size and to train multilayer neural networks. *IEEE Trans. Neural Networks* 2, 34–43 (1991)
2. Huang, G.B., Chen, L., Siew, C.K.: Universal approximation using incremental feedforward networks with arbitrary input weights. In: *ICIS* (2003)
3. Huang, G.B., Zhu, Q., Siew, C.K.: Extreme Learning Machine: A New learning Scheme of Feedforward Neural Networks. In: *IJCNN* (2004)
4. Tikhonov, A.N., Arsenin, V.Y.: *Solutions of Ill-posed Problems*. W.H. Winston, Washington (1997)
5. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. *Advances in Computational Mathematics* 13, 1–50 (2000)
6. Evgeniou, T., Pontil, M., Poggio, T.: Regularization networks and support vector machines. In: *Advances in Large Margin Classifiers*, pp. 171–203 (2000)
7. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. In: *KDD*, pp. 77–86 (2001)
8. Mangasarian, O.L., Wild, E.W.: Multisurface proximal support vector machine classification via generalized eigenvalues. *IEEE Transaction on Pattern Analysis and Machine Intelligence* 28(1), 69–74 (2006)
9. Suykens, J.A.K., Vandewalle, J.: Least Squares Support Vector Machine Classifiers. *Neural Processing Letters* 9(3), 293–300 (1999)

10. Huang, G.B., Babri, H.A.: Upper bounds on the number of hidden neurons in feedforward networks with arbitrary bounded nonlinear activation functions. *IEEE Transactions on Neural Networks* 9(1), 224–229 (1998)
11. Huang, G.B.: Learning capability and storage capacity of two-hidden-layer feed-forward networks. *IEEE Transactions on Neural Networks* 14(2), 274–281 (2003)
12. Baum, E.B.: On the capabilities of multilayer perceptions. *J. Complexity* 4, 193–215 (1988)
13. Huang, S.C., Huang, Y.F.: Bounds on number of hidden neurons in multilayer perceptrons. *IEEE Trans. Neural Networks* 2, 47–55 (1991)
14. Serre, D.: *Matrices: Theory and applications*. Springer, New York (2002)
15. Vapnik, V.N.: *Estimation of Dependences Based on Empirical Data*. Springer, Berlin (1982)
16. Vapnik, V.N.: *The Nature of Statistical Learning Theory*. Springer, New York (1995)
17. Vapnik, V.N.: *Statistical Learning Theory*. Wiley, New York (1998)
18. Bertero, M.: Regularization methods for linear inverse problems. In: *Inverse Problems*, Springer, Berlin (1986)
19. Bertero, M., Poggio, T., Torre, V.: Ill-posed problems in early vision. *Proc. IEEE*, 869–889 (1988)
20. Wahba, G.: *Splines Models for Observational Data*. Series in Applied Mathematics, vol. 59. SIAM, Philadelphia (1990)
21. Golub, G.H., Van Loan, C.F.: *Matrix Computations*, 3rd edn. The John Hopkins University Press, Baltimore (1996)
22. Nilsson, N.J.: *Learning Machine*. McGraw-Hill, New York (1965)
23. Chang, C.-C., Lin, C.-J.: LIBSVM: a library for support vector machines (2001)
24. Murphy, P.M., Aha, D.W.: UCI repository of machine learning databases (1992)

LCM over ZBDDs: Fast Generation of Very Large-Scale Frequent Itemsets Using a Compact Graph-Based Representation

Shin-ichi Minato¹, Takeaki Uno², and Hiroki Arimura¹

¹ Graduate School of Information Science and Technology,
Hokkaido University, Sapporo, 060-0814 Japan
{minato, arim}@ist.hokudai.ac.jp

² National Institute of Informatics, Tokyo 101-8430, Japan
uno@nii.ac.jp

Abstract. Frequent itemset mining is one of the fundamental techniques for data mining and knowledge discovery. In the last decade, a number of efficient algorithms have been presented for frequent itemset mining, but most of them focused on only enumerating the itemsets that satisfy the given conditions, and how to store and index the mining result in order to ensure an efficient data analysis is a different matter.

In this paper, we propose a fast algorithm for generating very large-scale all/closed/maximal frequent itemsets using Zero-suppressed BDDs (ZBDDs), a compact graph-based data structure. Our method, “LCM over ZBDDs,” is based on one of the most efficient state-of-the-art algorithms proposed thus far. Not only does it enumerate/list the itemsets, but it also generates a compact output data structure on the main memory. The result can be efficiently postprocessed by using algebraic ZBDD operations. The original LCM is known as an output linear time algorithm, but our new method requires a sub-linear time for the number of frequent patterns when the ZBDD-based data compression works well. Our method will greatly accelerate the data mining process and this will lead to a new style of on-memory processing for dealing with knowledge discovery problems.

1 Introduction

Considerable attention in the last decade has been placed on discovering useful information from large-scale databases. Frequent itemset mining is one of the fundamental data mining problems. Since the pioneering paper by Agrawal et al. [1], various algorithms have been proposed to solve the frequent pattern mining problem (cf., e.g., [3, 5, 16]). Among those state-of-the-art algorithms, Uno et al. [15, 13, 14] has a feature of the theoretical bound as output linear time. Their open source code [12] is known as one of the fastest implementations of a frequent itemset mining program.

LCM and most of the other itemset mining algorithms focus on only enumerating or listing the itemsets that satisfy the given conditions, and how to

store and index the result of itemsets for a more efficient data analysis was a different matter. If we want to postprocess the mining results by setting various conditions or restrictions, we have to dump the frequent itemsets into storage at least once. Even though LCM is an output linear time algorithm, it may require impracticable time and space if the number of frequent itemsets gets too large. We usually control the size of the output by using the minimum support threshold in the ad hoc setting, but we are unsure if this may cause some important information to be lost.

For representing very large-scale frequent itemsets, S. Minato proposed a method using [ZBDDs](#), an efficient graph-based data structure. ZBDD is a variant of a [BDD](#), which was originally developed in the VLSI logic design area, but has recently been applied to data mining problems [\[6,8\]](#). Last year, Minato et al. presented the [ZBDD-based](#) algorithm for computing all/closed/maximum frequent itemsets based on ZBDD operations, and that generates a compressed output data structure on the main memory. Unfortunately, the overhead of ZBDD-based frequency computation is not small when using their algorithm, so the computational advantage is limited to only the examples where the ZBDD-based data compression rate is extremely high. Otherwise, for example, when the number of frequent itemsets is not large, an ordinary LCM algorithm is much faster than the ZBDD-growth one.

In this paper, we propose a nice combination of an LCM algorithm and a ZBDD-based data structure. Our method, “LCM over ZBDDs,” can generate very large-scale frequent itemsets on the main memory that uses a very small overhead of computational time when compared with the original LCM algorithm. The mining result can be efficiently postprocessed by using algebraic ZBDD operations. The original LCM is an output linear time algorithm, but our new method requires a sub-linear time for the number of frequent itemsets when the ZBDD-based data compression works well. Our method will greatly accelerate the data mining process and this will lead to a new style of on-memory processing for dealing with knowledge discovery problems.

2 Preliminaries

Let $\mathcal{E} = \{1, 2, \dots, n\}$ be the set of items. A [transaction](#) on \mathcal{E} is a multiset $\mathcal{T} = \{T_1, T_2, \dots, T_m\}$ where each T_i is included in \mathcal{E} . Each T_i is called a [transaction](#) (or [itemset](#)). We denote the sum of sizes of all transactions in \mathcal{T} , with $|\mathcal{T}|$ that is, the size of database \mathcal{T} . A set $P \subseteq \mathcal{E}$ is called an [itemset](#) (or [subset](#)). The maximum element of P is called the [tail](#) of P , and is denoted by $tail(P)$. An itemset Q is a [tail extension](#) of P if and only if both $Q \setminus P = \{e\}$ and $e > tail(P)$ hold for an item e . An itemset $P \neq \emptyset$ is a tail extension of Q if and only if $Q = P \setminus tail(P)$, and therefore, Q is unique, i.e., any non-empty itemset is a tail extension of a unique itemset.

For itemset P , a transaction including P is an [occurrence](#) of P . The [occurrences](#) of P , which is denoted by $Occ(P)$, is the set of the occurrences of P . $|Occ(P)|$ is the [number](#) of P , and is denoted by $freq(P)$. In particular, for an item e ,

$freq(\{e\})$ is the frequency of e . For a given constant θ , called a threshold, itemset P is frequent if $freq(P) \geq \theta$. If a frequent itemset P is not included in any other frequent itemset, P is closed. We define the closure of itemset P in \mathcal{T} , denoted by $clo(P)$, with $\bigcap_{T \in Occ(P)} T$. An itemset P is closed if $P = clo(P)$.

3 LCM and ZBDDs

We briefly explain LCM algorithm and ZBDD-based techniques for representing frequent itemsets in this section.

3.1 LCM Algorithm

LCM is a series of algorithms for enumerating frequent itemsets, which was developed by Uno et al. These algorithms feature that the computation time is theoretically bounded as an output linear time. The first LCM algorithm was presented at FIMI2003 [15], and the second version of LCM demonstrated its remarkable efficiency at FIMI2004 [13]. The original LCM was developed for enumerating closed itemsets, and then LCMfreq and LCMmax were presented for mining all frequent itemsets and maximal itemsets [1]. Now the three variants are integrated into one program. These implementations are available on the developer’s web page [12] as open source software.

In general, frequent itemset mining algorithms are classified into two categories: forward (or breadth-first) algorithms [1] and backward (or depth-first) algorithms [16, 5]. LCM algorithms belong to the backtracking style.

Backtracking algorithms are based on recursive calls. The algorithm inputs a frequent itemset P , and generates new itemsets by adding one of the unused items to P . Then, for each itemset being frequent among them, it generates recursive calls with respect to it. To avoid duplications, an iteration of the backtracking algorithms adds items with indices larger than the tail of P . The following information is a description of the framework of the backtracking algorithms.

ALGORITHM Backtracking (P : itemset)
Output P
For each $e \in \mathcal{E}$, $e > tail(P)$ **do**
If $P \cup \{e\}$ **is frequent then**
call Backtracking ($P \cup \{e\}$)

LCM algorithms are based on backtracking algorithms, and use acceleration techniques for the frequency counting, called *fast counting* and *fast counting with pruning*. Therefore, LCM algorithms efficiently compute the frequency. Here, we omit the detailed techniques used in LCM, as they are described in references [13, 14].

¹ The complexity has been theoretically proven in generating all/closed itemsets, but is still open (only experimental) for a maximal one.

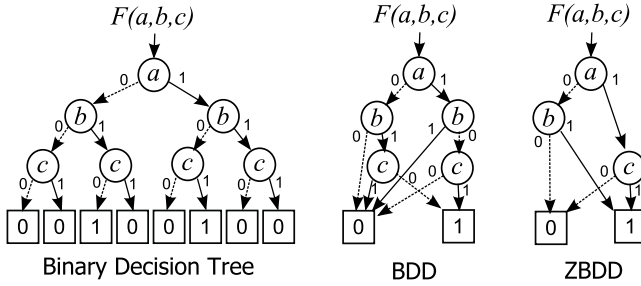


Fig. 1. Binary Decision Tree, BDDs and ZBDDs

a	b	c	F
0	0	0	0
0	0	1	0
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	0

$\rightarrow S$
 As a Boolean function:
 $F = a\bar{b}c \vee \bar{a}b\bar{c}$
 $\rightarrow b$
 As a set of combinations:
 $S = \{ac, b\}$
 $\rightarrow ac$

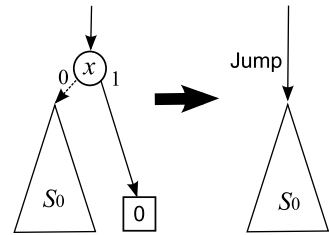


Fig. 3. ZBDD reduction rule

Fig. 2. Correspondence of Boolean functions and sets of combinations

Although LCM can efficiently enumerate large-scale frequent itemsets, how to store and index the result of itemsets for efficient data analysis is a different matter. Even though LCM is an output linear time algorithm, it may require impracticable time and space if the number of frequent itemsets becomes too large. We usually control the output size by using the minimum support threshold in the ad hoc setting, but we do not know if it may lose some of the important information that needed to be discovered.

3.2 ZBDDs

A ZBDD is a graph representation for a Boolean function. An example is shown in Fig. 1 for $F(a,b,c) = a\bar{b}c \vee \bar{a}b\bar{c}$. Given a variable ordering (a, b, c in our example), we can use Bryant's algorithm [2] to construct the BDD for any given Boolean function. For many Boolean functions appearing in practice this algorithm is quite efficient and the resulting BDDs are much more efficient representations than binary decision trees.

BDDs were originally invented to represent Boolean functions. However, we can also map a set of combinations into the Boolean space of n variables, where n is the cardinality of \mathcal{E} (Fig. 2). So, we could also use BDDs to represent sets of combinations. However, we can even obtain a more efficient representation by using zero-suppressed BDDs (ZBDDs) [7].

Table 1. Primitive ZBDD operations

" \emptyset "	Returns empty set. (0-terminal node)
" 1 "	Returns set of only null-combination. (1-terminal node)
$P.top$	Returns item-ID at root node of P .
$P.offset(v)$	Subset of combinations not including item v .
$P.onset(v)$	Gets $P - P.offset(v)$ and then deletes v from each combination.
$P.change(v)$	Inverts existence of v (add / delete) on each combination.
$P \cup Q$	Returns union set.
$P \cap Q$	Returns intersection set.
$P - Q$	Returns difference set. (in P but not in Q.)
$P.count$	Counts number of combinations.

If there are many similar combinations then the subgraphs are shared resulting in a smaller representation. In addition, ZBDDs have a special type of node deletion rule. As shown in Fig. 3, all of the nodes whose 1-edge directly points to the 0-terminal node are deleted. As the result, the nodes of items that do not appear in any sets of combinations are automatically deleted (Fig 4). This ZBDD reduction rule is extremely effective for handling a set of sparse combinations. If the average appearance ratio of each item is 1%, ZBDDs are possibly more compact than ordinary BDDs, even up to 100 times more.

ZBDD representation has another good property, which is that each path from the root node to the 1-terminal node corresponds to each combination in the set. Namely, the number of such paths in the ZBDD equals the number of combinations in the set. This attractive property indicates that, even if there are no equivalent nodes to be shared, the ZBDD structure explicitly stores all the items of each combination, as well as uses an explicit linear linked list data structure. In other words, (the order of) the size of the ZBDD never exceeds the explicit representation. If more nodes are shared, the ZBDD is more compact than the linear list.

Table 1 summarizes most of the primitive operations of the ZBDDs. In these operations, " \emptyset ," "**1**," and $P.top$ can be obtained in a constant time. $P.offset(v)$, $P.onset(v)$, and $P.change(v)$ operations require a constant time if v is the top variable of P , otherwise they consume linear time for the number of ZBDD nodes located at a higher position than v . The union, intersection, and difference operations can be performed in almost linear time to the size of the ZBDDs.

3.3 ZBDD-Growth Algorithm

Using a ZBDD-based compact data structure, we can efficiently manipulate large-scale itemset databases on the main memory. Recently, Minato et al. have developed a "ZBDD-growth" algorithm to generate all/closed/maximal frequent itemsets for given databases. The details of the algorithm are written in the article referenced in this paper [11]. The ZBDD-growth is based on the backtracking algorithm using recursive calls as well as the LCM. This algorithm has two following technical features:

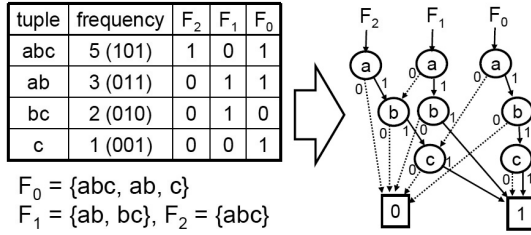


Fig. 4. ZBDD vector for frequency counting

- (i) Uses ZBDDs for the internal data structure, and
- (ii) Uses ZBDDs for the output data structure.

In the first feature, the internal data structure means that the given transaction database is converted to a ZBDD-based representation on the main memory. On each recursive step of the backtracking, frequency counting for the conditional (or restricted) database is performed by the ZBDD operations. This is similar to the algorithm [5], which manipulates the ZBDDs in the backtracking algorithm.

Since ZBDDs are representations of sets of combinations, a simple ZBDD only distinguishes the existence of each itemset in the database. In order to count the integer numbers of frequency, the ZBDD-growth algorithm uses the m -digits of the ZBDD vector $\{F_0, F_1, \dots, F_{m-1}\}$ to represent the integers up to $(2^m - 1)$, as shown in Fig. 4. The numbers are encoded into a binary digital code, as F_0 represents a set of itemsets appearing at odd times (LSB = 1), F_1 represents a set of itemsets whose appearance number's second lowest bit is a one, and which is similar to the way we define the set of each digit up to F_{m-1} . Notice that this ZBDD vector is used only for the internal data structure in the ZBDD-growth algorithm. The output data is represented by a simple ZBDD, because the result is just a set of frequent itemsets. (It does not keep the frequency of each itemset.)

ZBDD-growth algorithm manipulates the ZBDDs for both the internal and output data structures, so the advantage of the ZBDD-based data compression is fully employed. There are examples where billions of frequent itemsets can be represented by only thousands of ZBDD nodes. The mining result can be efficiently postprocessed by using algebraic ZBDD operations.

However, ZBDD-growth has a frequency computing overhead for using ZBDD vectors. The arithmetic operations of the ZBDD vectors are performed by a series of ZBDD operations on each binary digit, and this requires more steps than ordinary 32- or 64-bit arithmetic operations in the CPU normally use. Unless the ZBDD-based data compression rate is very high, the overhead becomes obvious. There are the two typical cases when the ZBDD is not very effective.

- The number of itemsets is small enough to be easily handled in anyway.
- The database is completely random and no similar itemsets are included.

In many practical cases, the ZBDD-growth algorithm is no faster than previous algorithms. As shown in the experimental results outlined in this paper, the

ZBDD-growth is 10 to 100 times slower than ordinary LCM when the output size is small. ZBDD-growth wins only when a huge number of frequent itemsets are generated.

4 LCM over ZBDDs

In this section, we discuss the combination of the LCM and ZBDDs. It is fortunate that we can observe a number of common properties in LCM algorithms and ZBDD manipulation, and they are listed as follows:

- Both are based on the backtracking (depth-first) algorithm.
- All the items used in the database have a fixed variable ordering.
- In the algorithm, we choose items one by one according to the variable ordering, and then recursively call the algorithm.
- In the current LCM implementation, the variable ordering is decided at the beginning of the algorithm, and the ordering is never changed until the end of execution.

These common properties indicate that LCM and ZBDDs may be a really good combination. Our algorithm, “LCM over ZBDDs,” does not touch the core algorithm of LCM, and just generates a ZBDD for the solutions obtained by LCM. In this way, we aim to efficiently generate very large-scale frequent itemsets with a very small overhead of ZBDD manipulation. We will now describe the techniques used in the new method.

4.1 ZBDD Construction in LCM Procedure

We recall the basic structure of the original LCM algorithm shown in Fig. 5. However, we omit the detailed techniques used in checking the frequency of each itemset, but basically the algorithm explores all the candidates of the itemsets in a backtracking (or depth-first) manner, and when a frequent itemset is found, they are appended one by one to the output file. On the other hand, “LCM over ZBDDs” constructs a ZBDD that is the union of all the itemsets found in the backtracking search, and finally returns a pointer to the root node of the ZBDD. A naive modification can be described using in Fig. 6. However, this naive algorithm has a problem with its efficiency.

In the LCM procedure, a ZBDD grows by repeating the union operations of the frequent itemsets found in the depth-first search. If we look at the sequence of itemsets generated by the algorithm, the consecutive itemsets are quite similar to each other in most cases, namely, only a few items near the tail are different and the other top items are completely identical. The ZBDD union operations look similar to those shown in Fig. 7 although only a few of the bottom levels are different, but almost all the other parts are the same. Since the procedures for the ZBDD operations are recursively executed from the top node to the bottom one, the computation of a union operation requires $O(n)$ steps, while only a few bottom items are meaningful. Namely, this algorithm will become n times

```

LCM_Backtrack(P: itemset)
{
  Output P
  For e = n to tail(P) + 1 step -1
  do
    If P ∪ {e} is frequent
      LCM_Backtrack(P ∪ {e})
}
    
```

Fig. 5. Basic structure of LCM algorithm

```

ZBDD LCMovZBDD_Naive(P: itemset)
{
  ZBDD F ← P
  For e = n to tail(P) + 1 step -1 do
    If P ∪ {e} is frequent {
      F' ← LCMovZBDD_Naive(P ∪ {e})
      F ← F ∪ F'
    }
  Return F
}
    
```

Fig. 6. Naive modification for generating ZBDDs

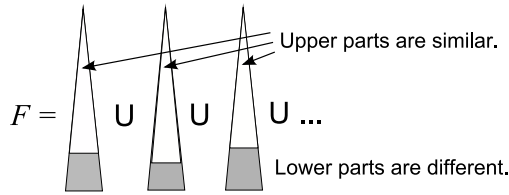


Fig. 7. ZBDD union operations in naive LCM over ZBDDs

slower. This is an unacceptable loss of efficiency, because n may be more than a hundred in practical datasets.

To address this problem, we improved the algorithm (Fig. 8). On each recursive step, we construct a ZBDD only for the lower items, and after returning from the subsidiary recursive call, we stack the top item up on the current result of ZBDD. In this way, we can avoid redundant traversals in the ZBDD union operation, as shown in Fig. 9. If we use the variable ordering of ZBDDs that is the same as the LCM’s item ordering, each ZBDD operation requires only a constant time, and the total overhead of the ZBDD generation can be bounded by a constant factor compared with the original LCM.

```

ZBDD LCMovZBDD(P: itemset)
{
  ZBDD F ← “1”
  For e = n to tail(P) + 1 step -1 do
    If P ∪ {e} is frequent {
      F' ← LCMovZBDD(P ∪ {e})
      F ← F ∪ F'.change(e)
    }
  Return F
}
    
```

Fig. 8. Improved version of “LCM over ZBDD”

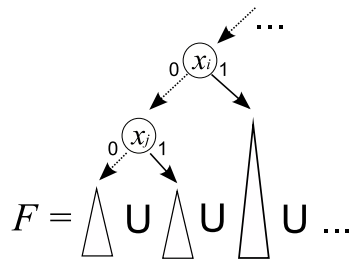


Fig. 9. Efficient ZBDD construction in LCM over ZBDDs

4.2 Employing Hypercube Decomposition

The original LCM finds a number of frequent itemsets all at once to reduce the computation time by using the technique of [15] (or, also called [16]). For a frequent itemset P , let $H(P)$ be the set of items e satisfying $e > tail(P)$ and $Occ(P) = Occ(P \cup \{e\})$. Then, for any $Q \subseteq H(P)$, $Occ(P) = Occ(P \cup Q)$ holds, and $P \cup Q$ is frequent. The original LCM avoids duplicated backtracking with respect to the items included in $H(P)$, by passing $H(P)$ to the subsidiary recursive calls. This algorithm is shown in Fig. 10.

Current LCM implementations have two output options: (i) printing out all the solutions to the output file, or (ii) just counting the total number of solutions. When counting the number of itemsets, we accumulate a 2's power to the hypercube size for each solution, without generating all the candidates derived from the hypercube. This technique greatly reduces the computation time because the LCM algorithm is dominated by the output size.

Also in LCM over ZBDDs, we can employ the hypercube decomposition technique. The algorithm is described in Fig. 11. A remarkable advantage of our method is that we can efficiently generate a ZBDD that includes all of the solutions, within a similar computation time as the original LCM when counting only the number of solutions. The original LCM is known as an output linear time algorithm, but our method can generate all the solutions in a sub-linear time for the number of solutions if the hypercubes appear often.

4.3 Closed/Maximal Itemset Mining

The original LCM can also generate closed/maximal itemsets. Our method does not touch the core algorithm of LCM, and just generates ZBDDs for the solutions obtained by LCM. Therefore, a ZBDD for closed/maximal itemsets as well as the original LCMs can be generated. The technique for hypercube decomposition should be slightly modified to generate a closed/maximal one, but it is a similar technique as to one used in the original LCMs.

```

LCM_Backtrack_H(P, S: itemset)
{
    S' ← S ∪ H(P)
    Output itemsets including P
        and included in P ∪ S'
    For e = n to tail(P) + 1 step -1
    do
        If e ∉ S' and P ∪ {e} is frequent
            LCM_Backtrack_H(P ∪ {e}, S')
}
    
```

Fig. 10. Original LCM with hypercube decomposition

```

ZBDD_LCMovZBDD_H(P, S: itemset)
{
    S' ← S ∪ H(P)
    ZBDD F ← "1"
    For e = n to tail(P) + 1 step -1 do
        If e ∈ S'
            F ← F ∪ F.change(e)
        Else if P ∪ {e} is frequent {
            F' ← LCMovZBDD_H(P ∪ {e}, S')
            F ← F ∪ F'.change(e)
        }
    Return F
}
    
```

Fig. 11. LCM over ZBDDs with hypercube decomposition

5 Experimental Results

Based on the above ideas, we implemented LCM over ZBDDs by modifying the open software, LCM ver. 5 [12]. We composed about 50 line modifications or additions to the main file of the original LCM, and compiled it with our own ZBDD package, which consists of about 2,300 lines of C codes. We used a 2.4GHz Core2Duo E6600 PC, 2 GB of main memory, with SuSE Linux 10 and a GNU C++ compiler. On this platform, we can manipulate up to 40,000,000 nodes of ZBDDs with up to 65,000 different items.

To evaluate the performance of our method, we applied it to a practical size of the datasets chosen from FIMI2003 repository [4] with various minimum support thresholds. We compared our results with those of the original LCM [12] and the ZBDD-growth [11]. In the datasets, a “mushroom” is known as an example where the ZBDD-growth is effective because the ZBDD-based data compression works well. “T10I4D100K” is known as the opposite, an artificial database consists of randomly generated combinations. In this case, ZBDD-based data compression is quite ineffective. “BMS-WebView-1” has an intermediate property between the two.

Table 2 shows our experimental results. In this table, $|ZBDD|$ represents the number of ZBDD nodes representing all the frequent itemsets. The column “LCM-count” shows the computational time of the original LCM when counting only the number of solutions, and “LCM-dump” represents the time for listing all the itemset data to the output file (using /dev/null). “LCMoverZBDD” and “ZBDD-growth” show the time for generating the results of the ZBDD on the main memory, including the time for counting the ZBDD nodes.

From the experimental results, we can clearly see that LCM over ZBDDs is more efficient than ZBDD-growth in most cases. The advantage of our method can be observed when a smaller number of solutions are generated. ZBDD-growth shows comparable performances to our method only in the “mushroom” with very low minimum support, but for all the other cases, our method overwhelms the ZBDD-growth.

We can also observe that LCM over ZBDDs is more efficient than the original LCM-dump. The difference becomes significant when very large numbers of itemsets are generated. The original LCM-dump is known as an output linear time algorithm, but our LCM over ZBDDs requires a sub-linear time for the number of itemsets. The computational time of our method is almost the same as executing an LCM-count. We must emphasize that LCM-count does not store the itemsets, but only counts the number of solutions. On the other hand, LCM over ZBDDs generates all the solutions and stores them on the main memory as a compact ZBDD. This is an important point.

After executing LCM over ZBDDs, we can apply various algebraic operations to the ZBDD to filter or analyze the frequent itemsets [11]. Storing the results as a ZBDD will be more useful than having a large dump file of all the frequent itemsets.

Finally, we show the experimental results for generating closed itemsets in Table 3. We compared our results with the original LCM and ZBDD-growthC [10], a

Table 2. Comparison of LCM over ZBDDs with previous methods

Dataset name: min. support	#Frequent itemsets	LCMoverZBDDs		LCM-count	LCM-dump	ZBDD-growth
		ZBDD	Time(s)	Time(s)	Time(s)	Time(s)
mushroom: 1,000	123,287	760	0.50	0.49	0.64	1.78
	500 1,442,504	2,254	1.32	1.30	3.29	3.49
	300 5,259,786	4,412	2.25	2.22	9.96	5.11
	200 18,094,822	6,383	3.21	3.13	31.63	6.24
	100 66,076,586	11,584	5.06	4.87	114.21	6.72
	70 153,336,056	14,307	7.16	7.08	277.15	6.97
	50 198,169,866	17,830	8.17	7.86	357.27	6.39
T10I4D100K: 100	27,533	8,482	0.85	0.85	0.86	209.82
	50 53,386	16,872	0.97	0.92	0.98	242.31
	20 129,876	58,413	1.13	1.08	1.20	290.78
	10 411,366	173,422	1.55	1.36	1.64	332.22
	5 1,923,260	628,491	2.86	2.08	3.54	370.54
	3 6,169,854	1,576,184	5.20	3.15	8.14	386.72
	2 19,561,715	3,270,977	9.68	5.09	22.66	384.60
BMS-WebView-1: 50	8,192	3,415	0.11	0.11	0.12	29.46
	40 48,544	10,755	0.18	0.18	0.22	48.54
	36 461,522	28,964	0.49	0.42	0.98	67.16
	35 1,177,608	38,164	0.80	0.69	2.24	73.64
	34 4,849,466	49,377	1.30	1.07	8.58	83.36
	33 69,417,074	59,119	3.53	3.13	144.98	91.62
	32 1,531,980,298	71,574	31.90	29.73	3,843.06	92.47
	chess: 1,000	29,442,849	53,338	197.58	197.10	248.18
connect: 40,000	23,981,184	3,067	5.42	5.40	49.21	212.84
pumsb: 32,000	7,733,322	5,443	60.65	60.42	75.29	4,189.09
BMS-WebView-2: 5	26,946,004	353,091	4.84	3.62	51.28	118.01

Table 3. Generating closed itemsets

Dataset name: min. support	#Closed itemsets	LCMoverZBDDs		LCM-count	LCM-dump	ZBDD-growthC
		ZBDD	Time(s)	Time(s)	Time(s)	Time(s)
mushroom: 1,000	3,427	1,059	0.58	0.55	0.55	1.86
	500 9,864	2,803	1.28	1.24	1.24	3.62
	100 45,944	9,884	3.06	2.93	2.40	6.54
	50 68,468	12,412	3.48	3.35	3.50	8.71
T10I4D100K: 100	26,806	8,548	0.89	0.89	0.92	1,931.21
	50 46,993	16,995	1.03	0.99	1.03	2,455.22
	10 283,397	164,773	1.69	1.54	1.75	(>5,000)
	2 2,270,195	1,476,698	6.62	4.76	6.47	(>5,000)
BMS-WebView-1: 50	7,811	3,477	0.12	0.12	0.13	32.09
	40 29,489	11,096	0.24	0.22	0.26	58.44
	35 76,260	29,553	0.84	0.79	0.88	102.87
	32 110,800	46,667	1.94	1.86	1.98	138.22

variation of ZBDD-growth to generate closed itemsets. Since the closed (or maximal) itemsets are a very small subset of all the frequent itemsets, in this case, the performances of LCM-count and LCM-dump were not so different. Anyway, LCM over ZBDDs can efficiently generate closed itemsets using a very small overhead of the ZBDD manipulation. As well as the ZBDD of all the frequent itemsets, various postprocessing is applicable to the ZBDD of closed itemsets. For example, we can easily obtain all the “non-closed” itemsets by using a ZBDD-based difference operation between all the frequent itemsets and closed itemsets.

6 Conclusion

We proposed our “LCM over ZBDDs” algorithm for efficiently generating very large-scale all/closed/maximal frequent itemsets using ZBDDs. Our method is based on LCM, one of the most efficient state-of-the-art algorithms previously proposed. The algorithm not only enumerates the itemsets but also generates a compact output data structure on the main memory. The result can efficiently be postprocessed by using algebraic ZBDD operations.

The original LCM is known as an output linear time algorithm, but our new method requires a sub-linear time for the number of frequent patterns when the ZBDD-based data compression works well. Our experimental results indicate that the ZBDD-based method will greatly accelerate the data mining process and will lead to a new style of on-memory processing for dealing with knowledge discovery problems.

Acknowledgment. This research was partially supported by the Ministry of Education, Science, Sports and Culture (MEXT), Grant-in-Aid for Scientific Research on Priority Area: “Cyber Infrastructure for the Information-explosion Era.”

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.) Proc. of the 1993 ACM SIGMOD International Conference on Management of Data, vol. 22(2) of SIGMOD Record, pp. 207–216 (1993)
2. Bryant, R.E.: Graph-based algorithms for Boolean function manipulation. IEEE Transactions on Computers C-35(8), 677–691 (1986)
3. Goethals, B.: Survey on frequent pattern mining (2003), <http://www.cs.helsinki.fi/u/goethals/publications/survey.ps>
4. Goethals, B., Zaki, M.J.: Frequent itemset mining dataset repository. In: Frequent Itemset Mining Implementations (FIMI 2003) (2003), <http://fimi.cs.helsinki.fi/data/>
5. Han, J., Pei, J., Yin, Y., Mao, R.: Mining frequent patterns without candidate generation: a frequent-pattern tree approach. Data Mining and Knowledge Discovery 8(1), 53–87 (2004)
6. Loekit, E., Bailey, J.: Fast mining of high dimensional expressive contrast patterns using zero-suppressed binary decision diagrams. In: Proc. The Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006), pp. 307–316 (2006)
7. Minato, S.: Zero-suppressed BDDs for set manipulation in combinatorial problems. In: Proc. of 30th ACM/IEEE Design Automation Conference, pp. 272–277 (1993)
8. Minato, S.: Symmetric item set mining based on zero-suppressed BDDs. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI), vol. 4265, pp. 321–326. Springer, Heidelberg (2006)
9. Minato, S., Arimura, H.: Efficient combinatorial item set analysis based on zero-suppressed BDDs. In: Proc. IEEE/IEICE/IPSJ International Workshop on Challenges in Web Information Retrieval and Integration (WIRI-2005), April 2005, pp. 3–10 (2005)

10. Minato, S., Arimura, H.: frequent closed item set mining based on zero-suppressed BDDs. *Trans. of the Japanese Society of Artificial Intelligence* 22(2), 165–172 (2007)
11. Minato, S., Arimura, H.: Frequent pattern mining and knowledge indexing based on zero-suppressed BDDs. In: Džeroski, S., Struyf, J. (eds.) *KDID 2006*. LNCS, vol. 4747, pp. 152–169. Springer, Heidelberg (2007)
12. Uno, T., Arimura, H.: Program codes of takeaki uno and hiroki arimura (2007), <http://research.nii.ac.jp/~uno/codes.htm>
13. Uno, T., Kiyomi, M., Arimura, H.: LCM ver.2: efficient mining algorithms for frequent/closed/maximal itemsets. In: Perner, P. (ed.) *ICDM 2004*. LNCS (LNAI), vol. 3275, Springer, Heidelberg (2004)
14. Uno, T., Kiyomi, M., Arimura, H.: LCM ver.3: collaboration of array, bitmap and prefix tree for frequent itemset mining. In: *Proc. Open Source Data Mining Workshop on Frequent Pattern Mining Implementations 2005* (2005)
15. Uno, T., Uchida, Y., Asai, T., Arimura, H.: LCM: an efficient algorithm for enumerating frequent closed item sets. In: *Proc. Workshop on Frequent Itemset Mining Implementations (FIMI 2003)* (2003), <http://fimi.cs.helsinki.fi/src/>
16. Zaki, M.J.: Scalable algorithms for association mining. *IEEE Trans. Knowl. Data Eng.* 12(2), 372–390 (2000)

Unusual Pattern Detection in High Dimensions

Minh Quoc Nguyen, Leo Mark, and Edward Omiecinski

School of Computer Science
College of Computing
Georgia Institute of Technology
Atlanta GA 30332, USA
{quocminh,leomark,edwardo}@cc.gatech.edu

Abstract. In this paper, we present an alternative approach to discover interesting unusual observations that can not be discovered by outlier detection techniques. The unusual pattern is determined according to the deviation of a group of observations from other observations and the number of observations in the group. To measure the degree of deviation, we introduce the concept of adaptive nearest neighbors that captures the natural similarity between two observations. The boundary points determined by the adaptive nearest neighbor algorithm are used to adjust the level of granularity. The adaptive nearest neighbors are then used to cluster the data set. Finally, we ran experiments on a real life data set to evaluate the result. According to the experiments, we discovered interesting unusual patterns that are overlooked by using outlier detection and clustering algorithms.

1 Introduction

Data mining is the process of discovering meaningful nontrivial patterns in large data sets. In this field, clustering analysis plays an important role. The clustering algorithms divide the similar observations into groups in order to extract the common patterns of the data. In order to learn the general patterns, small clusters and nondominant patterns are discarded or simply undetected. Despite their relatively small size, these clusters may be invaluable because their nondominant patterns may reveal important knowledge. Network Intrusion, malicious computer activity and fraudulent transaction detection are the typical applications for this kind of problem [1]. Recently, outlier detection has emerged as an approach for discovering nondominant patterns by measuring the deviation of the outliers from the norm [2,3,4]. The top outliers will be the most interesting ones. However, outlier detection has two major drawbacks. First, the false alarm rate is high. Even for a good outlier detection algorithm that can discover all true outliers in terms of the deviation from the norm, most outliers except for extreme ones are unimportant. This is the nature of an outlier detection approach because the outliers are defined based on the distance between the outliers and the norm. Noise is also classified as an outlier for its deviation from the norm. In real life, domain experts are often required to investigate and analyze the

outliers manually in order to understand their meaning. If the algorithm returns many outliers, which is likely in large and heterogeneous data sets, this approach becomes difficult when the interesting outliers do not always appear in the top outliers. Another drawback is that each point can be considered an outlier by itself in high dimensions. The volume of the data space grows exponentially when the dimensionality increases [5]. In other words, the data is very sparse and the density of the data set approaches zero. As a result, except for extreme outliers, using an outlier detection method to discover novel patterns is difficult in high dimensional and heterogeneous data.

To overcome the limitations associated with outlier detection, we propose to use the number of similar surrounding observations that deviate from others as a metric to measure the level of interestingness instead of the degree of deviation metric. From this perspective, all non extreme outliers are equal even though their rankings are different because most observations in high dimensions are outliers by themselves. However, when some outliers start to form a small cluster, they are not simply noise and do not appear as outliers accidentally. They indicate interesting unusual behaviors in the data set. One may ask if we can apply several clustering algorithms on the top outliers to cluster them in order to discover their pattern. As shown in the experiments, it is not the case. The interesting observations can be the outliers with low rankings and they are often removed from the list of top outliers. As a result, the clustering algorithms can not detect those important clusters.

In this paper, we introduce an algorithm that can discover small clusters in high dimensional and heterogeneous datasets. We have shown that our algorithm can effectively discover these clusters. In addition, our algorithm has discovered novel patterns based on our proposed metric of interestingness for unusual observations

2 Related Work

The closest work to our approach is that of density-based clustering algorithms. Among the well-known algorithms, i.e. DBSCAN, Chamelon, CURE, shared nearest neighbor (SNN), SNN [6] shows the best performance because it can find clusters of different sizes, shapes and densities. This algorithm is based on the concept of core points and the number of strong links of the core points. A point is a core point if the number of strong links exceeds a given threshold. A core point will be the representative of the cluster. Any point that has the strength of the link with a core point exceeding a certain threshold will be in the same cluster as the core point. Finally, a point that has a neighbor in a cluster and the strength of the link between them is greater than a threshold will be put into the same cluster as its neighbor. The algorithm works very well for a two dimensional data set. The algorithm can find small sized clusters but it is sensitive to the deviation of a point from its neighbors. In high dimensions, the clusters are broken into many tiny clusters. In contrast, our algorithm separates

a point from a cluster only if it deviates largely from the other points, which makes the algorithm more suitable for unusual pattern detection.

Another similar work is outlier detection. Recently, Breunig et al [2] introduced the local based outlier factor (LOF) which can detect local outliers. The main idea is that if a point is inside a tight cluster, its LOF value should be close to one and if it is outside the cluster or non-tight area, LOF should be greater than one. A point with LOF greater than one is an outlier. In high dimensionality, the clusters are no longer tight as assumed [5] and LOF becomes unbounded. Virtually, most observations are local outliers according to LOF.

3 Our Approach

Our approach is based on a variation of k-nearest neighbors (KNN) and the concept of dual-neighbor to cluster the data set. In clustering, KNN is used to cluster the data set by constructing a list of k nearest neighbors for each point in the data set. The distance from a point to its k^{th} -nearest neighbor is considered as its neighborhood distance. A point and its neighbors are considered to be similar to each other. The definition of similarity can be misleading since the close points may not be actually close to each other as illustrated in figure 1(a). Point q belongs to a dense region while point p is in a less dense region. With $k = 5$, s is in the list of KNNs of p and s is considered to be similar to p. However, as shown in the figure, s is not similar to p because the distance between q and its nearest neighbors is less than that between q and p. Those two dissimilar points will be in the same cluster.

To solve the problem, Jarvis and Patrick introduced the concept of shared nearest neighbor [7]. The strength of the similarity between two points is measured by the number of nearest neighbors shared between them. Two points belong to the same cluster if the strength of the link exceeds a certain threshold. The clustering algorithm can produce excellent results. However, it is non-trivial to select an appropriate value of k and to justify the results of SNN in high dimensions. Ertoz et al improved the SNN by introducing the topic threshold. A point with the number of strong links exceeding the topic threshold will represent its neighbors. Their clustering algorithm is based on the number of strong links and the link strength of the point in the data set. In high dimensions, the points in the small clusters can not have the number of strong links sufficient enough to form a cluster. The points in this cluster will be broken into smaller clusters even though they may be only slightly different from other points. Another problem is that the parameter k is the same for all points in data set. As illustrated in figure 1(a), the result will be inconsistent with a global parameter k. Figure 1(a) illustrates a simplified case when k is small. In the figure, the distance from p to its 4^{th} -nearest neighbor is twice the distance from q to its 4^{th} -nearest neighbor even though the distance from p and q to their 2^{th} nearest neighbors are the same. The volumes of k-distances of p and q will be different significantly with a small increase in k.

In this paper, we propose the use of Adaptive Nearest Neighbors (ANN) to define the neighborhood distance. The approach has three parameters to fine tune the adaptive neighborhood distance. From our perspective, the concept of neighborhood distance of a point, say p , is a relative concept since it can not be defined without surrounding points. As illustrated in figure 1(a), the neighborhood distance of p is greater than that of q because the first two nearest neighbors of p are farther than those of q .

With this observation, the first few nearest neighbors are used to define the neighborhood distance. Those neighbors are called the initial neighbors or i -neighbors in short. The distance from p to its i -neighbors is called i -distance. The i -distance defines the minimum neighborhood distance of p regardless of k . When p is in a dense cluster, the i -distance tends to be smaller.

The next parameter, α , is used to control the neighborhood distance around p . In figure 1(b), r, s and t are i -neighbors of p whereas q is the 4th nearest neighbor of p . First, we project r, s and t on the line passing two points p and q . ph is chosen for it is the longest projected line segment of r, s and t on pq . If the ratio between the line segment pq and ph is less than α , then q is included in the adaptive nearest neighbor list of p , denoted by $ANN(p)$. This process is repeated until there is a point w in the k -nearest neighbor list of p whose ratio is greater than α . Point w and all the nearest neighbors of p farther than w are excluded from the adaptive nearest neighbor list of p . Point w is called the α -boundary point. α controls the local maximum variation of the nearest neighbors. The idea of α is that the neighbor should be excluded from the list of nearest neighbors when it is significantly different from the others in the list and α measures the level of differences.

The last parameter is to adjust the neighborhood distance. For the small data set as in figure 1(a), it makes sense to cluster it into two distinct clusters. But in a larger data set, the two clusters should be merged into one if the distinction between them is small compared with others. The boundary points can be used for controlling the granularity. We use the parameter z for this. The procedure for constructing the lists of ANNs is modified as follows. Instead of stopping the construction of ANN list for p when a boundary point is reached, we continue to put it into the ANN list of p . The process is stopped when z equals the number of times we reach the boundary points. The algorithm achieves the finest granularity level when $z = 1$. The detailed procedure for constructing the ANN list is described in algorithm 1. In algorithm 1, s is the number of i -neighbors

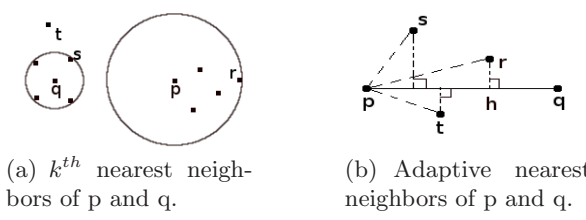


Fig. 1. Nearest neighbors of p and q

and z is the granularity tuning parameter. Also, k is the maximum number of nearest neighbors that are computed for a point p .

With adaptive nearest neighbors, we can define the neighborhood distance of a point independent of k with different levels of granularity. This neighborhood distance is called the adaptive neighbor distance, denoted by $a - distance$. According to the discussion above, we can say that any point within the adaptive neighborhood of a point p is truly a natural neighbor of p . Also, we observe that the similarity must be a mutual relation. In other words, if two points are considered naturally close to each other, they should be in the list of ANNs of each other. We formally define the closeness as follows:

Definition 1. $dual(p, q) \equiv true \iff p \in ANN(q) \wedge q \in ANN(p)$

In the definition, two points are considered neighbors to each other when they have a dual-neighbor relationship. With this definition, we can address the problem of KNN as illustrated in [1\(a\)](#) when p and q have no dual-neighbor relationship where $z = 1$. In a coarser granularity level, i.e. $z = 2$, p and q become neighbors to each other. Another useful concept is the indirect dual-neighbor relationship.

Definition 2. $indual(p, q) \equiv true$

$$(i) \quad dual(p, q) \equiv true, \text{ or}$$

$$(ii) \quad \exists r \in D : dual(p, r) \equiv true \wedge indual(r, q) \equiv true$$

As discussed above, we are interested in discovering unusual patterns. With the definition of the indirect dual-neighbor, we can formally define the concepts of usual and unusual patterns as follows:

Definition 3.

Definition 4.

In this paper, ANN and the dual-neighbor are used to define the similarity between two points. The indirect dual-neighbor shows the indirect similarity between two observations belonging to the same pattern. With the definitions of unusual pattern, the clustering criteria of our approach is stated as follows:

$$indual(p, q) \equiv true$$

This definition implies the chain affect and it can produce very large clusters. This, however, is acceptable because the observations in large clusters are usual. As mentioned above, we are interested in discovering unusual patterns. To be unusual, the observations should deviate from other usual patterns. Therefore, the chain affect will have no impact on the results for discovering unusual patterns.

The parameters i -neighbors, α and z play an important role in defining the level of similarity. In a uniformly distributed region, the choice of the number of i -neighbors has less affect since all points should belong to the same cluster. The concept of i -neighbor is useful in non-uniformly distributed regions. In this case, the number of i -neighbors should be small, which is usually less than 10. The parameter α is used to control the local variance of the neighborhood distance according to the i -neighbors. The parameter α defines the upperbound of the acceptable deviation between the neighbors. The last parameter is z which is used for adjusting the level of granularity. When $z = 1$, we can see all natural clusters in terms of ANN. When z is increased, the nearby clusters are merged together. In practice, the number of i -neighbors and α are less important than z since they can be easily selected without affecting the results. Intuitively, we can set $\alpha = 1.2$ and $z \in [2, 4]$.

Algorithm 1. Adaptive Nearest Neighbors

```

1: function ANN( $p$ )
2:   for  $i \leftarrow s, k$  do
3:      $r \leftarrow i^{th}neighbor(p)$ 
4:      $\tau_{max} \leftarrow 0$ 
5:      $\pi_{max} \leftarrow 0$ 
6:     for  $j \leftarrow 1, (i - 1)$  do
7:        $q \leftarrow j^{th}neighbor(p)$ 
8:        $\pi \leftarrow \overrightarrow{pq} \overrightarrow{pr} / \|\overrightarrow{pr}\|$ 
9:        $\tau \leftarrow \|\overrightarrow{pr}\| / \pi$ 
10:      if  $\pi > \pi_{max}$  then
11:         $\pi_{max} \leftarrow \pi$ 
12:         $\tau_{max} \leftarrow \tau$ 
13:         $idx \leftarrow j$ 
14:      end if
15:    end for
16:    if  $\tau_{max} > \alpha \parallel \tau_{max} = 0$  then
17:      if  $level < z$  then
18:         $level \leftarrow level + 1$ 
19:      else
20:        return all  $j^{th}neighbor(p)$ , where  $j \leq idx$ 
21:      end if
22:    end if
23:  end for
24: end function

```

Algorithm 2 shows the linear time processing steps to cluster the data set after the lists of adaptive nearest neighbors have been computed according to

algorithm [□](#). For every unclustered point, we randomly select a point to form a new cluster where the selected point is the representative of the cluster. Then, we expand the cluster by including all the dual-neighbors and the indirect dual-neighbors of the point into the cluster. To facilitate the algorithm, we create a stack S to store the dual-neighbors. As shown in steps 11-12, an unclustered point p is removed from the data set. Since p does not belong to any cluster, a new cluster C is created for p before pushing p onto stack S . In steps 13-16, a point q is popped from S and q is added to cluster C . Besides, all dual-neighbors of q are pushed onto the stack. Those steps are repeated until S is empty, which means the inner while loop is stopped when indirect dual-neighbors of the points in cluster C are included in the cluster.

Algorithm 2. Outcast Pseudocode

```

1: procedure OUTCAST(HashSet  $D$ )
2:   Stack  $S$ 
3:   Vector  $clsSet$ 
4:   HashSet  $C$ 
5:   while  $D \neq \emptyset$  do
6:      $p \leftarrow \text{remove } D$ 
7:      $\text{push } p \rightarrow S$ 
8:      $C \leftarrow \text{new HashSet}$ 
9:      $\text{add } C \rightarrow clsSet$ 
10:    while  $S \neq \emptyset$  do
11:       $q \leftarrow \text{pop } S$ 
12:       $\text{add } q \rightarrow C$ 
13:      for  $r \in ANN(q) \wedge \text{dual}(q, r)$  do
14:         $\text{push } r \rightarrow S$ 
15:         $\text{remove } r \text{ from } D$ 
16:      end for
17:    end while
18:  end while
19: end procedure

```

4 Experiments

In this section, we present experimental results using the SAM's Club data set [\[8\]](#). The data set contains the sales transaction data for 18 Sam's club stores between the dates of January 1 and January 31, 2000. From the sales transactions, we create a new data set of 34,250 tuples with 31 attributes. Each tuple represents a sale item and the attributes represent the total sales quantities for each individual item between the dates of January 1. The total sale varies from 0 to 16,788. The purpose of this experiment is to apply the well-known local outlier detection method LOF and the density-based clustering algorithm SNN on the data set in order to detect any unusual sales patterns. We first ran LOF on the data set to determine the top local outliers. We then ran KMEAN and SNN on the top 5% outliers to produce a summary of the outliers. We also ran SNN on the whole

data set with different values of k in the attempt to discover unusual patterns by studying the small clusters returned by SNN. We then compared the results with those from our algorithm.

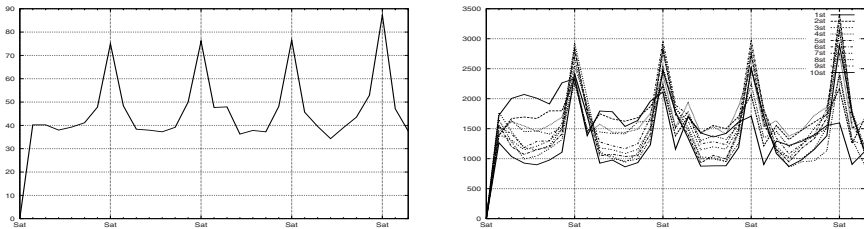
4.1 LOF, KMEAN and SNN

Figure 2(a) shows the average sales for each day in January for all items in the dataset. According to the figure, the sale follows the same pattern every week. Sales gradually decrease from the start of the week toward the middle of the week and then slightly increase toward the end of the week before achieving its peak on Saturday. The sales quickly drops on Sunday. This pattern repeats every week in January. The figure illustrates that most customers tend to go shopping on Saturdays.

For the first test, we computed the LOF values for all items. The LOF values vary greatly from 0.012 to 2681.28. There are 13990 items with LOF greater than 2 and 8495 items with LOF greater than 10. According to the LOF algorithm [2], most of items in the dataset are outliers. This confirms our remark that most data points become outliers in high dimensions due to the space sparsity. The values of the top 10 outliers and their sale information are shown in table 1(a) and figure 2(b). The strongest outlier is item 1 whose pattern deviates from the norm since its sales increase slightly on Saturdays and tends to fall toward the end of the month. For the next 9 outliers ranked by the LOF approach, the sale pattern resembles that in figure 2(a).

We take the top 5% of the items ranked by LOF to form a new dataset with the size of 1712 and then apply several clustering algorithms on the new data set. The purpose is to group the top outliers together in order to learn the common patterns of these outliers in an attempt to explain their significance. In this experiment, we use KMEAN and SNN to cluster the dataset.

Figure 3(a) shows the average sales amount and its standard deviation for items in the clusters clustered by KMEAN when $k = 20$. According to figure 3(a), KMEAN clusters the outliers into groups with different ranges of sale volume (less than 500, 500, 1000 and 1500) and the average size of the clusters is 85.6. The sale patterns for those clusters are the same as the common pattern of the whole data set. Similar results are obtained when we ran KMEAN with different values of k .



(a) For all items in the dataset.

(b) For top 10 LOF items.

Fig. 2. The average daily sale

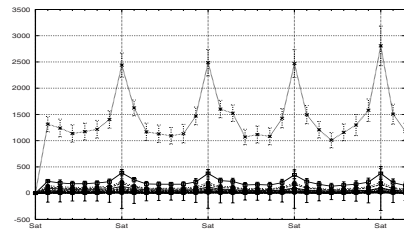
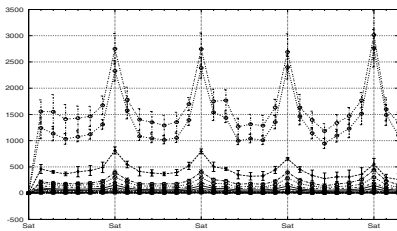
Table 1. Data Set clustered by KMEAN and SNN

(a) Top 10 LOF outliers.

Item	LOF	Item	LOF
1	2681.28	6	1798.9
2	2205.68	7	1789.0
3	1907.18	8	1710.38
4	1895.92	9	1699.56
5	1841.24	10	1686.28

(b) Clusters generated by SNN.

k	cluster	size	$\bar{\mu}$	$\bar{\sigma}$
80	86	30	73.75	195.02
	1571	41	101.07	267.16
110	85	33	122.96	300.23
	1522	82	87.76	213.85
140	85	33	122.96	300.23
	1561	112	74.66	213.85
170	14	32	90.83	267.01
	1600	155	78.66	207.4
200	1668	185	89.16	208.46



(a) Top 5% items clustered by KMEAN. (b) Top 10% items clustered by SNN.

Fig. 3. Top LOF outliers clustered by KMEAN and SNN

Figure 3(b) shows the results of SNN when $k = 20$. There are 32 clusters with the average size of 53.5. Clusters 1 and 17 are two main clusters with the size of 791 and 100 respectively. The average sale of cluster 1 ranges from 32.3 to 89.3 and its standard deviation ranges from 167 to 419.4. The sales volume of the items in the cluster are quite different even though they belong to the same cluster. As illustrated, the sales pattern of the clusters resembles the common sales pattern of the whole data set.

In the next experiment, we ran SNN on the whole data set, varying k from 20 to 200. Table 1(b) shows the list of clusters with the size greater than 30 for each k . In table 1(b), $\bar{\mu}$ is the average sales for each cluster and $\bar{\sigma}$ is the average standard deviation of the sales for the dates in January. We found that most items form a cluster by themselves and that there are at most two clusters with the size greater than 30 for each k . Also, the fourth and fifth columns of table 1(b) show that $\bar{\sigma}$ is twice $\bar{\mu}$. It means that the sale quantity varies greatly for the items in the same clusters as shown in figure 4(a). Consequently, We have found no interesting patterns in this experiment.

4.2 Outcast

Table 2(a) shows the size of the interesting clusters found by our algorithm with the granularity level 2. There is one major cluster with the size of 6203 and 16

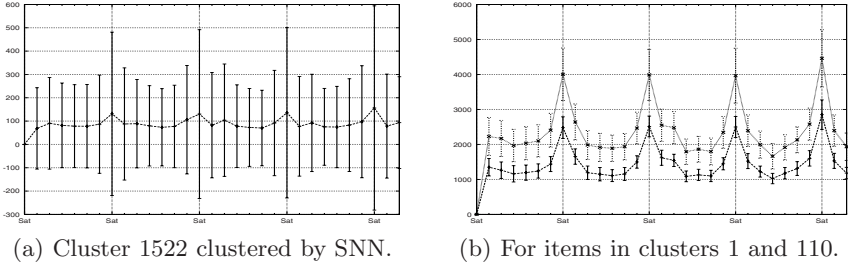


Fig. 4. The average sale volumes for each day in January

Table 2. Dataset clustered by Outcast

(a) Interesting patterns				(b) Top 4 highest items and 4 lowest items in cluster 241 ranked by LOF.					
Cluster	Size	Cluster	Size	Item	LOF	Rank	Item	LOF	Rank
93	70	652	40	3175	4.23	10974	2273	2.03	14368
363	54	663	40	3315	4.21	10984	10804	1.92	14718
241	49	444	209	1902	4.13	11068	1920	1.84	14989
				572	4.03	11128	3229	1.81	15111

small clusters with their size ranging from 40 to 209. Among them, cluster 1 and 110 (figure 4(b)) have sale patterns that resemble the common pattern. We found that the top 14 outliers recognized by LOF belong to cluster 1 and that 51 out of 58 items in cluster 1 are in the top 100 outliers.

Cluster 241 with the size of 49 is the most interesting pattern found by our algorithm. Figure 5(a) shows the sales pattern of the items in the cluster. Even though the average sale volumes of the items vary from 80.74 to 389.35, they follow the same pattern which is reversed from the common sale pattern (fig. 2(a)). The sale achieves the peak at the beginning of the week instead of on Saturday and then slightly decreases toward the weekend before reaching its lowest on Sunday. It is interesting to find that all the sales in the second week of the items in this cluster jump sharply on Friday instead of Saturday as the common pattern and then the sale drops quickly on the Saturday and Sunday. The sales on this day is almost double the sales on the peaks of the other weeks. When we further investigate the items in the clusters, we found that all of those items are cigarettes. Table 2(b) shows the top highest and lowest LOF values for items in cluster 241. Even though the items have interesting sales patterns, their LOF ranking is very low. The item with highest rank in the cluster is item 3175 and its rank is 10974th. The ranking varies greatly from 15111th to 10974th despite the fact that the sale patterns are very similar for those items.

Two other interesting patterns occur in clusters 93 and 652 as shown in figure 5(b). Even though cluster 93 resembles the weekly common sales pattern

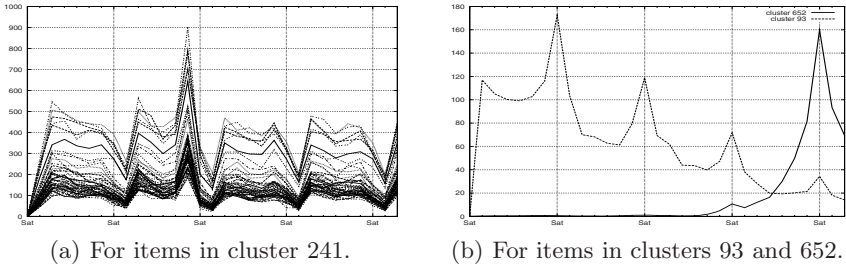


Fig. 5. The average sale volumes for each day in January

in the way that the sale is highest on Saturday as compared with the other days in the same week, the overall sales in every week tends to decrease toward the end of the month. In contrast, items in cluster 652 almost have no sale for the first three weeks. In the last week, the sales increase rapidly toward the end of the month and achieve their peak on the last Saturday of the month. Figure 6(a) shows the sale pattern for clusters 363 and 663. Those two clusters are similar to clusters 93 and 652 except that the sales for those clusters are four times less than that of clusters 93 and 652.

Figure 6(b) shows the sale patterns for clusters 60, 463, 444 and 331. Cluster 444 contains 209 items and those items have almost no sales except for a few sales on the last Saturday. The other clusters are less interesting than the ones mentioned above due to their small sale volume.

In summary, we ran experiments with different combinations of the outlier detection and clustering algorithms. With LOF, most items in the data set were classified as outliers. When examining the top 10 outliers. We found that the sales pattern of the top outlier is slightly different from the common weekly sales pattern. The top 10 outliers have high sale volumes and their sales pattern follow the weekly pattern. We then clustered the dataset with KMEAN and SNN. Those clustering algorithms divide the top 5% of the outliers into groups of different sale volumes but no interesting patterns are found. It is the same when we ran SNN on the whole data set. However, when we tested the data

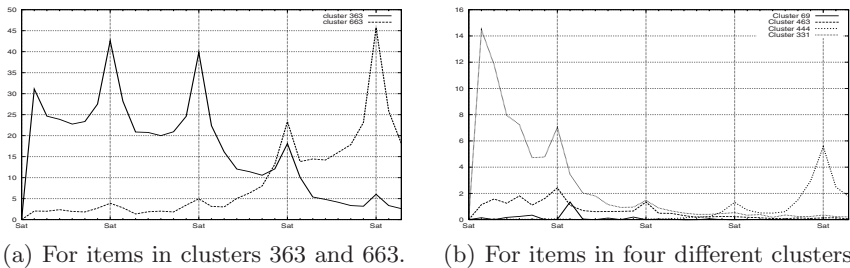


Fig. 6. The average sale volumes for each day in January

set with our algorithm, we discover six unusual patterns. Among them, the sale pattern of cluster 1 does not differ from the weekly sales pattern. We found that 89% of the items in the cluster are in the top 100 outliers ranked by LOF. Cluster 241 is the most interesting since we found that cigarette sales follow the Friday sales pattern rather than the Saturday pattern. The other four clusters do not follow the common sales pattern. The experiment confirms that interesting patterns may not be discovered by simply clustering the top outliers.

5 Conclusion

Clustering and outlier detection are two different approaches that can be used to learn general patterns and novel events. However, both of these approaches can not detect unusual patterns that appear in small clusters, which may be interesting. For most clustering algorithms, small size clusters are sacrificed in order to discover large size clusters. In contrast, the outlier detection approach simply focuses on single outliers rather than groups of outliers. Top outliers are the most interesting events. In our experiments, we have shown that top outliers are not always interesting since they may simply be noise in high dimensions, all data points may be considered outliers due to the sparsity of the data.

In this paper, we present an alternative approach for knowledge learning by introducing the concept of an unusual pattern, based on the size of the small clusters and their deviation from the common patterns. We have developed an algorithm to detect those unusual patterns. The parameters of the algorithm are used to adjust the granularity level of the output. Our experiments on a real world data set show that our algorithm can discover interesting unusual patterns which are undetected by two well-known outlier detection and clustering techniques, namely LOF and SNN, and their combination.

Acknowledgement

This research was funded in part by a grant from the Vietnam Education Foundation (VEF). The opinions, findings, and conclusions stated herein are those of the authors and do not necessarily reflect those of VEF.

References

1. Knorr, E.M., Ng, R.T.: Algorithms for mining distance-based outliers in large datasets. In: VLDB 1998: Proceedings of the 24rd International Conference on Very Large Data Bases, pp. 392–403. Morgan Kaufmann, San Francisco (1998)
2. Breunig, M.M., Kriegel, H.P., Ng, R.T., Sander, J.: LOF: identifying density-based local outliers. SIGMOD Rec 29(2), 93–104 (2000)
3. Lazarevic, A., Kumar, V.: Feature bagging for outlier detection. In: KDD 2005: Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining, pp. 157–166. ACM, New York (2005)
4. Hawkins, D.: Identification of outliers. Chapman and Hall, London (1980)

5. Beyer, K.S., Goldstein, J., Ramakrishnan, R., Shaft, U.: When is "nearest neighbor" meaningful? In: Beerl, C., Bruneman, P. (eds.) ICDT 1999. LNCS, vol. 1540, pp. 217–235. Springer, Heidelberg (1998)
6. Ertöz, L., Steinbach, M., Kumar, V.: Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In: Proceedings of the third SIAM international conference on data mining, pp. 47–58. Society for Industrial and Applied, Philadelphia (2003)
7. Jarvis, R.A., Patrick, E.A.: Clustering using a similarity measure based on shared near neighbors. *IEEE Transactions on Computers* C-22(11), 1025–1034 (1973)
8. Walton College Teradata: Walton College Teradata, <http://spartan.walton.uark.edu/sqlassistantweb1.2/>
9. Chawla, N.V., Lazarevic, A., Hall, L.O., Bowyer, K.W.: SMOTEBoost: Improving prediction of the minority class in boosting. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) PKDD 2003. LNCS (LNAI), vol. 2838, Springer, Heidelberg (2003)

Person Name Disambiguation in Web Pages Using Social Network, Compound Words and Latent Topics

Shingo Ono¹, Issei Sato¹, Minoru Yoshida², and Hiroshi Nakagawa²

¹ Graduate School of Information Science and Technology, The University of Tokyo

² Information Technology Center, The University of Tokyo

ono@r.dl.itc.u-tokyo.ac.jp, sato@r.dl.itc.u-tokyo.ac.jp,
mino@r.dl.itc.u-tokyo.ac.jp, nakagawa@dl.itc.u-tokyo.ac.jp

Abstract. The World Wide Web (WWW) provides much information about persons, and in recent years WWW search engines have been commonly used for learning about persons. However, many persons have the same name and that ambiguity typically causes the search results of one person name to include Web pages about several different persons. We propose a novel framework for person name disambiguation that has the following three components processes. Extraction of social network information by finding co-occurrences of named entities, Measurement of document similarities based on occurrences of key compound words, Inference of topic information from documents based on the Dirichlet process unigram mixture model. Experiments using an actual Web document dataset show that the result of our framework is promising.

Keywords: person name disambiguation, web people search, clustering, social network.

1 Introduction

The World Wide Web (WWW) provides much information about persons, and in recent years WWW search engines have been commonly used for learning about persons. However, ambiguity in person names (i. e. , many persons having the same name), typically causes the search results of one person name to result in Web pages about several different persons.

In this paper, the ambiguity of person name in Web pages is defined as follows. Each string appearing as a name on a Web page is a reference to a certain entity in the real world, i. e. , each name refers to an entity. The ambiguity of person name in Web pages is that person names that have the same string in many Web pages refers to different entities.

For example, if you want to know about a “George Bush” who is not the president but an ordinary person, many pages about the president that are returned as the search result may be a problem you. According to the circumstances, we may have to look once more to find Web pages about the target person among the many search results, which may be hard and time consuming work.

Hereinafter, we use a term “person name” to mean a string indicating the name of a person.

In this paper, we propose a novel framework for person name disambiguation (i. e. , the problem of clustering Web pages about persons with the same name according to the true entities.)

Our framework is based on the following three intuitions:

1. Each person has his/her own social network.
2. There are specific compound key words that characterize him/her.
3. Each person is related to some specific topics.

These intuitions led to our framework, which comprises the following steps.

First, we extract social networks by finding co-occurrences of person names with Named Entity extraction tools (NE taggers). Second, we measure document similarities based on occurrences of key compound words that are extracted by using statistics of compound nouns and their components. Third, we infer topic information from documents based on a basic topic model Unigram mixture, which is a probabilistic generative model of a document. In particular, we use Dirichlet Process Unigram Mixture (DPUM), which is an extension of unigram mixture that uses Dirichlet process. Finally, we cluster Web pages by using the above three types of features (i.e., social networks, document similarities, and documents topics.) Among these three steps, the first step is the one proposed in our previous work [14].

The remaining part of this paper is organized as follows. Section 2 and 3 explain the task definition and related works Section 4 explains our framework. Section 5 evaluates our framework with an actual Web document dataset. Section 6 summarizes our work.

2 Task Definition

Our task, the disambiguation of person names appearing on Web pages, is formalized as follows. The query (target person name) is referred to as q . The set of Web pages obtained by inputting query q to a search engine is denoted by $\mathcal{P} = \{d_1, d_2, \dots, d_k\}$. Each Web page d_i has at least one string q . Then, the j th appearance of string q on Web page d_i is assumed to be s_{ij} . Each s_{ij} indicates only one entity in the set $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ of entities in the real world having the name q . Now, the set of s_{ij} is assumed to be \mathcal{S} . We define function $\Phi : \mathcal{S} \rightarrow \mathcal{E}$. Function Φ is a mapping from the name appearing in the document to entities in the real world. In other words, Φ maps from a string to an entity. Our purpose is to find function $\check{\Phi}$ that will approximate function Φ .

The modeling above permits the same string q appearing in the same document to refer to different entities. Web pages with such properties are quite rare and dealing with them makes the system more complicated, so we decided to ignore such pages by assuming that all instances of the same string q on a certain Web page d_i refer to the same entity, i.e., for each i , there exists $e_m \in \mathcal{E}$, such that $\forall j, \Phi(S_{ij}) = e_m$. This assumption means that the same name that

appears multiple times on one page only refers to one entity. This results in a simpler model i.e., $\mathcal{F}' : \mathcal{P} \rightarrow \mathcal{E}$. In this research, our aim was to estimate \mathcal{F}' . The problem here is n (that appears in the definition of \mathcal{E}) is not known in advance. In other words, we do not know how many distinct entities have the string q . We actually estimated \mathcal{F}' by clustering Web pages.

Our system works as follows. Given query q , the system retrieves Web pages that have string q using a search engine and then disambiguates the reference. Finally, the system outputs a set of page clusters, each of which refers to a single entity.

3 Related Works

Several important works have tried to solve the task described in the previous section. Bagga and Baldwin [4] applied the vector space model to calculating similarity between names using only co-occurring words. Based on this, Niu et al. [13] presented an algorithm that uses information extraction results in addition to co-occurring words. However, these methods had only been tested on artificial small test data, leaving doubt concerning their suitability for practical use. Mann and Yarowsky [9] employed a clustering algorithm to generate person clusters based on extracted biographic data. However, this method was also only tested on artificial test data. Wan et al. [16] proposed a system that rebuilt search results for person names. Their system, called WebHawk, was aimed at practical use like our systems, but their task was somewhat different. Their system was designed for actual frequent queries. The algorithm of their system was specialized for English person name queries that consist of three words: family name, first name, and middle name. They mainly assumed queries such as “<f_i, f_j, . . . >” or “<f_i, f_j, . . . > < . . . , v_j, f_j, . . . >”, and took middle names into consideration, which may have improved accuracy. However, it would not be suitable for other types of names such as those in Japanese (consisting only of a family name and given name).

As another approach to this task, Bekkerman and McCallum [5] proposed two methods of finding Web pages that refer to a particular person. Their work consists of two distinct mechanisms: the first is based on link structure and the second uses agglomerative/conglomerative double clustering. However, they focused on disambiguating an existing social network of people, which is not the case when searching for people in real situations. In addition, our experience is that the number of direct links between pages that contain the same name are fewer than expected, so information on link structures would be difficult to use to resolve our task. Although there may be indirect links (i. e. , one page can be found from another page via other pages), it is too time consuming to find them.

4 Proposed Framework

In this section, we explain three types of features of the proposed framework: social networks, document similarities and documents topics.

4.1 Preprocessing

We eliminate noise tokens such as HTML tags and stop words. We extract local texts that appear within 100 words before and after the target person name (query). In this study, our analysis is limited to the local texts.

4.2 Extraction of Social Networks

This section explains how to extract social networks and to cluster pages by using social networks. This method was used in our previous work [14].

We use graph representation of relations between documents. Let G be an undirected graph with vertex set V and edge set E . Each vertex $v_i \in V$ corresponds to page d_i . Then, edge e_{ij} represents that d_i and d_j refer to the same entity.

On the other hand, social networks can be seen as another graph structure in which each node represents an entity, each edge represents that the fact that two entities have a relation, and each connected component represents one social network. We assume that every pair of entities that appears in the same page have a relation. We also assume that the same name in the same social network refers to the same entity.

In graph G , we make edge e_{ij} if the same person name m (other than q) appears in both of d_i and d_j because, roughly speaking, this means that both of d_i and d_j are related to m (i.e., both are in the same social network which m belongs to.)¹ Moreover, we utilize the place names and organization names that appear near the position of the target person name to extract more information of social networks. the place names and organization names can be discriminating as well as person names around the target parson name. To identify person, place, and organization names, we used `pos`, `pos`, `pos`² as an NE tagger which tags each proper noun according to context, such as person name, place name, or organization name.

The clustering algorithm by Social Networks is presented below.

Procedure: Clustering by Social Networks(SN)

1. From all documents d_j ($1 \leq j \leq k$), extract person names (full name), place names and organization names with a NE tagger.
2. Calculate SN similarity $\text{sim}_{\text{SN}}(d_x, d_y)$ as follows:

$$\begin{aligned} \text{sim}_{\text{SN}}(d_x, d_y) &= \mu * (\text{number of person names appearing in both } d_x \text{ and } d_y) \\ &+ \nu * (\text{number of place or organization names appearing in both } d_x \text{ and } d_y) \end{aligned}$$

3. If $\text{sim}_{\text{SN}}(d_x, d_y) \geq \theta_{\text{SN}}$, then $\Phi'(d_x) = \Phi'(d_y)$, where θ_{SN} is the threshold.

¹ We ignore the ambiguity of m by assuming that it is rare that two or more social networks contain the same person name pair (q, m) .

² <http://chasen.org/~taku/software/cabochoa/>

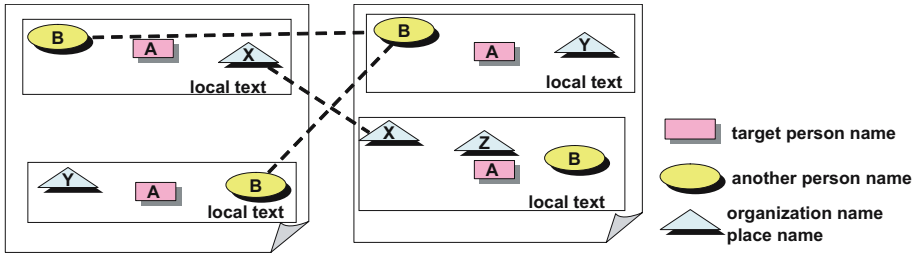


Fig. 1. Co-occurrence of Person Name, Place Name and Organization Name

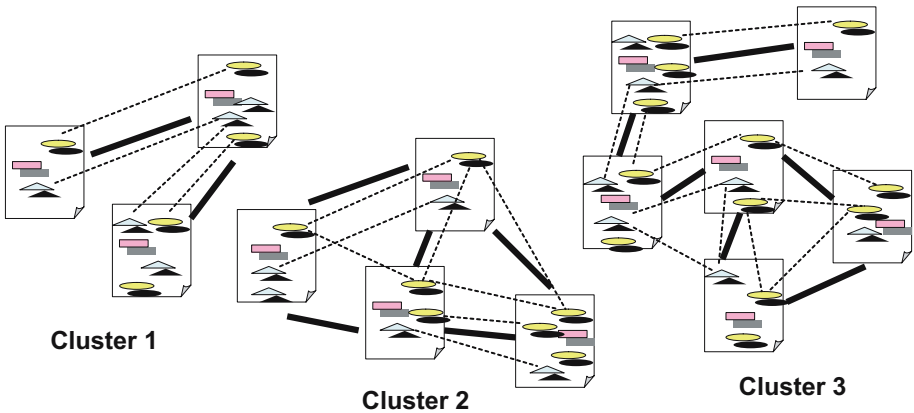


Fig. 2. Clusters constructed by Social Networks corresponding to μ , ν and θ_{SN}

This is where μ and ν are parameters for weighting and θ_{SN} is a threshold. In this study, μ and ν are constrained as $\mu \gg \nu$. The constraint says that person names are more important than other names.

$\Phi'(d_x) = \Phi'(d_y)$ means two pages, d_x and d_y , are to be in the same cluster and clustering is done as follows. Let G be an undirected graph with vertex set V and edge set E . Each vertex $v_i \in V$ corresponds to page d_i . The result of the above procedure gives edge set E . Each edge $e_{ij} \in E$ exists if and only if constraint $\Phi'(d_i) = \Phi'(d_j)$ was added in Step 3 of the above algorithm. Then, graph $G = \langle V, E \rangle$ has some connected components. Each connected components means one cluster of Web pages all of which refer to the same entity.

In Fig. 1, the dotted-lines show occurrences of the same person name, place name or organization name. In Fig. 2 the solid lines show the connection among documents whose SN similarities are over the threshold.

4.3 Document Similarities Based on Compound Key Words

This section explains how to measure document similarities based on key compound words and to cluster documents by similarity.

First, we calculate an importance score of compound words in a document with the method proposed by Nakagawa et al. [11]. Next, we construct a compound word vector $\mathbf{c}\mathbf{w}\mathbf{v} = (s_1, s_2, \dots, s_{V_c})$ for each document where $\{1, 2, \dots, V_c\}$ are the indices of the compound words in documents and s_v is the score of compound word v . Then, we measure the document similarity by using the scalar product of the compound word vectors. Finally, we cluster the documents by the similarity and a threshold.

The importance score for the compound words is calculated as follows: Let $CW(= W_1W_2 \dots W_L)$ be a compound word, where W_i ($i = 1, 2, \dots, L$) is a simple noun. $f(CW)$ is the number of independent occurrences of compound word CW in a document where “independent” occurrence of CW means that CW is not a part of any longer compound nouns. The importance score of compound word CW is

$$Score(CW) = f(CW) \cdot LR(CW), \tag{1}$$

$LR(CW)$ is defined as follows:

$$LR(CW) = \left(\prod_{i=1}^L (LN(W_i) + 1)(RN(W_i) + 1) \right)^{\frac{1}{2L}} \tag{2}$$

$LN(W_i)$ and $RN(W_i)$ are the frequencies of nouns that directly precede or succeed simple noun W_i .

This system can be obtained as “Term Extraction System³”.

The clustering algorithm by key compound words is presented below.

Procedure: Clustering by Compound Key Words(CKW)

1. From all documents d_j ($1 \leq j \leq M$), extract key compound words and construct compound word vectors $\mathbf{c}\mathbf{w}\mathbf{v}_j$ ($1 \leq j \leq k$) with Term Extraction System .
2. Calculate CKW similarity $\text{sim}_{\text{CKW}}(d_x, d_y)$ as,

$$\text{sim}_{\text{CKW}}(d_x, d_y) = \mathbf{c}\mathbf{w}\mathbf{v}_x \cdot \mathbf{c}\mathbf{w}\mathbf{v}_y$$

3. If $\text{sim}_{\text{CKW}}(d_x, d_y) \geq \theta_{\text{CKW}}$, then $\Phi'(d_x) = \Phi'(d_y)$, where θ_{CKW} is the threshold.

Having constrains $\Phi'(d_x) = \Phi'(d_y)$, clustering is done in the same way as Social Networks.

4.4 Estimate Latent Topic of Document

In this paper, we assume that pages referring to the same entity have the same latent topic that indicates a word distribution. Therefore, inferring the latent topic of a page allows the pages that have the same topic to be categorized into the same cluster.

³ <http://www.r.dl.itc.u-tokyo.ac.jp/~nakagawa/resource/termext/atr-e.html>

As a clustering algorithm that can treat latent topics, we adopt unigram mixture that is a basic topic model [12]. Moreover, we use unigram mixture expanded by Dirichlet process [7]: Dirichlet process unigram mixture(DPUM). DPUM can estimate the number of latent topics corresponding to a set of pages. In the person name disambiguation, the number of true entities(topics) is unknown at first, so DPUM is suitable to our purpose.

Unigram mixture is a probabilistic generative model of a document based on unigram model, which assumes that the words of every document are drawn independently from a single multinomial distribution. In unigram mixture, each document is generated by the topic-conditional multinomial distribution $p(w|z, \phi)$. $z \in \{1, 2, \dots, T\}$ is a latent topic and T is the number of latent topics. $\phi = \{\phi_t\}_{t=1}^T$ is the parameter of the multinomial distribution corresponding to latent topic t where $\phi_t = (\phi_{t1}, \phi_{t2}, \dots, \phi_{tN_v})$ and N_v is the number of vocabulary items and ϕ_{tw} is the probability that word w is generated from topic t . It is a problem that the number of latent topics is unknown in advance. To solve this problem, a nonparametric Bayes model using Dirichlet process was proposed [7][6]. This model can change the model structure (the number of latent topics, etc...) in correspondence with the data. A mixture model expanded by Dirichlet process is called Dirichlet process mixture(DPM) [1].

Sethuraman provides a constructive representation of Dirichlet process as stick-breaking process [15]. By using Stick-breaking process, the effective learning algorithm of DPM can be proposed [6].

The stick-breaking process is based on countably infinite random variables $\{\beta_t\}_{t=1}^\infty, \{\pi_t\}_{t=1}^\infty$ and $\{\phi_t\}_{t=1}^\infty$ as follows:

$$\beta_t \sim Beta(1, \alpha_0), \pi_t = \beta_t \prod_{i=1}^{t-1} (1 - \beta_i) \tag{3}$$

$$\phi_t \sim G_0 \tag{4}$$

α_0 is a concentrate parameter and G_0 is a base measure of Dirichlet process. In DPUM, G_0 is Dirichlet distribution $p(\phi|\lambda)$ where λ is a parameter of Dirichlet distribution. *Beta* is a beta distribution.

We write $\boldsymbol{\pi} (= \{\pi_t\}_{t=1}^\infty) \sim SB(\boldsymbol{\pi}; \alpha_0)$ if π is constructed by Eq. (3).

The process of generating a document in DPUM by using the stick-breaking process is as follows:

1. Draw $\boldsymbol{\pi} (= \{\pi_t\}_{t=1}^\infty) \sim SB(\boldsymbol{\pi}; \alpha_0)$
2. Draw $\phi_t \sim G_0$ ($t = 1, 2, \dots, \infty$)
3. For each document d :
 - (a) $z_d \sim \text{Dir}(\mathbf{z}; \boldsymbol{\pi})$
 - (b) For each of the N_d words w_{dn} : $w_{dn} \sim p(w|z_d, \boldsymbol{\phi})$

Note that Dir is a multinomial distribution and $p(w = v|z = t, \boldsymbol{\phi}) = \phi_{tv}$.

Therefore, DPUM can be formulated in the joint probability distribution as follows.

$$p(D, \mathbf{z}, \boldsymbol{\pi}, \boldsymbol{\phi} | \alpha_0, \lambda) = p(\boldsymbol{\pi} | \alpha_0) p(\boldsymbol{\phi} | \lambda) \prod_{d=1}^M p(\mathbf{w}_d | z_d, \boldsymbol{\phi}) p(z_d | \boldsymbol{\pi}) \quad (5)$$

$$p(\mathbf{w}_d | z_d, \boldsymbol{\phi}) = \prod_{n=1}^{N_d} p(w_{dn} | z_d, \boldsymbol{\phi}) \quad (6)$$

M is the number of documents. N_d is the number of words in a document d . $\mathbf{w}_d = (w_{d1}, w_{d2}, \dots, w_{dN_d})$ is a sequence of N_d words where w_{dn} denotes the n th word in the sequence. $p(\boldsymbol{\pi} | \alpha_0)$ is $SB(\boldsymbol{\pi} | \alpha_0)$.

For inference of latent topics in DPUM, we adopt Variational Bayes inference, which provides a deterministic method [3]. Blei et al. proposed a framework of Variational Bayes inference for DPM that was restricted to an exponential family mixture and was formulated by Stick-breaking process [6]. This inference scheme does not need to set the number of latent topics, but it does need to set a maximum number of latent topics due to computational cost.

5 Experimentation of Proposed Framework

5.1 Data Set

As we mentioned, the English corpus for the Name Disambiguation task is developed in WePS [2]. Because our system targets Japanese Web pages, however, we developed an original Japanese Web page test set for this task as follows.

We first input Japanese person name queries into a search engine. Some of the person queries were chosen from among ambiguous popular names. For example, “Taro Kimura” is a very common name in Japan, and we found there were many people called “Taro Kimura”, including a famous commentator, a member of the Diet, a translator, and a schoolmaster. Some other queries were selected from persons in our laboratory, and other person name queries were generated automatically.

Second, we tried to extract Web pages containing these names. We retrieved these pages with a search engine. If the query hit many pages, we collected the top 100-200 Web pages.

Finally, these pages were manually annotated⁴. Annotators removed pages that violated our assumption that one page refers to only one entity. As a result, we collected 5015 Web pages on 38 person names, and all page references were clarified.

5.2 Evaluation

Precision (P), recall (R), and F-measure (F) were used as the evaluation metrics in our experiments. All metrics were calculated as follows [8]. Assume $\mathcal{C} =$

⁴ Note that the annotators were unable to determine pages perfectly; there were a few pages that were too ambiguous to determine. To standardize the results, each ambiguous page was regarded as referring to another independent entity, i.e., each of them composed a cluster by itself in correct grouping.

$\{C_1, C_2, \dots, C_n\}$ is a set with correct grouping, and $\mathcal{D} = \{D_1, D_2, \dots, D_m\}$ is a set for the result of clustering, where C_i and D_j are sets of pages. For each correct cluster $C_i (1 \leq i \leq n)$, we calculated precision, recall, and F-measure for all clusters $D_j (1 \leq j \leq m)$ as

$$P_{ij} = \frac{|C_i \cap D_j|}{|D_j|}, R_{ij} = \frac{|C_i \cap D_j|}{|C_i|}, F_{ij} = \frac{2P_{ij}R_{ij}}{P_{ij} + R_{ij}}.$$

The F-measure of $C_i (F_i)$ was calculated by $F_i = \max_j F_{ij}$. Using $j' = \operatorname{argmax}_j F_{ij}$, P_i and R_i were calculated as $P_i = P_{ij'}, R_i = R_{ij'}$.

The entire evaluation was conducted by calculating the weighted average where weights were proportional to the number of elements in the clusters, calculated as

$$F = \sum_{i=1}^n \frac{|C_i|F_i}{|\mathcal{C}|},$$

where $|\mathcal{C}| = \sum_{i=1}^n |C_i|$. The weighted average precision and recall were also calculated in the same way for the F-measure.

5.3 Baseline

Baseline is a clustering method that uses well-known document similarities by word frequency.

First, we construct a word frequency vector $\mathbf{wfv}_j = (f_1, f_2, \dots, f_W)$ for each document where $\{1, 2, \dots, W\}$ are the indices of the vocabulary in documents and f_w is the frequency of word w in a document d_j . Then, we measure the document similarity by using the scalar product of the word frequency vectors: $\operatorname{sim}_{\text{Base}}(d_x, d_y) = \mathbf{wfv}_x \cdot \mathbf{wfv}_y$. Finally, we cluster the documents by the similarity $\operatorname{sim}_{\text{Base}}$ and a threshold θ_{Base} . The clustering is done in the same way as Compound Key Words.

Moreover, we tested the k -means algorithm in which all documents are categorized into different clusters, that is, there are not any documents that are categorized into the same cluster.

5.4 Experimentation

We investigated which of the k -means algorithm (SN), Compound Key Words (CKW), k -means algorithm (DP) or their combinations were the best. Combinations of two or three methods means different methods together used together.

More precisely, the result of the combination of SN and CKW is given by considering graph $G = \langle V, E_{SN} \cup E_{CKW} \rangle$ and $G = \langle V, E_{SN} \cap E_{CKW} \rangle$ where $G_{SN} = \langle V, E_{SN} \rangle$ is the result for SN and $G_{CKW} = \langle V, E_{CKW} \rangle$ is the result for CKW. DP needs to initialize the latent topic z_d of a document and the maximum number of latent topics. Since we had to determine thresholds θ_{SN} and θ_{CKW} , we used 5-fold cross validation for the evaluation of SN/CKW methods

Table 1. Results: Average of 38 queries

	F	P	R
No Cluster	0. 2996	1. 0000	0. 2536
Baseline	0. 5409	0. 6668	0. 6950
DP	0. 3853	0. 8443	0. 3526
SN	0. 7163	0. 9000	0. 6692
CKW	0. 6974	0. 8195	0. 7050
SN \cap CKW	0. 6196	0. 9469	0. 5180
SN \cup CKW	0. 7443	0. 8328	0. 7683
SN+DP	0. 7388	0. 8994	0. 6975
CKW+DP	0. 7048	0. 8454	0. 6944
(SN \cap CKW)+DP	0. 6535	0. 9457	0. 5542
(SN \cup CKW)+DP	0. 7529	0. 8496	0. 7640

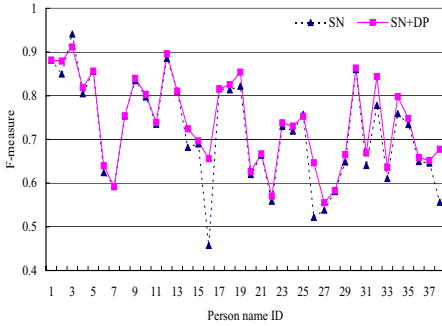


Fig. 3. F-measure of SN and SN+DP with respect to each person name

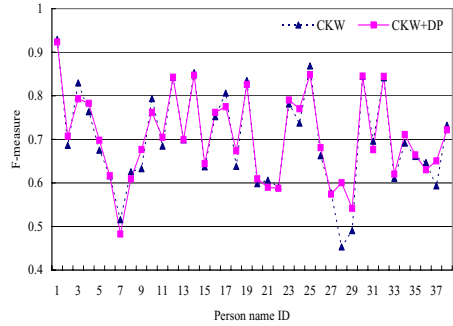


Fig. 4. F-measure of CKW and CKW+DP with respect to each person name

or their combinations. θ_{SN} and θ_{CKW} were estimated to maximize training set F-measure, and then test set F-measure was calculated using these estimated parameters.

When DP was applied in a stand-alone way, the latent topic was initialized randomly and the maximum number of topics was set to 100. When DP was combined with SN/CKW methods, SN/CKW methods were applied first, and DP was then initialized with the SN/CKW results. That is, we regarded the clusters constructed by SN/CKW methods as the initial latent topics of DP and applied DP. In this case, the maximum number of latent topics was set to the number of the cluster constructed by SN/CKW methods.

Table 1 lists the results of an average of 38 queries. Fig. 3-6 shows F-measure of SN, CKW, SNUCKW, SN+DP, CKW+DP and (SNUCKW)+DP with respect to each person name. According to the results, Either SN or CKW showed the great improvement from the baseline. In addition, they seem to employ distinct type of information to a certain extent because SNUCKW shows four to

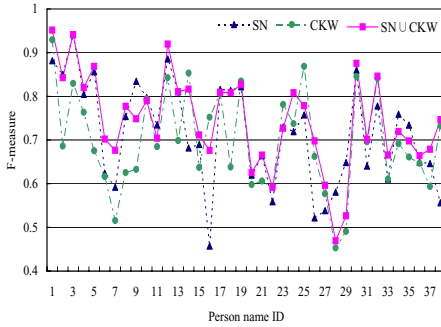


Fig. 5. F-measure of SN, CKW and SNUCKW with respect to each person name

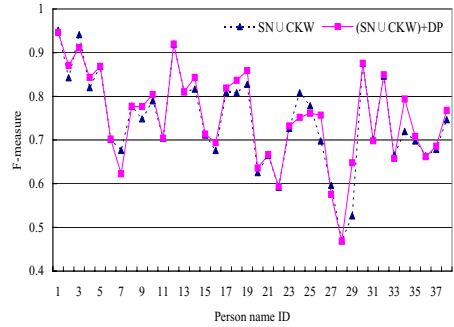


Fig. 6. F-measure of SNUCKW and (SNUCKW)+DP with respect to each person name

five points improvement from SN or CKW alone. The fact that DP also improves SN or CKW on F-measure means that DP introduces another aspect of the information, i.e., documents topics. As expected from these results, proposed methods (SNUCKW)+DP showed the highest performance on F-measure among others.

6 Conclusion

We propose a novel framework for person name disambiguation that has the following three components processes: social networks, document similarities by compound key words and documents topics. Experiments using an actual Web document dataset show that the result of our framework is promising because our framework uses distinct type of information potentially being within documents.

Acknowledgments. This research was funded in part by Category “A” of “Scientific Research” Grants in Japan.

References

1. Antoniak, C.E.: Mixtures of Dirichlet Processes with Applications to Bayesian Nonparametric Problems. *The Annals of Statistics* 2(6) (1974)
2. Artiles, J., Gonzalo, J., Sekine, S.: The SemEval-2007 WePS Evaluation: Establishing a benchmark for the Web People Search Task. In: *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pp. 64–69 (2007)
3. Attias, H.: Learning parameters and structure of latent variable models by Variational Bayes. In: *Proceedings of Uncertainty in Artificial Intelligence* (1999)
4. Bagga, A., Baldwin, B.: Entity-Based Cross-Document Coreferencing Using the Vector Space Model. In: *Proceedings of COLING-ACL 1998*, pp. 79–85 (1998)

5. Bekkerman, R., McCallum, A.: Disambiguating Web Appearances of People in a Social Network. In: Proceedings of WWW 2005, pp. 463–470 (2005)
6. Blei, D.M., Jordan, M.I.: Variational inference for Dirichlet process mixtures. *Journal of Bayesian Analysis* 1(1), 121–144 (2005)
7. Ferguson, T.S.: A Bayesian Analysis of Some Nonparametric Problems. *The Annals of Statistics* 1(2) (1973)
8. Larsen, B., Aone, C.: Fast and effective text mining using linear-time document clustering. In: Proceedings of the 5th ACM SIGKDD, pp. 16–22 (1999)
9. Mann, G.S., Yarowsky, D.: Unsupervised Personal Name Disambiguation. In: Proceedings of CoNLL 2003, pp. 33–40 (2003)
10. Morton, T.S.: Coreference for NLP Applications. In: Proceedings of ACL-2000, pp. 173–180 (2000)
11. Nakagawa, H., Mori, T.: Automatic Term Recognition based on Statistics of Compound Nouns and their Components. *Terminology* 9(2), 201–219 (2003)
12. Nigam, K., McCallum, A.K., Thrun, S., Mitchell, T.M.: Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning* 39, 103–134 (2000)
13. Niu, C., Li, W., Srihari, R.K.: Weakly Supervised Learning for Cross-document Person Name Disambiguation Supported by Information Extraction. In: Proceedings of ACL-2004, pp. 598–605 (2004)
14. Ono, S., Yoshida, M., Nakagawa, H.: NAYOSE: A System for Reference Disambiguation of Proper Nouns Appearing on Web Pages. In: Ng, H.T., Leong, M.-K., Kan, M.-Y., Ji, D. (eds.) AIRS 2006. LNCS, vol. 4182, pp. 338–349. Springer, Heidelberg (2006)
15. Sethuraman, J.: A Constructive Definition of Dirichlet Priors. *Statistica Sinica* 4, 639–650 (1994)
16. Wan, X., Gao, J., Li, M., Ding, B.: Person Resolution in Person Search Results: WebHawk. In: Proceedings of CIKM 2005, pp. 163–170 (2005)

Mining Correlated Subgraphs in Graph Databases

Tomonobu Ozaki¹ and Takenao Ohkawa²

¹ Organization of Advanced Science and Technology, Kobe University

² Graduate School of Engineering, Kobe University

1-1 Rokkodai-cho, Nada, Kobe, 657-8501, Japan

{tozaki@cs., ohkawa@}kobe-u.ac.jp

<http://www25.cs.kobe-u.ac.jp/>

Abstract. In this paper, we bring the concept of hyperclique pattern in transaction databases into the graph mining and consider the discovery of sets of highly-correlated subgraphs in graph-structured databases. To discover frequent hyperclique patterns in graph databases efficiently, a novel algorithm named HSG is proposed. By considering the generality ordering of subgraphs, HSG employs the depth-first/breadth-first search strategy with powerful pruning techniques based on the upper bound of h-confidence measure. The effectiveness of HSG is assessed through the experiments with real world datasets.

1 Introduction

Recently, the research area of *hyperclique pattern mining*, that extracts the underlying dependency among objects, attracts a big attention and extensive studies have been reported [25,23,7,15,12]. Among these researches on correlation mining, we focus on the *hyperclique pattern mining* [26,27] in this paper.

While the most of researches aim at finding mutually dependent ‘pairs’ of objects efficiently, a hyperclique pattern is a ‘set’ of highly-correlated items that has high value of an objective measure named *h-confidence* [26,27]. The h-confidence measure of an itemset $P = \{i_1, \dots, i_m\}$ is designed for capturing the strong affinity relationship and is defined as follows.

$$hconf(P) = \min_{l=1, \dots, m} \{conf(i_l \rightarrow P \setminus \{i_l\})\} = sup(P) / \max_{l=1, \dots, m} \{sup(\{i_l\})\}$$

where sup , $conf$, and $hconf$ are the conventional definitions of support and confidence in association rules [1], respectively. A hyperclique pattern P states that the occurrence of an item $i_l \in P$ in a transaction implies the occurrence of all other items $P \setminus \{i_l\}$ in the same transaction with probability at least $hconf(P)$. In addition, the cosine similarity between any pair of items in P is greater than or equals to $hconf(P)$ [27]. By these features, hyperclique pattern discovery has been applied successfully to some real world problems [9,18,24].

While hyperclique pattern discovery aims at finding valuable patterns in transaction databases, *hyperclique pattern mining* is becoming increasingly abundant in many application domains recently. Although we can easily expect to get a more powerful

tool for structured data by introducing correlation mining, the most of current research on correlation mining are designed for transaction databases and little attention is paid to mining correlations from structured data. Motivated by these background, in this paper, we tackle the problem of hyperclique pattern discovery in the context of graph mining [21,22] and discuss the effectiveness of the correlation mining in structured domains.

The basic idea of hyperclique patterns in graph databases is simple: Instead of items, we employ subgraphs (or patterns) as building blocks of hyperclique patterns. While this simple replacement might seem to be trivial, it gives us new expectations and difficulties. On one hand, the proposed framework extracts sets of mutually dependent or affinitive subgraphs in graph databases. Because each pattern gives another view to other patterns in the same set, we can expect to obtain new findings and precise insights. On the other hand, as easily imagined, hyperclique pattern discovery in graph databases is much harder than the traditional tasks because there are exponentially many subgraphs in graph databases and any combinations of those subgraphs are to be potentially candidates. In order to alleviate this combinatorial explosion and to discover hyperclique patterns efficiently, in this paper, we propose a novel algorithm named HSG. HSG reduces the search space effectively by taking into account the structure of hyperclique patterns.

The main contributions of this paper are briefly summarized as follows. First, we formulate the new problem of hyperclique pattern discovery in graph databases. Second, we propose a novel algorithm named HSG for solving this problem efficiently. Third, through the experiments with real world datasets, we assess the effectiveness of our proposal.

This paper is organized as follows. In section 2, after introducing basic notations, we formulate the problem of hyperclique pattern discovery in graph databases. In section 3, the proposed algorithm HSG is explained in detail. After mentioned related work in section 4, we show the results of the experiments in section 5. Finally, we conclude the paper and describe future work in section 6.

2 Preliminaries

Let \mathcal{L} be a finite set of labels. A graph $g = (V_g, E_g, l_g)$ on \mathcal{L} consists of a vertex set V_g , an edge set E_g and a labeling function $l_g : V_g \cup E_g \rightarrow \mathcal{L}$ that maps each vertex or edge to a label in \mathcal{L} . Hereafter, we refer labeled graph as graph simply.

Each graph can be represented in so called *code word* [3,28], that is a unique string which consists of a series of edges associated with connection information. Especially, we employ canonical code word [3,28] which is minimal code word among isomorphic graphs to represent each graph. The *lexicographic order* on code word gives a total order on graphs. Given two graphs g and g' , $g <_{\text{lex}} g'$ denotes that the code word of g is smaller than that of g' . If the code word of g is a prefix of that of g' , we denote it as $g <_{\text{pre}} g'$. Examples of graphs and those code words are shown in Fig 1.

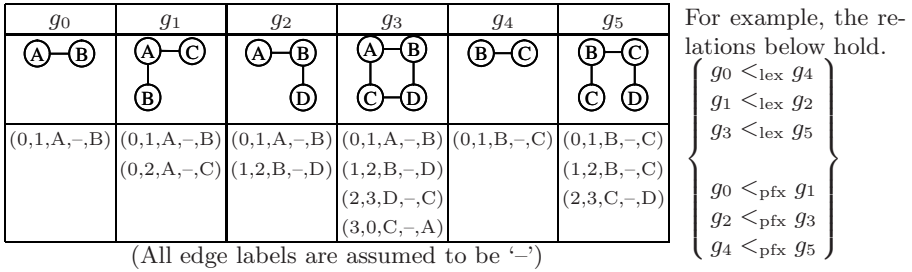


Fig. 1. Examples of Labeled Graphs and those Code Words

A graph $g = (V_g, E_g, l_g)$ is called a *subgraph* of another graph $g' = (V_{g'}, E_{g'}, l_{g'})$, denoted as $g \preceq g'$, if there exists an injective function $f : V_g \rightarrow V_{g'}$ such that $\forall u \in V_g \ l_g(u) = l_{g'}(f(u))$ and $\forall (u, v) \in E_g \ (f(u), f(v)) \in E_{g'} \wedge l_g(u, v) = l_{g'}(f(u), f(v))$. If $g \preceq g'$, then we say that g is a *subgraph* of g' . Note that, if $g <_{\text{pfx}} g'$ holds, then $g \preceq g'$ also holds [3, 28].

Based on the relationship of subgraphs, we consider the *closure* of a set of subgraphs in a graph. The most intuitive definition is as follows: Given a set of subgraphs G and a graph g' , if $\forall g_i \in G \ g_i \preceq g'$ holds, then G is considered as to be occurred in g' . However, this simple definition might not be suitable for the hyperclique patterns of subgraphs because large number of uninteresting combinations of subgraphs having large overlaps in a graph will be obtained. Therefore, we introduce another definition in consideration of *closure* to suppress the redundancy. Given a set of m subgraphs $G = \{g_1, \dots, g_m\}$ and a graph g' , G is called a set of *k-closure* of g' , denoted as $G \leq_k g'$, if there exists the following set of injective functions $\{f_i : V_{g_i} \rightarrow V_{g'} \mid i = 1, \dots, m\}$:

- (1) $\forall g_i \in G \ g_i \preceq g'$
- (2) $\sum_{i=1}^m |E_{g_i}| - |\bigcup_{i=1, \dots, m} \{(f_i(u), f_i(v)) \mid (u, v) \in E_{g_i}\}| \leq k$

The second condition gives the constraint on the edge overlaps. By this constraint, the redundant combinations can be expected to be controlled. For example in Fig. 1, while both $g_1 \preceq g_3$ and $g_2 < g_3$ hold, if k is set to be 0, then $\{g_1, g_2\} \leq_0 g_3$ does not hold because of an overlap of edge ‘A-B’ in g_3 .

We introduce the definitions of *closure* and *k-closure* for a set of subgraphs. Let $D = \{d_1, \dots, d_N\}$ be a database of N graphs. The *closure* and *k-closure* of a set of subgraph $G = \{g_1, \dots, g_m\}$ in D are defined as follows:

$$\begin{aligned} \text{sup}_D(G) &= \sum_{d' \in D} \sigma(G, d') / N \text{ where } \sigma(G, d') = \begin{cases} 1 & (G \leq_k d') \\ 0 & (\text{otherwise}) \end{cases} \\ \text{hconf}_D(G) &= \text{sup}_D(G) / \max_{i=1, \dots, m} \{\text{sup}_D(\{g_i\})\} \end{aligned}$$

Based on the above preparation, we formulate the problem of “mining frequent hyperclique patterns in graph databases” (*mining frequent hyperclique patterns* in short) below. Given a database D of labeled graphs, a positive number called *support threshold* $\sigma \ (0 <$

$\sigma \leq 1$) and a positive number called h_c ($0 \leq h_c \leq 1$), then the problem of HSG mining is to find all G in D such that $sup_D(G) \geq \sigma$, $hconf_D(G) \geq h_c$ and the cardinality of G is more than one. Note that, because we are interested in the sets of mutually dependent subgraphs, the hyperclique patterns of cardinality one are excluded.

3 Mining Hyperclique Patterns of Subgraphs

In this section, we propose an algorithm named HSG for mining frequent hyperclique patterns in graph databases. Before describing the concrete algorithm, we show some properties of hyperclique patterns and a data structure called \mathcal{H} , that are utilized for the effective pruning based on the \mathcal{H} of hyperclique patterns.

3.1 Properties of Hyperclique Patterns

Given two sets G_1 and G_2 of subgraphs, if there exists an injective function $\phi : G_1 \rightarrow G_2$ which satisfies $\forall g \in G_1, g \preceq \phi(g) \in G_2$, then we say that $G_1 \sqsubseteq G_2$ and denote it as $G_1 \sqsubseteq G_2$.

As shown formally below, given a set of subgraphs G_1 , there are two kinds of G_2 to obtain a more specific set of subgraphs G_2 from G_1 . Note that, while only first kind of specialization is considered in item set mining, the second one also plays the key role in HSG mining.

- (1) **Specialization by addition** G_2 is obtained by adding a new subgraph g' to G_1 , $G_2 = G_1 \cup \{g'\}$
- (2) **Specialization by replacement** G_2 is obtained by replacing a subgraph $g \in G_1$ to a more specific subgraph $g' (\succeq g)$, $G_2 = (G_1 \setminus \{g\}) \cup \{g'\}$.

The following two lemmas hold in hyperclique patterns of subgraphs based on the generality ordering introduced above.

Lemma 1 (Anti-monotone property of support value). $G_1 \sqsubseteq G_2 \implies sup_D(G_1) \geq sup_D(G_2)$

Obvious from the definition of support value. □

By this lemma, if a set of subgraphs G_1 does not satisfy the minimum support, then all sets of subgraphs $G_2, G_1 \sqsubseteq G_2$ can be eliminated safely from the candidate of frequent hyperclique patterns.

Lemma 2 (Upper bound of h-confidence). $G_1 = G_A \cup G_B, G_A \neq \emptyset, G_A \cap G_B = \emptyset, G_2 = G_A \cup G'_B, G_A \cap G'_B = \emptyset, G_B \sqsubseteq G'_B$

$$up(G_1, G_A) = sup_D(G_1) / \max_{g \in G_A} \{sup_D(\{g\})\} \geq hconf_D(G_2)$$

Since $G_A \subseteq G_2$, $\max_{g \in G_A} \{sup_D(\{g\})\} \leq \max_{g' \in G_2} \{sup_D(\{g'\})\}$ holds. By lemma 1, $sup_D(G_1) \geq sup_D(G_2)$ also holds. Therefore, $sup_D(G_1) / \max_{g \in G_A} \{sup_D(\{g\})\} \geq sup_D(G_2) / \max_{g' \in G_2} \{sup_D(\{g'\})\} = hconf_D(G_2)$ holds. \square

This lemma gives the upper bound of h-confidence. If $up(G_1, G_A)$ does not satisfy the minimum h-confidence h_c , then any set of subgraphs $G_2 = G_A \cup G'_B$, $G'_B \supseteq G_B$ must not satisfy h_c . Furthermore, this lemma also shows the anti-monotone property of h-confidence with respect to the specialization by addition. By definition, $hconf_D(G_1) = up(G_1, G_1)$ holds. Thus, if $hconf_D(G_1) < h_c$, then no set of subgraphs obtained by adding some subgraphs to G_1 can satisfy h_c .

3.2 A Conditional Prefix Tree of Hyperclique Patterns

Here, we consider the enumeration of hyperclique patterns in graph databases.

According to the reverse search 2, the repeated enumeration of the same pattern can be avoided by generating each pattern from its unique parent. In case of hyperclique patterns of subgraphs, the parent can be uniquely defined by using the total order of graphs formed by code word. The parent of a set of subgraphs G , denoted as $p(G)$, is a set obtained by removing the smallest element with respect to $<_{lex}$ from G , $p(G) = G \setminus \{g \in G \mid \nexists g' \in G \ g' <_{lex} g\}$.

Because of the anti-monotone property of hyperclique patterns with respect to the specialization by addition shown in lemma 1 and 2, all subsets of a frequent hyperclique pattern must be also frequent hyperclique patterns. Furthermore, a hyperclique pattern should be enumerated via its parent to avoid the repeated enumerations. Therefore, in our strategy, a new hyperclique pattern G' will be generated by joining two hyperclique patterns $G_1 = G \cup \{g_1\}$ and $G_2 = G \cup \{g_2\}$ as $G' = G \cup \{g_1\} \cup \{g_2\} = G_1 \cup \{g_2\}$. Note that “the enumeration via parent” can be naturally realized through the join operation.

Since a hyperclique pattern will be generated by joining two hyperclique patterns having the same parent, it is convenient to treat all hyperclique patterns which have the same parent as an unit. Furthermore, in order to effectively utilize the pruning based on the generality ordering, hyperclique patterns in this unit should be organized in consideration of the generality ordering. Motivated by these backgrounds, we propose a tree-shaped data structure called CPT_G , on which our algorithm HSG works, for storing hyperclique patterns which have the same parent in common.

A conditional prefix tree of hyperclique patterns $CPT_G = (V_G, E_G, B_G, root)$ is an ordered tree and it stores hyperclique patterns which have a hyperclique pattern G as those parent. The root node $root$ is a dummy node. Each node v in V_G , except for $root$, corresponds to a hyperclique pattern $G \cup \{g(v)\}$ and has an graph $g(v)$. $E_G \subseteq V_G \times V_G$ and $B_G \subseteq V_G \times V_G$ represent the set of parent-child and sibling relationships, respectively. These are formally defined as follows.

$$\begin{aligned}
 E_G &= \{(v_1, v_2) \mid g(v_1) <_{pfx} g(v_2), \nexists v' \in V_G [g(v_1) <_{pfx} g(v') \wedge g(v') <_{pfx} g(v_2)]\} \\
 &\quad \cup \{(root, v_3) \mid \nexists v' \in V_G [g(v') <_{pfx} g(v_3)]\} \\
 B_G &= \{(v_1, v_2) \mid g(v_1) <_{lex} g(v_2), \exists v' \in V_G [(v', v_1) \in E_G \wedge (v', v_2) \in E_G]\}
 \end{aligned}$$

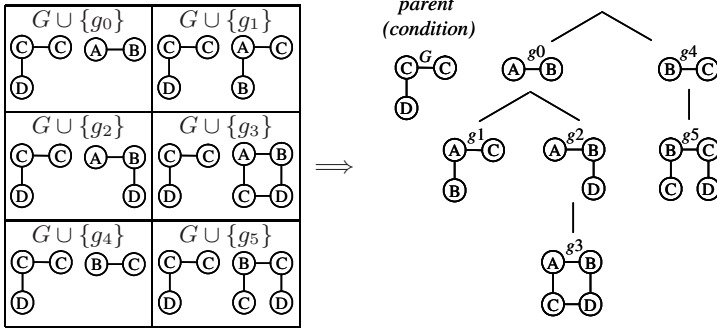


Fig. 2. An Example of Conditional Prefix Tree

Intuitively speaking, v_1 is the parent of v_2 if the code word of $g(v_1)$ is the longest prefix of that of $g(v_2)$. If v_3 has no such parent, then *root* is assigned as the parent of v_3 . Note that, $\forall (g_1, g_2) \in E_G g_1 \preceq g_2$ holds. The children of a node are ordered in the lexicographic order $<_{lex}$. An example of conditional prefix tree is shown in Fig. 2. This tree is constructed from six hyperclique patterns that have $\{G\}$ as parent in common.

3.3 HSG: A Hyperclique Pattern Miner in Graph Databases

In this subsection, we propose an algorithm HSG and explain it in detail.

The algorithm HSG for mining frequent hyperclique patterns in graph databases is shown in Fig. 3. In the following explanation, we use the notations below for the sake of simplicity: $G_x = G \cup \{g(g_x)\}$, $G_{x'} = G \cup \{g(g'_x)\}$ and $G_{x,y} = G \cup \{g(g_x), g(g_y)\}$ where we assume $g(g_x) < g(g'_x)$.

As an input, HSG takes an unconditional prefix tree CPT_\emptyset of hyperclique patterns that stores frequent hyperclique patterns of cardinality one, frequent subgraphs potentially obtained by the conventional graph miners [28, 11, 10, 16]. Then, HSG calls a procedure LoopV with $T_a = T_b = CPT_\emptyset$ (line1 in HSG).

HSG consists of two main procedures LoopV and LoopH which realize the join of elements in a conditional prefix tree mutually while considering the generality ordering. LoopV traverses a tree T_a in preorder by using recursive call (line5 in LoopV). By using the preorder traversal, elements in T_a will be considered in the order of $<_{lex}$. During the traversal, LoopV calls LoopH with G , g_a and T_b (line3 in LoopV). LoopH also traverses a tree T_b in preorder (line16 in LoopH). Since T_a and T_b refer to the same tree at the beginning, if no pruning is applied, all pairs of elements in a conditional prefix tree will be considered. Note that, no repeated enumeration occurs due to the check of $g(g_a) \leq_{lex} g(g_b)$ (line2 in LoopH).

During the recursive calls, LoopH constructs two new conditional prefix trees NT_a and NT_b which form the search spaces afterwards. NT_a is a prefix tree under the condition G_a and it is used as an input for discovering hyperclique patterns whose parent is $G_{a,b}$ (line4 in LoopV). NT_a will be constructed by adding a new hyperclique pattern $G_{a,b}$ whenever it is obtained (line10 in LoopH). NT_b is a

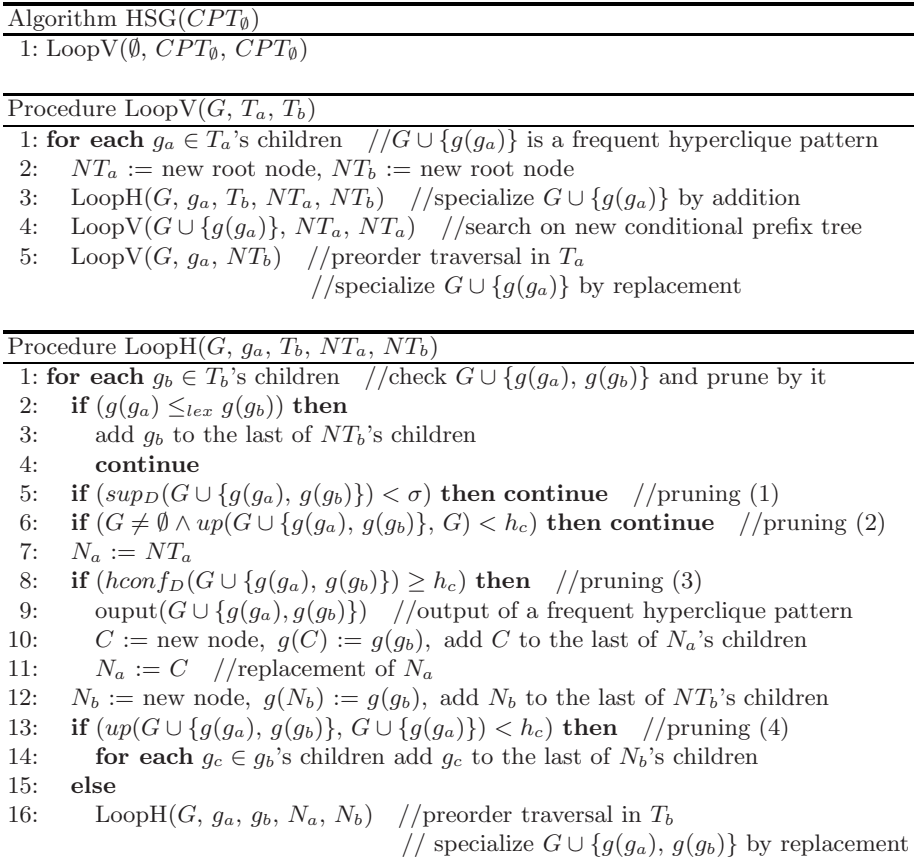


Fig. 3. An algorithm HSG of mining hyperclique patterns in graph databases

prefix tree under the condition G , on which hyperclique patterns having $G_{a'}$ as parents will be mined (line5 in LoopV). Conceptually, NT_b will be obtained by pruning some branches in T_b .

Four prunings will be applied in LoopH. They are achieved partially by “not adding new vertices to NT_a and NT_b ”. The first pruning is based on the anti-monotone property of support value in lemma 1 (line5 in LoopH). If the support of $G_{a,b}$ is less than the minimum support, then all patterns which are more specific than $G_{a,b}$ must not satisfy the minimum support. Thus, we ignore the following specializations of $G_{a,b}$ by skipping the loop of line1 in LoopH: (1) $G_{a,b'}$ by not calling LoopH (line16 in LoopH), (2) patterns obtained by “specialization of $G_{a,b}$ by addition” by not updating NT_a , and (3) $G_{a',b}$ and $G_{a',b'}$ by not updating NT_b . The second pruning is derived from the upper bound of h-confidence in lemma 2 (line6 in LoopH). As similar to the first pruning, all specializations of $G_{a,b}$ will be ignored in the same way. The third pruning is by anti-monotone property of h-confidence with respect to the specialization by addition in lemma 2

(line8 in LoopH). If $G_{a,b}$ dose not satisfy minimum h-confidence, the search for patterns having $G_{a,b}$ as parent will be avoided by not adding $G_{a,b}$ to NT_a . The fourth pruning is based on the upper bound of h-confidence in lemma2 (line13 in LoopH). The search for $G_{a,b'}$ can be avoided by not calling LoopH. Note that, $G_{a',b}$ as well as $G_{a',b'}$ must be considered. Therefore, NT_b has to be updated. This is achieved through the update of N_b .

As shown above, HSG makes the best use of the pruning based on the specializations by using the conditional prefix trees. For HSG, the following theorem holds.

Theorem 1. *Let G be a graph and f be a frequent hyperclique pattern. The set of frequent hyperclique patterns containing f can be derived from the complete enumeration procedure by the double preorder traversals and the safety prunings guaranteed by lemma1 and 2. □*

Although HSG can discover all frequent hyperclique patterns, the obtained set of hyperclique patterns may contain some redundancy. Since each frequent subgraph in the unconditional prefix tree is treated as an item, if some subgraphs which are equivalent in some senses are contained in the tree, they cause the redundancy. To eliminate obviously redundant patterns, we believe that the frequent subgraphs included in the unconditional prefix tree should be limited to the representatives such as closed subgraphs (a graph g_c , $\exists g' g_c \preceq g' \wedge sup_D(g_c) = sup_D(g')$) or minimal subgraphs (a graph g_m , $\exists g' g' \preceq g_m \wedge sup_D(g_m) = sup_D(g')$). In particular, minimal subgraphs might be more suitable if the $sup_D(g)$ is considered in the joint occurrence. Although, to the best of our knowledge, the method which finds minimal subgraphs directly has not been proposed yet, those subgraphs can be obtained by some post-processing of the conventional graph miners [28,11,10,16].

4 Related Work

The concept of HSG mining is inspired by the hyperclique pattern discovery in transaction databases [26,27].

The methods of mining correlated pairs of items have been proposed [25,23,7]. Furthermore, correlated pattern mining based on a pattern-growth methodology in transaction databases has been proposed [15]. Compared with these methods, HSG is different in the point of finding sets of affinitive structured patterns.

On the correlation mining in graph databases, a new problem named *graph correlation mining* has been proposed recently [12]. In this problem, Pearson's correlation coefficient [20] is employed as correlation measure and all correlated subgraphs with a query graph will be discovered. This framework is greatly different from our proposal because the different measure is employed and only subgraphs correlated with a query are considered.

The *graph pattern mining* proposed in [13] is a set of patterns that optimizes some quality measure. The discovery of pattern team may look similar to the HSG mining

Table 1. Statistics of Datasets

	$ D $	V_a	E_a	$ V $	$ E $	Description
D_1	1000	11.6	20.5	20	20	A synthetic dataset generated by graph generator [5]
PTE	340	27.0	27.4	66	4	The Predictive Toxicology Evaluation Challenge [8]
DTP_CM	877	29.1	31.5	12	4	The DTP AIDS Antiviral Screen dataset [4]

$|D|$: # of graphs in datasets. V_a, E_a : average number of vertices and edges per graph. $|V|, |E|$: # of distinct labels of vertices and edges.

because both find the set or combination of patterns. However, pattern team discovery is done by selecting patterns from the given set. In addition, pattern team usually consists of a set of mutually dissimilar and independent patterns for optimizing the quality measure. Similar to the pattern team in some senses, the concept of α -orthogonal patterns in graph databases has been proposed recently [6]. In this framework, a set of frequent maximal subgraphs that are mutually dissimilar with each other will be obtained by employing a randomized search. While treating a set of subgraphs, this framework is also different from the HSG mining because HSG discovers the complete sets of affinitive subgraphs.

From the aspect of finding similar patterns, α -orthogonal patterns [19, 17, 29] is closely related to the HSG mining. In redescription mining, patterns consist of any combinations of conjunction, disjunction and negation of items and pairs of patterns that occur in almost the same transactions will be discovered. While this framework is very general, neither the application to the structured data nor precise algorithms which use the generality ordering have been proposed yet.

5 Experimental Evaluation

To assess the effectiveness of the proposed algorithm, we implement HSG in Java and conduct some experiments with the datasets shown in Table 1 on a PC (CPU: Intel(R) Core2Quad 2.4GHz) with 4Gbytes of main memory running Windows XP. Furthermore, another miner pHSG, that is ‘‘HSG without pruning (2) and (4)’’, is also prepared to demonstrate the effects of pruning related to the ‘‘specialization by replacement’’. In the experiments, we construct the unconditional prefix trees CPT_θ by using minimal subgraphs only. Experimental results are shown in Table 2.

The obtained number of hyperclique patterns decreases when the value of k is reduced. Furthermore, though not shown in Table 2, about 231 million and 17 thousand of hyperclique patterns were obtained if we set $\sigma = 0.1, h_c = 0.9$ and $k = \infty$ in HSG and pHSG , respectively. This means that the consideration of edge-disjointness succeeds in suppressing the generation of redundant patterns.

In all cases, pHSG discovers all frequent hyperclique patterns in a reasonable time though at least $\mathcal{O}(|CPT_\theta|^2)$ of candidates will be generated if no pruning applied. Thus, it is understood that the pruning by minimum support is effective enough. Note that, this pruning eliminates the patterns obtained by the ‘‘specialization by addition’’ as well as the ‘‘specialization by replacement’’. Compared

Table 2. Experimental Results

h_c	k	P	Time	Cand.	P	Time	Cand.
Results for D_1							
				$\sigma = 0.025$ ($ CPT_\emptyset = 1208$)			
				$\sigma = 0.01$ ($ CPT_\emptyset = 8946$)			
0.8	0	0	0.3 (0.6)	17.4 (32.6)	0	1.3 (4.4)	102.5 (337.9)
	1	2	0.4 (0.6)	22.5 (38.2)	4	2.0 (6.1)	155.9 (470.6)
	∞	7	0.3 (0.5)	22.7 (38.3)	756	2.1 (5.7)	191.3 (513.1)
0.7	0	0	0.3 (0.6)	18.0 (32.6)	0	1.4 (4.4)	109.6 (337.9)
	1	45	0.4 (0.6)	23.2 (38.2)	64	2.2 (6.1)	174.5 (470.6)
	∞	81	0.3 (0.5)	23.4 (38.4)	3066	2.3 (5.7)	213.7 (514.3)
Results for PTE							
				$\sigma = 0.1$ ($ CPT_\emptyset = 561$)			
				$\sigma = 0.05$ ($ CPT_\emptyset = 1441$)			
0.9	0	16	0.3 (1.1)	2.3 (8.3)	154	1.7 (8.0)	9.6 (48.0)
	1	93	0.6 (2.0)	4.2 (13.9)	565	3.0 (13.2)	16.2 (67.0)
0.8	0	85	0.9 (2.1)	2.9 (8.4)	821	3.2 (9.4)	13.8 (49.9)
	1	524	3.4 (7.0)	5.9 (14.8)	3815	16.5 (28.1)	29.5 (77.2)
0.7	0	165	1.3 (4.0)	3.5 (9.1)	1165	5.5 (11.8)	17.0 (51.4)
	1	1228	9.0 (35.5)	7.9 (17.1)	6000	45.4 (77.2)	38.5 (85.4)
Results for DTP_{CM}							
				$\sigma = 0.1$ ($ CPT_\emptyset = 417$)			
				$\sigma = 0.05$ ($ CPT_\emptyset = 1592$)			
0.9	0	9	0.5 (2.4)	2.2 (11.3)	10	1.2 (7.7)	9.9 (62.6)
	1	48	0.7 (2.8)	3.1 (13.9)	70	1.6 (9.0)	15.8 (79.0)
0.8	0	32	1.0 (2.7)	3.3 (11.3)	40	2.2 (8.0)	14.7 (62.7)
	1	109	1.4 (3.4)	4.4 (14.0)	242	2.9 (9.7)	22.7 (79.3)
0.7	0	110	2.6 (8.8)	4.3 (11.6)	129	4.1 (14.1)	19.0 (63.0)
	1	371	48.3 (116.6)	5.7 (14.7)	628	50.3 (123.1)	28.5 (80.2)

k : # of the edge overlaps permitted in the joint occurrence (∞ means no restriction).
 P : # of obtained hyperclique patterns. Time: execution time after CPT_\emptyset is given (in second). Cand.: # of candidates enumerated during the search (in thousand). Numbers in parentheses in Time and Cand. are for pHSG.

with pHSG, the execution time of HSG for real world problems decreases to 16.0% in the maximum and to 33.9% on the average. The number of candidate patterns is also reduced to 15.9% in the maximum and to 30.8% on the average. It is also observed that HSG runs about two times faster than pHSG in the synthetic dataset on the average. These reductions are the strong evidences to show the effectiveness of the pruning based on the generality ordering, especially on the “specialization by replacement”.

6 Conclusion

In this paper, we formulate the problem of hyperclique pattern discovery in graph databases. To solve this problem efficiently, a novel algorithm named HSG is proposed that utilizes the depth-first/breadth-first search with the effective pruning based on the generality ordering. We believe that HSG can mine hyperclique

patterns efficiently not only in other types of structured data but also in transaction databases with the conceptual hierarchy because the conditional prefix trees, on which HSG works, can be constructed naturally from these kinds of datasets.

For future work, the theoretical analysis of the proposed algorithm and further experiments with large-scale datasets are necessary. In addition, some more efficient mechanism is required for computing support value of a set of edge disjoint subgraphs. For this objective, we plan to employ the idea of support value computation of edge disjoint subgraphs in a large graph [14]. We also plan to apply the proposed algorithm to top-k correlated pattern discovery as well as to redescription mining in structured databases.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. of 20th International Conference on Very Large Data Bases (VLDB 1994), pp. 487–499 (1994)
2. Avis, D., Fukuda, K.: Reverse search for enumeration. *Discrete Applied Mathematics* 65(1-3), 21–46 (1996)
3. Borgelt, C.: On canonical forms for frequent graph mining. In: Working Notes of the 3rd International ECML/PKDD- Workshop on Mining Graphs, Trees and Sequences (MGTS 2005), pp. 1–12 (2005)
4. Borgelt, C., Berthold, M.R.: Mining molecular fragments: Finding relevant substructures of molecules. In: Proc. of the 2002 IEEE International Conference on Data Mining (ICDM 2002), pp. 51–58 (2002)
5. Cheng, J., Ke, Y., Ng, W.: Graphgen: A graph synthetic generator (2006), <http://www.cse.ust.hk/graphgen/>
6. Hasan, M., Chaoji, V., Salem, S., Besson, J., Zaki, M.: ORIGAMI: Mining representative orthogonal graph patterns. In: Proc. of the 7th IEEE International Conference on Data Mining (2007)
7. He, Z., Xu, X., Deng, S.: Mining top-k strongly correlated item pairs without minimum correlation threshold. *International Journal of Knowledge-based and Intelligent Engineering Systems* 10(2), 105–112 (2006)
8. Helma, C., King, R.D., Kramer, S., Srinivasan, A.: The predictive toxicology challenge 2000-2001. *Bioinformatics* 17(1), 107–108 (2001)
9. Hu, T., Xiong, H., Sung, S.Y.: Co-preserving patterns in bipartite partitioning for topic identification. In: Proc. of the 7th SIAM International Conference on Data Mining, pp. 509–514 (2007)
10. Huan, J., Wang, W., Prins, J.: Efficient mining of frequent subgraphs in the presence of isomorphism. In: Proc. of the 3rd IEEE International Conference on Data Mining, pp. 549–552 (2003)
11. Inokuchi, A., Washio, T., Motoda, H.: Complete mining of frequent patterns from graphs: Mining graph data. *Machine Learning* 50, 321–354 (2003)
12. Ke, Y., Cheng, J., Ng, W.: Correlation search in graph databases. In: Proc. of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2007), pp. 390–399 (2007)
13. Knobbe, A.J., Ho, E.K.Y.: Pattern teams. In: Fürnkranz, J., Scheffer, T., Spiliopoulou, M. (eds.) PKDD 2006. LNCS (LNAI), vol. 4213, pp. 577–584. Springer, Heidelberg (2006)

14. Kuramochi, M., Karypis, G.: Finding Frequent Patterns in a Large Sparse Graph. *Data Mining and Knowledge Discovery* 11(3), 213–321 (2005)
15. Lee, Y.-K., Kim, W.-Y., Cai, Y.D., Han, J.: Comine: Efficient mining of correlated patterns. In: *Proc. of the 3rd IEEE International Conference on Data Mining*, pp. 581–584 (2003)
16. Nijssen, S., Kok, J.: A quickstart in frequent structure mining can make a difference. In: *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, pp. 647–652 (2004)
17. Parida, L., Ramakrishnan, N.: Redescription mining: Structure theory and algorithms. In: *Proc. of the 20th National Conference on Artificial Intelligence and the 17th Innovative Applications of Artificial Intelligence Conference*, pp. 837–844 (2005)
18. Qian, T., Xiong, H., Wang, Y., Chen, E.: On the strength of hyperclique patterns for text categorization. *Information Sciences* 177(19), 4040–4058 (2007)
19. Ramakrishnan, N., Kumar, D., Mishra, B., Potts, M., Helm, R.F.: Turning cartwheels: an alternating algorithm for mining redescrptions. In: *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 266–275 (2004)
20. Tan, P.-N., Kumar, V., Srivastava, J.: Selecting the right interestingness measure for association patterns. In: *Proc. of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 32–41. ACM Press, New York (2002)
21. Washio, T., Motoda, H.: State of the art of graph-based data mining. *SIGKDD Explorations* 5(1), 59–68 (2003)
22. Washio, T., Kok, J.N., De Raedt, L. (eds.): *Advances in Mining Graphs, Trees and Sequences*. IOS Press, Amsterdam (2005)
23. Xiong, H., Brodie, M., Ma, S.: Top-cop: Mining top-k strongly correlated pairs in large databases. In: *Proc. of the 6th International Conference on Data Mining*, pp. 1162–1166 (2006)
24. Xiong, H., He, X., Ding, C., Zhang, Y., Kumar, V., Holbrook, S.R.: Identification of functional modules in protein complexes via hyperclique pattern discovery. In: *Proc. of the Pacific Symposium on Biocomputing*, pp. 221–232 (2005)
25. Xiong, H., Shekhar, S., Tan, P.-N., Kumar, V.: Exploiting a support-based upper bound of pearson's correlation coefficient for efficiently identifying strongly correlated pairs. In: *Proc. of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 334–343. ACM Press, New York (2004)
26. Xiong, H., Tan, P.-N., Kumar, V.: Mining strong affinity association patterns in data sets with skewed support distribution. In: *Proc. of the 3rd IEEE International Conference on Data Mining (ICDM 2003)*, pp. 387–394 (2003)
27. Xiong, H., Tan, P.-N., Kumar, V.: Hyperclique pattern discovery. *Data Mining and Knowledge Discovery* 13(2), 219–242 (2006)
28. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: *Proc. of the 2002 IEEE International Conference on Data Mining (ICDM 2002)*, pp. 721–724 (2002)
29. Zaki, M.J., Ramakrishnan, N.: Reasoning about sets using redescription mining. In: *Proceeding of the 11th ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 364–373 (2005)

A Minimal Description Length Scheme for Polynomial Regression

Aleksandar Pečkov, Sašo Džeroski, and Ljupčo Todorovski

Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

Abstract. The paper addresses the task of polynomial regression, i.e., the task of inducing polynomials from numeric data that can be used to predict the value of a selected numeric variable. As in other learning tasks, we face the problem of finding an optimal trade-off between the complexity of the induced model and its predictive error. One of the approaches to finding this optimal trade-off is the minimal description length (MDL) principle. In this paper, we propose an MDL scheme for polynomial regression, which includes coding schemes for polynomials and the errors they make on data. We empirically compare this principled MDL scheme to an ad-hoc MDL scheme and show that it performs better. The improvements in performance are such that the polynomial regression approach we propose is now comparable in performance to other commonly used methods for regression, such as model trees.

Keywords: regression, polynomial regression, minimal description length.

1 Introduction

Regression models are used to predict the value of a dependent numeric variable from the values of independent (predictor) variables. Commonly used regression methods include linear regression and regression trees [1]. While linear regression tries to find a global model of the data (a linear equation), regression tree induction finds piecewise models that partition the data space into a number of sub-spaces and induce use a constant or a linear model in each of them. While linear models tend to be oversimplistic, regression trees can sometimes overfit the data. In this paper, we address the task of polynomial regression, i.e., the task of inducing polynomial equations that can be used to predict the value of a numeric variable. Polynomials can also overfit the data. Namely, it is well known that a data set of n points can be perfectly interpolated (and often overfitted) with a polynomial of an $(n - 1)$ -th degree.

In order to address the problem of overfitting, different approaches to model selection have been proposed in the literature [3] (pages 193-222). Each approach tries to find an optimal trade-off between the complexity of the induced model and its predictive error and thus avoid overfitting. The minimal description length (MDL) principle is one such approach. Following the MDL principle, the quality of a model is estimated by combining the model complexity and the

predictive error that the model makes on the training data. The complexity of the model and the error are measured in terms of the number of bits necessary for encoding them. Therefore, MDL based measures of model quality heavily depend on the encoding scheme chosen. Different encoding schemes have been proposed for linear equations and regression trees [7], but to our knowledge no encoding scheme has been proposed for multivariate polynomials.

The aim of this paper is to identify an appropriate MDL scheme for polynomial regression. We consider two MDL schemes. The first one is an ad-hoc scheme proposed in [8]. The second is a novel scheme proposed by this paper and is based on the refined MDL principle [2]. The MDL schemes are implemented and used in the context of CIPER [8], a machine learning system for finding polynomial equations from numeric data. CIPER algorithm performs a heuristic beam search through the space of equations, proceeding from simple to more complex equations by using a refinement operator that at each step increases the equation complexity by one. In this paper we also introduce a new refinement operator in CIPER which can increase the complexity of the equation by more than one, combined with a simplification step.

We perform an empirical evaluation on several standard regression datasets from the UCI repository [4]. Within CIPER, we compare the old and the new refinement operator, as well as the two MDL schemes to each other and to linear regression, regression trees and model trees.

The paper is organized as follows. In Section 2, we introduce the task of polynomial regression and outline CIPER, a method for inducing polynomials based on heuristic search through the space of candidate polynomials. This section also includes description of the refinement operator used in CIPER and our proposal for a new refinement operator. Section 3 presents the two MDL schemes compared in the paper. Section 4 presents and discusses the results of the empirical evaluation of the MDL schemes. Finally, Section 5 concludes the paper, discusses related work, and proposes directions for further research.

2 Polynomial Regression

The task of polynomial regression is defined as follows: given numeric data, induce a polynomial equation that can predict the value of a target variable.

A polynomial over variables x_1, x_2, \dots, x_n can be written in the form:

$$P = C_0 + C_i \cdot \sum_{i=1}^m T_i$$

where $T_i = \prod_{j=1}^n x_j^{a_{i,j}}$, $C_i, i = 1..m$ and C_0 are constants, and $C_i \neq 0$. We say T_i is a \dots or \dots in P . The length of P is $\dots(P) = \sum_{i=1}^m \sum_{j=1}^n a_{i,j}$, the size of P is $\dots(P) = m$; and the degree of P is $\dots(P) = \dots \sum_{i=1}^m \sum_{j=1}^n a_{i,j}$.

An example polynomial equation is $P = 1.2x^2y + 3.5xy^3 + 5xz + 2$. This equation has size 3, degree 4 and length 9.

2.1 CIPER

CIPER [8] (Constrained Induction of Polynomial Equations for Regression) is a machine learning algorithm for finding polynomial equations. It uses beam search to heuristically search through the space of possible equations for ones that fit the data best.

The top-level outline of the CIPER algorithm is shown in Table 1. First, the beam is initialized either with the simplest polynomial equation $P = C$, or with a user specified minimal polynomial. In every search iteration, a set of polynomials is generated from the polynomials in the beam by using a refinement operator. The coefficients before the terms in a polynomial are fitted by using the method of least squares. For each of the generated polynomials, the value of the minimal description length (MDL) heuristics is calculated. At the end of the iteration, the equations with smallest MDL values are retained in the beam. The evaluation stops when the refinement operator can not generate new equations or when the content of the beam is unchanged in the last iteration. Such a situation occurs when every polynomial generated in the last iteration has a worse MDL estimate than the polynomials already in the beam.

We have introduced some optimizations for fitting the coefficients of the generated polynomial structure. The data is represented as a matrix M , where the number of rows is the number of instances, and the number of columns is the number of terms plus one (the first column is filled with ones). The least squares estimate for the coefficients C_i of the equation is

$$C = (M^T \cdot M)^{-1} \cdot (M^T \cdot y)$$

where y is the vector of values we are trying to predict. In this equation the multiplication is computationally expensive because of the large number of rows. Let T_1 and T_2 be terms in equation A , T_3 and T_4 terms in equation B , such that $T_1 \cdot T_2 = T_3 \cdot T_4$. Then the appropriate elements in the matrices $M_A^T \cdot M_A$ and $M_B^T \cdot M_B$ are equal. We store all generated elements from the matrices

Table 1. A top-level outline of the CIPER algorithm. Q is the set of best b equations and Q_r is the set of refined equations.

```

procedure CIPER(Data, InitialPolynomial)
  InitialPol = FITPARAMETERS(InitialPolynomial, Data)
   $Q = \{InitialPolynomial\}$ 
  repeat
     $Q_r =$  refinements of equation structures in  $Q$ 
    foreach equation structure  $E \in Q_r$  do
       $E =$  FITPARAMETERS( $E$ , Data)
    endfor
     $Q = \{\text{best } b \text{ equations from } Q \cup Q_r\}$ 
  until  $Q$  unchanged during the last iteration
  print  $Q$ 

```

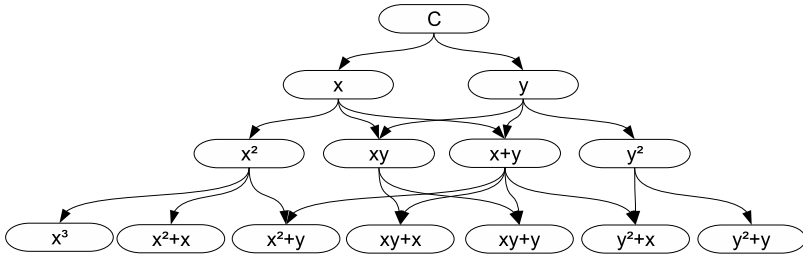


Fig. 1. A lattice of polynomial equations generated by the original CIPER refinement operator. Equation length is increased by one in each refinement step.

$M^T \cdot M$. We reuse them for calculating the matrices of the subsequently generated polynomials. This optimization considerably lowers the computational cost at the expense of some memory.

2.2 The CIPER Refinement Operator

A refinement operator is a function that takes as input an equations structure and generates a new equation structure by modifying the old one. The original CIPER refinement operator increases the length of an equation by one, either by adding a first degree term or by multiplying an existing term with a variable (Figure 1). Starting with the simplest equation, and iteratively applying this refinement operator, all polynomial equations can be generated.

Given an expression $x + y$, we can refine it in two ways. First, we can include a new linear term yielding $x + y + z$. Second we can replace an existing term in the expression (e.g. x) by multiplying it with a variable (e.g. z), yielding a new expression (e.g. $xz + y$).

2.3 The New Refinement Operator

Adding a term to a linear (in the parameters) equation always decreases its error (at least on training data). However, replacing a term with a more complex version thereof (multiplied by a variable) doesn't necessarily decrease the error of the equation. If we add z to $x + y$, yielding $x + y + z$, we will reduce the error of the equation. However, if we replace x with xz , yielding $xz + y$, xz need not be strongly correlated with x and the replacement might actually increase the error of the equation.

This has motivated us to modify the refinement operator in CIPER. Besides the two types of refinement considered in the original version of CIPER, we introduce a third one. We take a term in the equation, make a copy thereof, multiply the copy with a new variable and add the product back to the equation. For example, $x + y$ can be refined to $x + y + xz$ by the new operator.

The old refinement operator always increases the complexity of an equation by one. The new refinement operator can increase the complexity of an equation

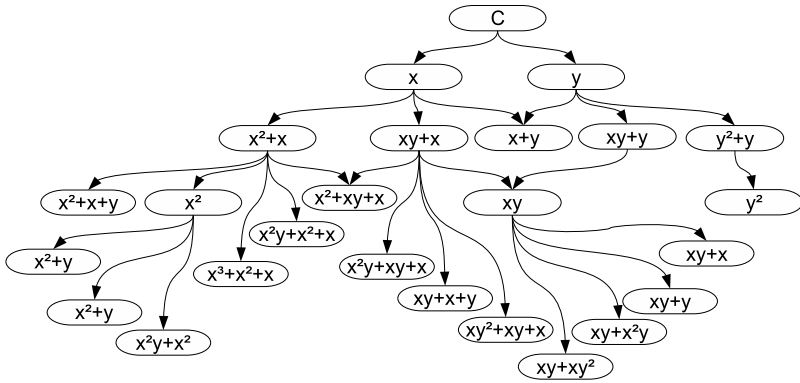


Fig. 2. The improved CIPER refinement operator. The length of the equation can increase by more than one.

considerably. Because of this, we introduce an extra simplification step in CIPER. For every equation added to the beam, we try removing each of its terms: if this yields an equation with better heuristic value, we add the newly formed equation to the beam.

We will refer to CIPER using the new refinement operator as CIPER-R.

We came to the part of identifying the best equations in the beam. For this we need an objective measure that will combine complexity and the error.

3 Minimal Description Length Heuristics for Polynomial Models

We will give two alternatives for measuring the complexity of the model. The first is an ad-hoc solution, used in the first version of the CIPER algorithm [8]. The second is based on theoretical results in MDL theory. We present the encoding and the associated complexity measure.

3.1 Ad-Hoc MDL Heuristic

The original CIPER implementation used an ad-hoc MDL heuristics, defined as follows

$$f(P) = \lambda_l(P) \cdot \lambda_e(m) + m \cdot \lambda_e(\sigma^2(P))$$

where P is the polynomial equation being evaluated, $\lambda_l(P)$ is its length, $\sigma^2(P)$ is its mean squared error, and m is the number of training examples.

This evaluation function is based on the Akaike and Bayesian information criteria for regression model selection [3]. The second term of the ad-hoc MDL heuristic function measures the degree of fit of a given equation and the first term introduces a penalty for the complexity of the equation. With this penalty, the MDL heuristic function introduces a preference toward simpler equations.

3.2 Improved MDL Heuristic

Following the minimal description length (MDL) principle, among a number of candidate models, we select the one that represents a good trade-off between the model's predictive error and its complexity. The MDL principle combines two ideas (or assumptions) about the relation between learning and data compression:

- regularities in the data can be used to compress the data, i.e., the more regularities there are, the more the data can be compressed;
- the more we are able to compress the data, the more we have learned about the data.

Thus, the complexity of a model can be estimated as its ability to compress data: the larger the compression, the smaller the complexity of the obtained model. More specifically, an MDL estimate of model quality is composed of two components:

$$L(H) = L(H) + L(D|H),$$

where the first component $L(H)$ corresponds to the length of the encoding of the model (hypothesis) H , while the second one $L(D|H)$ is the length of the description of the data when encoded using the model H .

3.3 Encoding Polynomial Structure

In order to encode the structure of a polynomial, we follow the refined MDL approach [2]. We first partition the space of candidate models into subgroups \mathcal{H}_c of models with equal complexity c . A particular model $H \in \mathcal{H}_c$ can be then encoded using $N = \lceil \log_2 |\mathcal{H}_c| \rceil$ bits, where $|\mathcal{H}_c|$ denotes the number of models in the class \mathcal{H}_c .

In the case of polynomials, we partition the space of candidate polynomial structures in to classes at several levels. At the highest level, we group together the candidate polynomials with the same length l and the same number of terms (size) m . We refer to these classes as $G(m, l)$; for example $G(1, 1)$ contains polynomial structures with one linear term, while $G(1, 2)$ contains polynomial structures with only one term of second degree. Note that $m \leq l$. At the second level, we partition each $G(m, l)$ in subgroups with fixed term degrees $G'(a_1, a_2, \dots, a_m)$. All polynomials in this subgroup have m terms with degrees $a_1 \geq a_2 \geq \dots \geq a_m$. Note that $\sum_{i=1}^m a_i = l$. Now we have to calculate how many sub-groups G' there are in a single $G(m, l)$ group and also calculate how many polynomial structures there are in each $G(a_1, a_2, \dots, a_m)$ group.

The number $|G'(a_1, a_2, \dots, a_m)|$ can be easily calculated using a procedure roughly depicted in Figure 3. Given the degree of the first term a_1 we have to choose a_1 variables from the set $\{x_1, x_2, \dots, x_n\}$, where variables can appear in the selection more than once. Thus, the number of possibilities for the first term equals the number of combinations with repetition where we select a_1

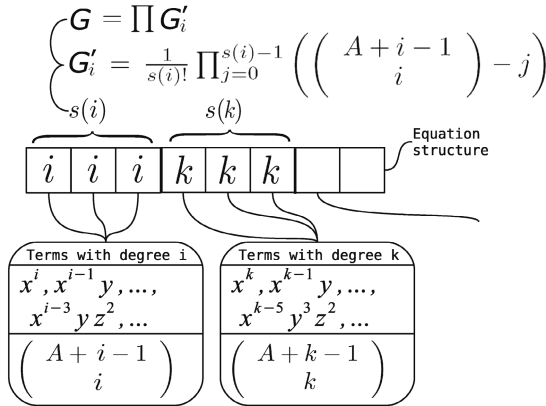


Fig. 3. Calculating the number of polynomial structures in $G'(a_1, a_2, \dots, a_m)$. At the bottom, we have the sets of terms (two sets are depicted, one with terms of degree i and one with terms of degree k). In the middle layer, they are combined in to equation structures, where $s(i)$ and $s(k)$ denote the number of repetitions of i and k values respectively.

elements from a set of n elements. This number equals $\binom{n+a_1-1}{a_1}$. Continuing the same reasoning for all m terms we obtain the number of possible structures in $G'(a_1, a_2, \dots, a_m)$ to be $\prod_{i=1}^m \binom{n+a_i-1}{a_i}$. However, if there are several a_i values that are equal, we will encounter the same term many times, which means that the above formula over-estimates the number of possible structures. The remedy is to divide the number with the factorial of repetitions observed in the tuple. For example, when dealing with the case $G'(5, 5, 3, 2, 2, 2)$, we have to divide with $2!3!$, since 5 is repeated twice and 2 is repeated three times. Note also that each multiplicative term decreases by 1 for each degree value repetition (see Figure 3).

Having the number of equation structures in each G' group, we now turn to the problem of calculating the number of G' groups within each $G(m, l)$. The size of G grows according to the recursive formula $|G(m, l)| = |G(m-1, l-1)| + |G(m, l-m)|$. The first additive term corresponds to the cases when the G' groups contain linear terms (there is an a_i with value 1), while the second corresponds to the cases when all terms in the G' groups have a degree at least 2 (all $a_i > 1$). In the first case, when removing the linear term, we obtain polynomials with $m-1$ terms and length $l-1$. In the second case, we can remove one variable from each of the terms, which leads to polynomials with the same number of terms (m) and length $m-l$. Figure 4 depicts the relationship between G and G' classes of polynomial structures.

Now, having this partitioning and the number of polynomials in each partition, we can decompose the code for each candidate polynomial in four components. First, we have to encode its length l and for this we need $\lceil \log_2(l) \rceil + 2 \lceil \log_2(\lceil \log_2(l) \rceil) \rceil$ bits (the second double logarithm term is necessary, since we do not know the magnitude of l in advance). Second, we encode the number of terms m , for which

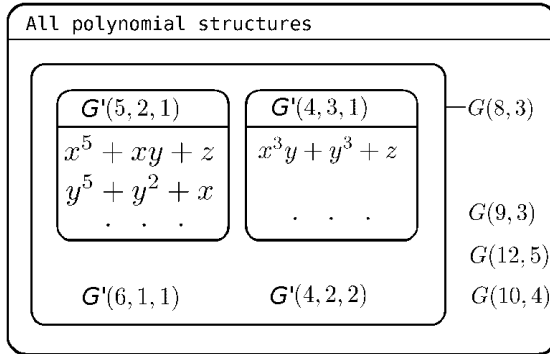


Fig. 4. A general overview of the partitioning of polynomial structures. The small sets correspond to G' classes (e.g., the set $G'(5, 2, 1)$). In turn, we group them into larger classes of structures G that have the same length and size.

we need $\iota_{\gamma}(l)$ bits (remember that $m \leq l$). Third, we can identify a particular G' class within the class $G(m, l)$ using $\iota_{\gamma}(|G(m, l)|)$ bits. Finally, we identify the specific polynomial structure within G' using $\iota_{\gamma}(|G'(a_1, a_2, \dots, a_m)|)$ bits. Putting these four components together gives us the final formula:

$$L(H) = 2_{\gamma}(l) + 2_{\gamma}(\iota_{\gamma}(l)) + \iota_{\gamma}(|G(n, l)|) + \iota_{\gamma}(G'(a_1, a_2, \dots, a_n))$$

for number of bits necessary to encode the polynomial structure.

3.4 Encoding the Linear Regression Model

Rissanen provides a formula for calculating the stochastic complexity of a linear regression model generated by using the method of least squares [6]:

$W = \min_{\gamma} \{ (N - k)_{\gamma}(\hat{\tau}) + k_{\gamma}(\hat{R}) + (N - k - 1)_{\gamma}(\frac{1}{N-k}) - (k - 1)_{\gamma}(k) \}$ where the γ index goes through all the possible subsets of variables involved in the linear regression, k is the number of elements in γ , N is the size of the dataset, $\hat{\tau}$ is the maximum likelihood estimation of the model error, and $\hat{R} = \frac{1}{n} \hat{c}^T (M^T M) \hat{c}$ (where $\hat{c} = (M^T M)^{-1} M^T y$ and M is the matrix of data). The stochastic complexity of the model is then $2W$. Intuitively, this corresponds to the length of the code necessary to encode the errors of the linear regression model ($L(D|H)$) together with the constant parameters of the linear model. The two are closely related and thus the constant parameters are not encoded separately or with the model structure, which is what is usually done in machine learning algorithms when using the MDL principle in an ad-hoc manner. For further details, see [6].

4 Empirical Evaluation

Our work is an extension of CIPER [8]. As described above we have implemented a new refinement operator as CIPER-R. In addition we have implemented the new MDL heuristic in CIPER-R yielding CIPER-MR.

The main goal of the performed experiments is to evaluate the predictive performance of CIPER, CIPER-R and CIPER-MR i.e. evaluate the two different heuristics and the two refinement operators described above. We also made a comparison with the standard regression methods, implemented in the data mining suite Weka [10]. The performance of the methods is evaluated on twenty data sets from the UCI Repository [4] and another publicly available collection of regression data sets [9]. These data sets have been widely used in other comparative studies. In all the experiments presented here, we estimate the predictive performance on unseen examples using 10-fold cross validation. The predictive performance of a model M is measured in terms of relative root mean squared error (RMSE).

In the first phase of the evaluation we do a performance comparison between the original CIPER algorithm (using the old refinement operator) and the CIPER-R algorithm. In this phase we use the ad-hoc heuristic as used in the original implementation of the algorithm [8]. We show that the new refinement operator has better predictive capabilities than the old one. In the next phase we do a performance comparison between the improved MDL heuristics (CIPER-MR) and the ad-hoc MDL heuristic (CIPER-R), now using the new refinement operator. We show that the improved MDL heuristic performs better than the ad-hoc heuristic. In the last phase, we compare CIPER-MR to standard regression algorithms linear regression, regression trees, and model trees.

Table 2. Comparison of the predictive performance of the CIPER-R algorithm and CIPER algorithm

Data set	CIPER	vs	CIPER-R
2dplanes	0.2617	+	0.2276
autoprice	0.3977	-	0.4273
basketball	0.8521		0.8165
bank32nh	0.8525	+	0.8119
bodyfat	0.1617		0.1632
cal-housing	0.5729		0.5975
cpu-small	0.5007		0.4529
elusage	0.4009		0.4009
fried-delve	0.1996		0.1996
house-8l	0.6166	+	0.6097
housing	0.4252		0.4068
kin-8nm	0.8522	+	0.8461
mv	0.0828	+	0.0671
pw-linear	0.4550	+	0.3326
vineyard	0.5899	+	0.5102
puma32h	0.8835		0.8835
delta-aileron	0.6354	+	0.6279
delta-elevators	0.7580		0.7567
elevators	0.7663	+	0.6756
quake	1.0000		1.0000

4.1 Evaluating the New Refinement Operator

Table 2 summarises the results of the performance comparison between the old and the new refinement operator. The statistical significance is tested using a paired t-test and a Wilcoxon signed-rank test. If the p -value is smaller than 0.05 then we reject the null hypothesis, and conclude that the difference is statistically significant. The + sign in the table is used when the improvements we introduce perform significantly better according to the paired t-test and the - sign is used when they perform worse. We can see that CIPER-R performs significantly better than CIPER on nine datasets according to the paired t-test and worse on one dataset. The p -value calculated from the two-tailed Wilcoxon signed-rank test is 0.011 which means that CIPER-R and CIPER have significantly different performance. The p -value calculated from the left-tailed Wilcoxon signed-rank test is 0.005 which means that the difference is not negative. We can conclude that CIPER-R performs significantly better than CIPER.

4.2 Evaluating the New MDL Heuristic

Table 3 summarises the results of the performance comparison between CIPER-MR and CIPER-R. We can see that CIPER-MR performs significantly better than CIPER-R on eight datasets according to the paired t-test. The p -value calculated from the two-tailed Wilcoxon signed-rank test is 0.007 which means

Table 3. Comparison of the predictive performance of the CIPER-R algorithm and CIPER-MR algorithm

Data set	CIPER-R	vs	CIPER-MR
2dplanes	0.2276		0.2270
autoprice	0.4273		0.3946
basketball	0.8165		0.7737
bank32nh	0.8119	+	0.6698
bodyfat	0.1632		0.1626
cal-housing	0.5975		0.5507
cpu-small	0.4529	+	0.1626
elusage	0.4009		0.4009
fried-delve	0.1996		0.1996
house-8l	0.6097		0.5884
housing	0.4068		0.4172
kin-8nm	0.8461	+	0.4637
mv	0.0671	+	0.0314
pw-linear	0.3326		0.3339
vineyard	0.5102		0.6748
puma32h	0.8835	+	0.2453
delta-aileron	0.6279	+	0.5652
delta-elevators	0.7567	+	0.5984
elevators	0.6756	+	0.3387
quake	1.0000		1.0000

Table 4. Predictive performance of commonly used regression methods implemented in Weka: linear regression (LR), model trees (MT), and regression trees (RT) compared to CIPER-MR

Data set	LR	RT	MT	CIPER-MR
2dplanes	0.5427 +	0.2273	0.2270	0.2270
autoprice	0.4839 +	0.5642 +	0.3790	0.3946
basketball	0.7902 +	0.8819	0.7902	0.7737
bank32nh	0.6852	0.7522 +	0.6728	0.6698
bodyfat	0.1648	0.3294 +	0.1580	0.1626
cal-housing	0.6034 +	0.5153	0.4858 -	0.5507
cpu-small	0.5365 +	0.2230 +	0.1725	0.1626
elusage	0.4722 +	0.6569 +	0.4125	0.4009
fried-delve	0.5265 +	0.3573 +	0.2784 +	0.1996
house-8l	0.7878 +	0.6216 +	0.5942	0.5884
housing	0.5330 +	0.5226 +	0.4067	0.4172
kin-8nm	0.7663 +	0.6882 +	0.6070 +	0.4637
mv	0.4309 +	0.0477 +	0.0131 -	0.0314
pw-linear	0.5047 +	0.5687 +	0.3243	0.3339
vineyard	0.6645	0.8321	0.6739	0.6748
puma32h	0.8845 +	0.2897 +	0.2694 +	0.2453
delta-aileron	0.5684	0.5766	0.5434	0.5652
delta-elevators	0.6102 +	0.6220 +	0.6003	0.5984
elevators	0.4324 +	0.5210 +	0.3221	0.3387
quake	0.9984	1.0005	0.9964	1.0000

that CIPER-MR and CIPER-R have significantly different performance. The p -value calculated from the left-tailed Wilkinson signed-rank test is 0.003 which means that the difference is not negative. We can conclude that CIPER-MR performs significantly better than CIPER-R.

4.3 Comparison with Standard Regression Algorithms

Table 4 gives an overview of the predictive performance of standard regression methods on our datasets. We see that CIPER-MR performs better than linear regression and regression trees on most of the datasets. Also according to the Wilkinson signed-rank test CIPER-MR is significantly better than both of them. Compared to model trees CIPER-MR performs significantly better on three and worse on two datasets. The p -value calculated from the two-tailed Wilkinson signed-rank test between CIPER-MR and model trees is 0.952 which means that the performance of the two algorithms is not significantly different.

5 Conclusion and Future Work

In this work, we focus on improving the CIPER algorithm for polynomial regression. Two key parts in the algorithm are the refinement operator on equation

structures and the heuristic evaluation function used to compare equations. The latter takes into account both the error and the complexity of an equation and is based on the MDL principle.

We have proposed a new refinement operator that makes larger steps in the refinement of structures. We have complemented this by a simplification step in the CIPER algorithm. We have proposed a principled MDL function that replaces the ad-hoc MDL function used in CIPER so far.

We empirically evaluate these proposed changes by applying the different variants of CIPER on a number of standard regression datasets. The results suggest that CIPER with the new refinement operator performs better than with the old one. Also, using the principled MDL heuristic function is advantageous to using the ad-hoc MDL heuristic. The new CIPER outperforms linear regression and regression trees and is comparable to model trees.

A number of directions for further work have been identified. We focus here on the question of producing piece-wise polynomial models, by combining tree-based/rule-based models with polynomial equations. Such models may have better predictive capabilities than both equations and tree-based/rule-based models. One way of doing this is clustering the values of the attributes, followed by generating binary attributes for each cluster. We intend to investigate this in the near future.

References

1. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International, Belmont (1984)
2. Grünwald, P., Myung, I., Pitt, M. (eds.): Advances in minimum description length: Theory and applications. MIT Press, Cambridge (2005)
3. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning. Springer, New York (2001)
4. Newman, D., Hettich, C.B.S., Merz, C.: UCI repository of machine learning databases (1998)
5. Rissanen, J.: A universal prior for integers and estimation by minimum description length. *The Annals of Statistics* 11, 416–431 (1983)
6. Rissanen, J.: Mdl denoising. *IEEE Transactions on Information Theory* 46, 2537–2543 (1999)
7. Robnik, M.: Pruning regression trees with mdl. In: Proceedings of the European Conference on Artificial Intelligence, pp. 455–459. John Wiley and Sons, Brighton (1998)
8. Todorovski, L., Ljubič, P., Džeroski, S.: Inducing polynomial equations for regression. In: Proceedings of the Fifteenth International Conference on Machine Learning, pp. 441–452. ACM Press, Banff, Alberta, Canada (2004)
9. Torgo, L.: Regression datasets (1998)
10. Witten, I.H., Frank, E. (eds.): Data mining: Practical machine learning tools and techniques. Morgan Kaufmann, San Francisco (2005)

Handling Numeric Attributes in Hoeffding Trees

Bernhard Pfahringer, Geoffrey Holmes, and Richard Kirkby

University of Waikato, Hamilton, New Zealand
{bernhard, geoff, rkirkby}@cs.waikato.ac.nz

Abstract. For conventional machine learning classification algorithms handling numeric attributes is relatively straightforward. Unsupervised and supervised solutions exist that either segment the data into pre-defined bins or sort the data and search for the best split points. Unfortunately, none of these solutions carry over particularly well to a data stream environment. Solutions for data streams have been proposed by several authors but as yet none have been compared empirically. In this paper we investigate a range of methods for multi-class tree-based classification where the handling of numeric attributes takes place as the tree is constructed. To this end, we extend an existing approximation approach, based on simple Gaussian approximation. We then compare this method with four approaches from the literature arriving at eight final algorithm configurations for testing. The solutions cover a range of options from perfectly accurate and memory intensive to highly approximate. All methods are tested using the Hoeffding tree classification algorithm. Surprisingly, the experimental comparison shows that the most approximate methods produce the most accurate trees by allowing for faster tree growth.

1 Introduction

The ability to learn from numeric attributes is very useful because many attributes needed to describe real-world problems are most naturally expressed by continuous numeric values. The decision tree learners C4.5 and CART successfully handle numeric attributes. Doing so is relatively straightforward, because in the batch learning setting every numeric value is present in memory and available for inspection.

For stream classification algorithms such as the Hoeffding tree [5] the situation is more complicated, although Domingos and Hulten claim that handling numeric attributes is, . . . While this statement is true, the practical implications warrant serious investigation. The storage of sufficient statistics needed to exactly determine every potential numeric threshold, and the result of splitting on each threshold, grows linearly with the number of unique numeric values. A high speed data stream potentially has an infinite number of numeric values, and it is possible that every value in the stream is unique. Essentially this means that the storage required to precisely track numeric attributes is unbounded and can grow rapidly.

For a Hoeffding tree learner to handle numeric attributes, it must track them in every leaf it intends to split. This is extremely expensive and necessitates the development of an effective memory management strategy that will deactivate some leaves in favour of more promising ones when facing memory shortages. This may reduce the impact of leaves with heavy storage requirements but may also significantly hinder growth.

Several approaches to handling numeric attributes during Hoeffding tree induction have been suggested before, and are discussed in Section 2. Prior to this study the methods have not been compared, so Section 3 explores the tradeoff of accuracy versus model size by empirical comparison.

2 Numeric Attributes and Hoeffding Trees

All the methods described in this section attempt to handle numeric attributes at each node of the Hoeffding tree, in a similar fashion to C4.5. Each approach represents an alternative approximation of the C4.5 method.

2.1 VFML

Domingos and Hulten released working source code for a numeric handling method in their VFML package [10]. Numeric attribute values are summarized by a set of ordered bins. The range of values covered by each bin is fixed at creation and does not change as more examples are seen. A hidden parameter serves as a limit on the total number of bins allowed—in the VFML implementation this is hard-coded to allow a maximum of one thousand bins. Initially, for every new unique numeric value seen, a new bin is created. Once the fixed number of bins have been allocated, each subsequent value in the stream updates the counter of the nearest bin.

Essentially the algorithm summarizes the numeric distribution with a histogram, made up of a maximum of one thousand bins. The boundaries of the bins are determined by the first one thousand unique values seen in the stream, and after that the counts of the static bins are incrementally updated.

There are two potential issues with the approach. Clearly, the method is sensitive to data order. If the first one thousand examples seen in a stream happen to be skewed to one side of the total range of values, then the final summary will be incapable of accurately representing the full range of values. The other issue is estimating the optimal number of bins. Too few bins will mean the summary is small but inaccurate, whereas too many bins will increase accuracy at the cost of space. In the experimental comparison the maximum number of bins is varied to test this effect.

2.2 Exhaustive Binary Tree

This method represents the extreme case of achieving perfect accuracy at the necessary expense of storage space. The decisions made are the same that a batch

method would make, because essentially it is a batch method—no information is discarded other than the observed order of values.

Gama et al. present this method in their *Streaming* system [7]. It works by incrementally constructing a binary tree structure as values are observed. The path a value follows down the tree depends on whether it is less than, equal to or greater than the value at a particular node in the tree. The values are implicitly sorted as the tree is constructed.

The only way that this structure saves space over remembering the entire sequence of values is if a value that has already been recorded reappears in the stream. In this case the counter in the binary tree node responsible for tracking that value can be incremented. In every other case a new node will be introduced to the tree. Even then, the overhead of the tree structure will mean that space can only be saved if there are many repeated values. If the number of unique values were limited, as is the case in some data sets, then the storage requirements will be less intensive. In all of the synthetic data sets used for this study the numeric values are generated randomly across a continuous range, so the chance of repeated values is almost zero. The impact of the space cost is measured in the experimental comparison.

Beside memory cost, this method has other potential issues. Because every value is remembered, every possible threshold is also tested when the information gain of split points is evaluated. This makes the evaluation process more costly than more approximate methods. This method is also prone to data order issues. The layout of the tree is established as the values arrive, such that the value at the root of the tree is the first value seen. There is no attempt to balance the tree, so data order is able to affect the efficiency of the tree. In the worst case, an ordered sequence of values will cause the binary tree algorithm to construct a list.

2.3 Quantile Summaries

Researchers in the field of database systems are concerned with accuracy guarantees associated with quantile estimates, helping to improve the quality of query optimizations. Random sampling is often considered as a solution to this problem. Vitter [15] shows how to randomly sample from a data stream, but the non-deterministic nature of random sampling and lack of accuracy guarantees motivate search for other solutions. Munro and Paterson [13] show how an exact quantile can be deterministically computed from a single scan of the data, but this procedure requires memory proportional to the number of elements in the data. Using less memory means that quantiles must be approximated. Early work in quantile approximation includes the P^2 algorithm proposed by Jain and Chlamtac [11], which tracks five markers and updates them as values are observed via piece-wise fitting to a parabolic curve. The method does not provide guarantees on the accuracy of the estimates. Agrawal and Swami [2] propose a method that adaptively adjusts the boundaries of a histogram, but it too fails to provide strong accuracy guarantees. More recently, the method of Alsabti et al. [3] provides guaranteed error bounds, continued by Manku et al. [12] who demonstrate an improved method with tighter bounds.

The quantile estimation algorithm of Manku et al. [12] was the best known method until Greenwald and Khanna [8] proposed a more efficient method with even stronger accuracy guarantees. The method works by maintaining an ordered set of tuples, each of which records a value from the input stream, along with implicit bounds for the range of each value’s true rank. An operation for compressing the quantile summary is defined, guaranteeing that the error of the summary is kept within a desired bound. The quantile summary is said to be ϵ -approximate, after seeing N elements of a sequence any quantile estimate returned will not differ from the exact value by more than ϵN . The worst-case space requirement is shown by the authors to be $O(\frac{1}{\epsilon} \log(\epsilon N))$, with empirical evidence showing it to be even better in practice.

Greenwald and Khanna mention two variants of the algorithm. The first variant is the basic form of the algorithm, that allocates more space only as error is about to exceed the desired ϵ . The other form, used here, is referred to as the k -tuple variant, which imposes a fixed limit on the amount of memory used. The k -tuple method was chosen because it guarantees stable approximation sizes throughout the tree, and is consistent with the majority of other methods by placing upper bounds on the memory used per leaf.

When used to select numeric split points in Hoeffding trees, a k -tuple approach is used where a separate quantile summary is maintained per class label. When evaluating split decisions, all values stored in the tuples are tested as potential split points. Different limits on the maximum number of tuples per summary are examined in the experimental comparison.

2.4 Gaussian Approximation

This method approximates the numeric distribution on a per-class basis in small constant space, using a Gaussian (i.e. normal) distribution. Such a distribution can be incrementally maintained by storing only a few numbers in memory (such as the mean and variance), and is completely insensitive to data order.

Algorithm 1 is a method for incrementally computing the mean and variance of a stream of values. The method only requires three numbers to be remembered. It was derived from the work of Welford [16], and its advantages are studied in [4].

A method similar to this is described by Gama et al. in their UFFT system [6]. When evaluating split points, a single optimal point is computed as derived from the crossover point of two distributions. It is possible to extend their approach to search for split points allowing any number of classes: a set of points spread equally across the range between the minimum and maximum values observed, are evaluated as potential split points. The number of points is determined by a parameter, so the search for split points is parametric, even though the underlying Gaussian approximations are not. For each candidate point the weight of values to either side of the split can be approximated for each class, using their respective Gaussian curves, and the information gain is computed from these weights. This procedure can also cope with extreme cases like distribution with very similar means, but different standard deviations.

Algorithm 1. Numerically robust incremental Gaussian

```

weightSum = weightfirst
mean = valuefirst
varianceSum = 0
for all data points (value, weight) after first do
    weightSum = weightSum + weight
    lastMean = mean
    mean = mean +  $\frac{\text{value} - \text{lastMean}}{\text{weightSum}}$ 
    varianceSum = varianceSum + (value - lastMean) × (value - mean)
end for

anytime output:
return mean = mean
return variance =  $\frac{\text{varianceSum}}{\text{weightSum} - 1}$ 

```

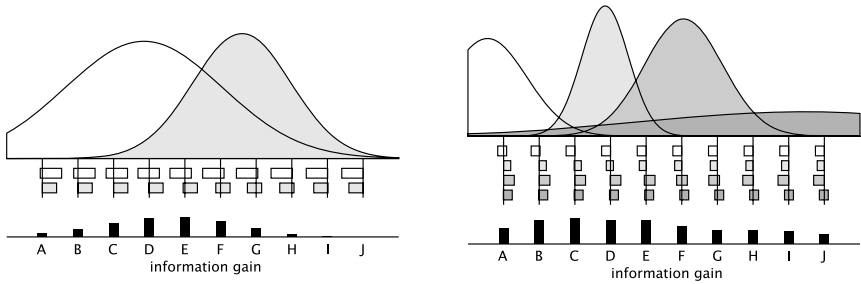


Fig. 1. Gaussian approximation of 2 and 4 classes

The process is illustrated in Figure 1. At the top of each figure are Gaussian curves, each approximating the distribution of values seen for a numeric attribute and labeled with a particular class. The curves can be described using three values; the mean, variance, and the total weight of examples. For example, in the leftmost figure the class shown to the left has a lower mean, higher variance and higher example weight (larger area under the curve) than the other class. Below the curves the range of values has been divided into ten split points, labeled A to J. The horizontal bars show the proportion of values that are estimated to lie on either side of each split, and the vertical bar at the bottom displays the relative amount of information gain calculated for each split. For the two-class example (the left figure), the split point that would be chosen as the best is point E, which according to the evaluation has the highest information gain. In the four-class example (the right figure) the split point C is chosen which nicely separates the first class from the others.

A refinement of this method, found to increase precision at low cost, is used in the final implementation. It involves additionally tracking the minimum and maximum values of each class (the distribution cutoff points in Figure 1 depict these values). This requires storing an extra two counts per class, but precisely

maintaining these values is simple and fast. When evaluating split points the per-class minimum and maximum information is exploited to determine when class values lie completely to one side of a split, eliminating the small uncertainty otherwise present in the tails of the Gaussian curves. From the per-class minimum and maximum, the minimum and maximum of the entire range of values can be established, which helps to determine the position of split points to evaluate.

This simplified view of numeric distributions is not necessarily harmful to the accuracy of the trees it produces because there will be further opportunities during training to refine split decisions on a particular attribute by splitting again further down the tree. The method does not have only one chance of getting the optimal value but can have multiple attempts, where each subsequent attempt will be in a more focused range of values based on increasingly more confident information. In addition, the approximation may prove more robust and resistant to noise than more complicated methods, which concentrate on finer details.

3 Experimental Comparison of Methods

Each method is tested to see how efficiently it produces Hoeffding trees. The methods compared are all based on the basic HTMC (Hoeffding Tree Majority Class) algorithm described in [5], with only the method for handling numeric attributes varied. Predictions are made using majority class at each leaf [4].

The methods compared are listed in Table 11 including the memory limits imposed per numeric attribute per leaf, and with reference to the text explaining each method. Three realistic application scenarios are envisaged where memory for learning is limited to a pre-specified maximum. A sensor node environment (memory limit 100K), a handheld computer environment (32MB) and a server environment (400MB). Eighteen datasets are used (see Table 3) in the evaluation. They are all *synthetic* in order to provide a proper evaluation and have all appeared previously in the data stream literature (see for example [9] and [7]). The experimental methodology used is consistent with other studies, particularly [5], but on a larger scale. In all cases, training takes place over a period of ten hours and testing is accomplished with a holdout set of one million examples.

Table 2 lists the final results averaged over all 18 data sources, sorted by scenario. For the sensor environment the figures for the number of training examples are low because learning was stopped when the last active leaf in a tree had been deactivated. In the handheld case these figures are much higher than in the server case because the former generates smaller trees with fewer active nodes and therefore processes examples faster. The speeds achievable are quoted as percentages of the maximum speed at which these streams can be generated by the experimental software and hardware.

In terms of average accuracy, the four different approaches are easily ranked from best to worst. In all three memory environments, VFML10 is the most

¹ The methods for handling numeric attributes would have a direct influence on predictions if functional leaves [6] were used.

Table 1. Methods compared

name	description	memory limit	section
VFML10	VFML binning method	10 bins	2.1
VFML100	VFML binning method	100 bins	2.1
VFML1000	VFML binning method	1000 bins	2.1
BINTREE	exhaustive binary tree	none	2.2
GK100	Greenwald-Khanna quantile summary	100 tuples per class	2.3
GK1000	Greenwald-Khanna quantile summary	1000 tuples per class	2.3
GAUSS10	Gaussian approximation evaluating 10 split points	5 values per class	2.4
GAUSS100	Gaussian approximation evaluating 100 split points	5 values per class	2.4

accurate on average over all data sources. The second most accurate method in every environment is GAUSS10. The GK x methods are generally third, and BINTREE is consistently the least accurate of the methods on average.

The default number of 1000 bins hard-coded in the original VFML implementation turns out to be the worst performer of the three VFML configurations. The general trend is that smaller numbers of bins, sacrificing accuracy for space per leaf, leads to more accurate trees overall. Requesting more space for numeric approximation reduces the numbers of active tree nodes that can reside in memory, slowing tree growth in a way that impacts final tree accuracy.

The Gaussian method follows this trend, in that it is the smallest approximation tested, permitting the most tree growth and correspondingly accurate trees. Comparing the number of split evaluations tested, it is apparent that the finer grained exploration of GAUSS100 can be harmful. The GAUSS100 trees are on average much deeper than any of the other methods, suggesting that splits on certain numeric attributes are being repeated more often, because in many cases the tree depth exceeds the number of attributes available for splitting. These additional splits are probably very small and unnecessary refinements of previous split choices, and they may be very skewed. This is a symptom of trying to divide the range too finely based on approximation by a single smooth curve.² The GAUSS10 method uses a suitably matched coarse division of only 10 possibilities, which is far less susceptible to this problem.

Comparing the quantile summary methods GK100 and GK1000, having 1000 tuples is helpful in the higher memory environments but harmful in 100KB of memory. Lower numbers of tuples can severely hinder the quantile summary method—a parameter setting of 10 was tested but found to be much worse than any other method, so was omitted from the final results. Figure 2 shows some examples of how much worse the 10-tuple summary can perform. In

² Similar overfitting behaviour is produced by GAUSS1000 which has been omitted from presentation here for space reasons.

Table 2. Final results averaged over all data sources comparing the eight methods

method	accuracy (%)	training examples (millions)	active leaves (hundreds)	inactive leaves (hundreds)	total nodes (hundreds)	tree depth	training speed (%)	prediction speed (%)
100KB memory limit / sensor								
VFML10	87.70	21	0.00	8.13	10.6	11	70	82
VFML100	79.47	13	0.00	3.65	4.50	7	76	85
VFML1000	76.06	1	0.00	0.09	0.14	3	81	88
BINTREE	74.45	1	0.00	0.07	0.11	3	75	89
GK100	82.92	12	0.00	4.03	5.03	8	71	84
GK1000	74.65	1	0.00	0.08	0.13	3	60	88
GAUSS10	86.16	20	0.00	8.87	12.1	12	68	81
GAUSS100	85.33	16	0.01	8.08	11.7	20	64	79
32MB memory limit / handheld								
VFML10	91.53	909	31.8	675	1009	22	16	72
VFML100	90.97	973	5.99	481	704	24	17	73
VFML1000	90.97	951	4.22	412	604	27	17	73
BINTREE	90.48	808	3.68	373	540	22	15	73
GK100	89.96	961	6.89	530	777	34	17	73
GK1000	90.94	937	2.66	403	581	27	16	75
GAUSS10	91.35	874	93.7	683	1166	24	15	69
GAUSS100	90.91	853	92.6	639	1167	50	14	66
400MB memory limit / server								
VFML10	91.41	293	320	80.4	591	24	4	74
VFML100	91.19	142	73.9	143	316	23	4	75
VFML1000	91.12	108	19.0	127	206	22	3	79
BINTREE	90.50	60	13.7	92.9	147	19	2	81
GK100	89.88	158	84.0	145	346	32	4	75
GK1000	91.03	91	17.6	122	197	21	3	80
GAUSS10	91.21	518	540	26.8	891	28	6	73
GAUSS100	90.75	538	566	38.7	998	63	6	66

particular, the graph on RTS (left figure) shows other settings getting very close to 100% accuracy in contrast to the 10-tuple variant achieving less than 65%. Like GAUSS100, GK10 results in excessively deep trees which strongly indicates poor split decisions. More fine grained quantile summaries perform well but the tradeoff between space and accuracy is not as effective as for the GAUSS x and VFML x methods. The performance of GK1000 is similar to BINTREE in several situations, suggesting that it is highly accurate. At the same time, it manages to build larger trees, suggesting that it is more space efficient.

The poor performance of BINTREE shows that in limited memory situations, striving for perfect accuracy at the local level can result in lower accuracy

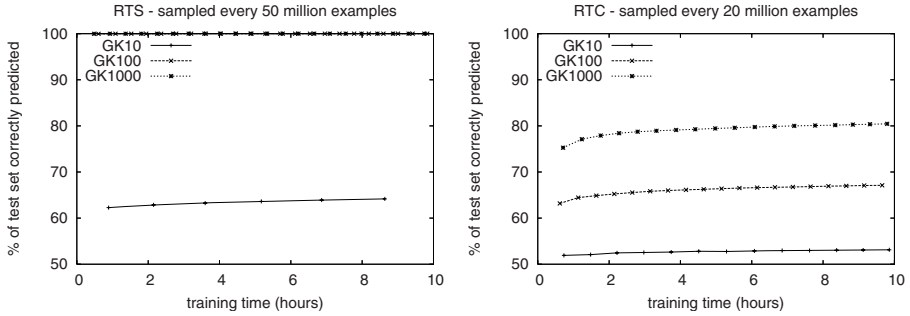


Fig. 2. Examples of poor accuracy achieved by GK10 in 32MB

globally. The problem is most pronounced in the 100KB environment, where tree growth for every data source was halted before the first evaluation took place, some time before 1 million training examples. Similar behaviour is evident in the other two most memory-intensive methods VFML1000 and GK1000, but BINTREE has the highest memory requirements of all, thus suffers the most in tree growth and accuracy. The method is simply too greedy to support reasonable tree induction in this environment. In the other environments it fares better, but is not as successful on average as the more approximate methods.

Table 3 compares the individual final accuracies of the best two methods, VFML10 and GAUSS10. Bold figures indicate a better result, in this case both methods win 20 times each. GAUSS10 loses to VFML10 by a fair margin on RTCN in 400MB, although on this dataset some of the other methods are not much better than GAUSS10 and some are worse still. Some of the worst losses for GAUSS10 occur on GENF2 and GENF5 in 100KB, where it is outperformed by all other methods. These functions are very similar (see 11). The function GENF2 relies on two numeric attributes *salary* and *age*, and GENF5 includes further dependency on a third numeric attribute, *loan*. The trees induced by the Gaussian method were inspected to find the cause of the problem. The trees make the mistake of choosing a discrete attribute *car* with many possible values that is completely irrelevant. After making this mistake the example space is highly segmented, so a lot of extra effort is required to make corrections further down the tree. The Gaussian methods slowly recover to come within reasonably close accuracy, except for the 100KB environment where the lack of space limits any opportunity of recovering. This demonstrates a limitation of the Gaussian method, where the high level of approximation causes the best attributes to be underrated, although the true underlying cause of the issue is unknown. It might relate to an unintentional bias towards certain split types that could potentially be corrected in a style similar to Quinlan’s correction in 14.

Conversely, there are situations where the high level of approximation gives the Gaussian method an advantage over all others. The clearest cases of this are on the data sources RRBFS, RRBFC, WAVE21 and WAVE40. Such a bias is

Table 3. VFML10 vs GAUSS10 accuracy (%)

method→	VFML10			GAUSS10		
	memory limit			memory limit		
dataset	100KB	32MB	400MB	100KB	32MB	400MB
RTS	96.49	99.99	99.98	96.95	99.99	99.99
RTSN	75.80	78.54	78.53	75.20	78.48	78.45
RTC	61.37	83.58	83.87	62.49	83.00	83.02
RTCN	53.63	64.95	66.06	53.63	62.45	61.87
RRBFS	87.69	93.13	92.43	88.56	93.27	92.93
RRBFC	87.84	98.61	97.41	91.36	98.72	98.21
WAVE21	80.80	84.20	83.50	81.21	84.37	84.01
WAVE40	80.28	84.00	83.31	81.20	84.21	83.80
GENF1	95.07	95.07	95.07	95.07	95.07	95.07
GENF2	93.94	94.10	94.10	78.46	94.03	94.00
GENF3	97.52	97.52	97.52	97.50	97.52	97.52
GENF4	94.46	94.67	94.66	93.68	94.67	94.65
GENF5	92.45	92.89	92.84	71.73	92.36	92.15
GENF6	89.70	93.35	93.28	91.89	93.31	93.28
GENF7	96.41	96.82	96.79	96.51	96.81	96.79
GENF8	99.40	99.42	99.42	99.41	99.42	99.42
GENF9	95.80	96.81	96.72	96.07	96.78	96.74
GENF10	99.89	99.89	99.89	99.88	99.89	99.89
average	87.70	91.53	91.41	86.16	91.35	91.21

perhaps not surprising since the generators responsible for these streams use numeric values drawn from random Gaussian distributions.

Analysing space complexity, the amount of memory required per leaf to track n numeric attributes and c classes is $10n + 10nc$ for VFML10 and $5nc$ for GAUSS10. For VFML10 the $10n$ term accounts for storage of the boundary positions, while the $10nc$ term accounts for the frequency counts. This simplified analysis underestimates the true cost of the VFML implementation, which also retains information about the class and frequency of values that lie exactly on the lower boundary of each bin, increasing the precision of decisions. For GAUSS10 the multiplying constant is 5 values per attribute and class because there are 3 values tracking the Gaussian curve and additional 2 numbers tracking the minimum and maximum values.

In theory, at the local level VFML10 should be very sensitive to data order, whereas GAUSS10 should not be sensitive at all. Whether this translates into poorer global decisions during tree induction is not tested by the benchmark generators because all examples are randomly drawn uniformly from the space of possible examples. The right hand side of Figure 3 shows a constructed example where data order has been manipulated to expose VFML10's weakness. GENF2 has been modified so that every sequence of one million examples drawn from the stream has been sorted by the *salary* attribute. In this case the accuracy of GAUSS10 has improved while the early accuracy of VFML10 has dropped markedly. On average GAUSS10 trees reach much larger sizes than the other

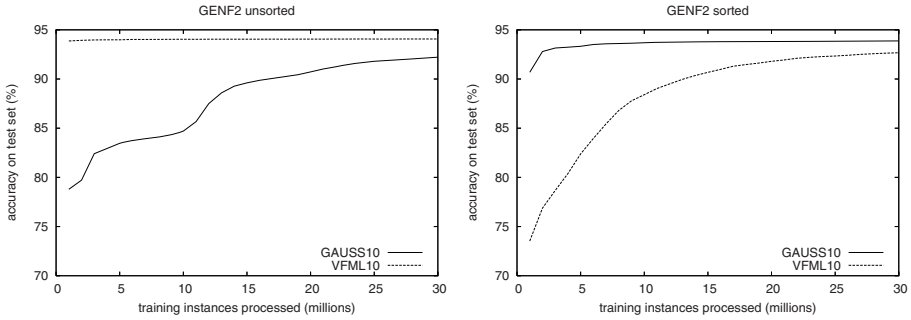


Fig. 3. Effect that example ordering has on learning accuracy in 32MB on the GENF2 data. Left hand side: default random order. Right hand side: modified stream where every consecutive sequence of one million training examples has been sorted on the value of the *salary* attribute.

numeric methods in the same time and space, with many more active leaves. The ability of VFML10 to slowly recover may be partly due to additional tree structure increasing the dispersion of examples down different paths of the tree, reducing the degree to which values encountered at leaves are sorted.

4 Conclusion

We have presented an extension to Gama’s method of using Gaussian distributions to approximate numeric attributes encountered in tree-based classification of data streams. In order to evaluate its efficacy we have designed an experiment involving three realistic memory-limiting data stream environments and eighteen datasets from previous studies. Five main approaches from the literature were implemented, and eight final configurations of algorithm were tested, ranging from perfectly accurate and memory intensive to highly approximate. In experimental comparison, the most approximate methods produced the most accurate trees, by virtue of allowing the most tree growth. The two methods GAUSS10 and VFML10 are highly competitive on most datasets. Of these, GAUSS10 uses less memory and is less susceptible to data order, but is prone to choosing irrelevant attributes in some cases. Adding a bias to correct for this behaviour will be explored in future work.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Database mining: A performance perspective. *IEEE Transactions on Knowledge and Data Engineering* 5(6), 914–925 (1993)
2. Agrawal, R., Swami, A.: A one-pass space-efficient algorithm for finding quantiles. In: *International Conference on Management of Data* (1995)

3. Alsabti, K., Ranka, S., Singh, V.: A one-pass algorithm for accurately estimating quantiles for disk-resident data. In: International Conference on Very Large Databases, pp. 346–355 (1997)
4. Chan, T.F., Lewis, J.G.: Computing standard deviations: Accuracy. *Communications of the ACM* 22(9), 526–531 (1979)
5. Domingos, P., Hulten, G.: Mining high-speed data streams. In: International Conference on Knowledge Discovery and Data Mining, pp. 71–80 (2000)
6. Gama, J., Medas, P., Rocha, R.: Forest trees for on-line data. In: ACM Symposium on Applied Computing, pp. 632–636 (2004)
7. Gama, J., Rocha, R., Medas, P.: Accurate decision trees for mining high-speed data streams. In: International Conference on Knowledge Discovery and Data Mining, pp. 523–528 (2003)
8. Greenwald, M., Khanna, S.: Space-efficient online computation of quantile summaries. In: ACM Special Interest Group on Management Of Data Conference, pp. 58–66 (2001)
9. Holmes, G., Kirkby, R., Pfahringer, B.: Stress-testing hoeffding trees. In: European Conference on Principles and Practice of Knowledge Discovery in Databases, pp. 495–502 (2005)
10. Hulten, G., Domingos, P.: VFML – a toolkit for mining high-speed time-changing data streams (2003), <http://www.cs.washington.edu/dm/vfml/>
11. Jain, R., Chlamtac, I.: The P^2 algorithm for dynamic calculation of quantiles and histograms without storing observations. *Communications of the ACM* 28(10), 1076–1085 (1985)
12. Manku, G.S., Rajagopalan, S., Lindsay, B.G.: Approximate medians and other quantiles in one pass and with limited memory. In: ACM Special Interest Group on Management Of Data Conference, pp. 426–435 (1998)
13. Munro, J.I., Paterson, M.: Selection and sorting with limited storage. *Theoretical Computer Science* 12, 315–323 (1980)
14. Ross Quinlan, J.: Improved use of continuous attributes in C4. *Journal of Artificial Intelligence Research* 4, 77–90 (1996)
15. Vitter, J.S.: Random sampling with a reservoir. *ACM Transactions on Mathematical Software* 11(1), 37–57 (1985)
16. Welford, B.P.: Note on a method for calculating corrected sums of squares and products. *Technometrics* 4(3), 419–420 (1962)

Scaling Record Linkage to Non-uniform Distributed Class Sizes

Steffen Rendle and Lars Schmidt-Thieme

University of Hildesheim, Machine Learning Lab,
Samelsonplatz 1, D-31141 Hildesheim, Germany
{srendle, schmidt-thieme}@ismll.uni-hildesheim.de

Abstract. Record linkage is a central task when information from different sources is integrated. Record linkage models use so-called blockers for reducing the search space by discarding obviously different record pairs. In practice, important problems have Zipf distributed class sizes with some large classes where blocking is not applicable any more. Therefore we propose two novel meta algorithms for scaling arbitrary record linkage models to such data sets. The first one parallelizes problems by creating overlapping subproblems and the second one reduces the search space for large classes effectively. Our evaluation shows that both scaling techniques are effective and are able to scale state-of-the-art models to challenging datasets.

1 Introduction

When data from different sources is collected, objects of different sources may refer to the same underlying entity. For integration of the datasets, duplicates have to be identified. This task is known among others as record linkage [1,2], duplicate detection [3,4] and object identification [5]. For example a price comparison system collects offers from different shops that may refer to the same product (see Table 1). Another example are citation strings that refer to the same publication.

Recent models for solving this task rely on machine learning techniques [6,3,5,7]. For scaling with growing problems they use blockers which restrict the pairs that have to be regarded in time-consuming parts. The key problem with these blockers is that they are supposed to return all positive pairs and remove only those pairs that are obviously negative. Although this technique might scale up for some problems with small uniformly distributed class sizes, it cannot be utilized for other distributions of class sizes like Zipf-distribution.

The contributions of this paper are as follows: (i) We show that the class sizes of some important linkage problems are Zipf distributed which leads to $\Omega(n^2 / \ln^2 n)$ positive pairs. Thus, these problems cannot be solved with standard blocking techniques. (ii) We provide two novel scaling methods for record linkage that efficiently scale arbitrary linkage models to Zipf distributed data sets.

Table 1. Example of price comparison data

Product Name	Brand	Price	Class Label
Photosmart 435 Digital Camera	Hewlett Packard	118.99	c_1
HP Photosmart 435 16MB memory	HP	110.00	c_1
Canon EOS 300D black Kit 18-55	Canon	786.00	c_2
EOS300D+EF-S18-55	<i>unspecified</i>	873.00	c_2
Digital Camera, Olympus, E-300	Olympus	899.00	c_3
Olympus Camedia IR-300 - Digital-Foto	<i>unspecified</i>	273.00	c_4

2 Related Work

The problem of record linkage was first formulated by Newcombe [8] and later put into a mathematical model by Fellegi and Sunter [1]. Today state-of-the-art methods use an adaptive approach based on machine learning techniques like classifiers, clustering or markov logic networks [4,7]. Almost all models for record linkage rely on predicting the equivalence of a pair of objects. As the number of all different pairs is $\frac{n \cdot (n-1)}{2}$ where n is the number of all records, even small problems are not manageable any more. To avoid this problem, record linkage models typically use blockers, which restrict the number of pairs by discarding all obviously different pairs. There have been many proposals for blocking techniques like sorted neighbourhood methods [9], Canopies [10], and adaptive blocking [11,12]. An overview of blocking techniques is given by Baxter et al. [13].

Blocking works fine if there are lots of classes and class sizes are small. In fact we will show that this does not hold for some important record linkage tasks because they have Zipf distributed class sizes which leads to $\Omega(n^2 / \ln^2 n)$ true pairs. This means even a perfect blocker which only returns the true pairs would generate $\Omega(n^2 / \ln^2 n)$ pairs. Consequentially no model exclusively relying on blockers can scale to large problems with Zipf distributed class sizes.

There are some studies of record linkage on large datasets [2,14], but their problems have different characteristics in terms of only two datasets to be merged or very small class sizes. This differs from our problem setting of non-uniformly distributed class sizes that were built up by automatically crawling many sources, like crawling the web. A second main difference is that they use rather simple models for record linkage. The work of Hernández and Stolfo [9] is similar to the above discussed research in terms of small class sizes and simple record linkage models. Similarly to our work, Hernández and Stolfo propose to use clustering for parallelization. They propose hard-clustering in conjunction with their own blocking method of sorted neighborhoods. Whereas inexpensive hard-clustering might be effective for scaling when dealing with small classes, it is difficult to provide high quality splits in problems with large classes. Moreover, parallelizing without any further reduction step of true pairs does not tackle the problem of having $\Omega(n^2 / \ln^2 n)$ true pairs.

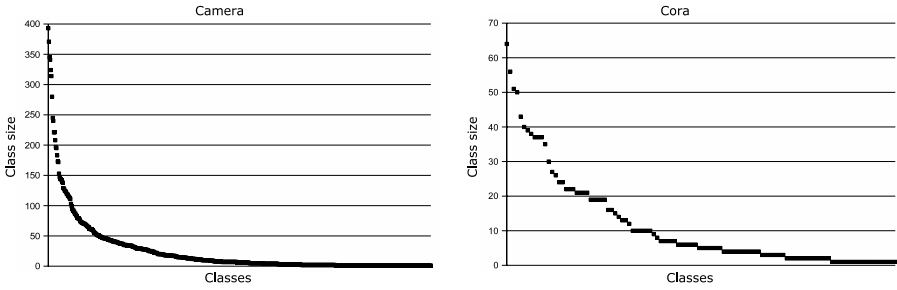


Fig. 1. Distribution of class sizes for the *Camera* and *Cora* dataset

Reducing the size and complexity of a graph has already been studied in multilevel graph partitioning [15]. In their work a graph $G_0 = (V_0, E_0)$ is iteratively coarsened to graphs $G_i = (V_i, E_i)$ with $|V_i| > |V_{i+1}|$. Then partitioning is done on the coarsest graph and afterwards the partitioned graph is uncoarsened. Our proposed method for object reduction is related to coarsening as we also reduce the number of objects, perform the expensive calculations on a the small problem and finally expand the solution. Besides this, graph partitioning and record linkage have different problem settings. The differences to record linkage are that in graph partitioning (1) the number of classes is known, (2) all clusters should have roughly equal size and (3) a sparse set of vertices is given in advance.

3 Problem

Figure 1 shows the distribution of class sizes for the bibliographic *Cora* [10] and the *Camera* dataset from a price comparison system¹. *Cora* contains 1,295 citations to 112 different papers and has 17,184 true pairs. *Camera* has 15,481 offers on 608 digital cameras and has 956,957 true pairs. The classes are sorted by size in descending order. As one can see, the class sizes for both datasets are Zipf-like distributed. Zipf's law² states that the most frequent item occurs twice as often as the next frequent one. The third one's frequency is one third of the most frequent class, etc. Applied to class sizes Zipf's law states that the class size of the i -th class is $1/i$ of the size of the largest class, that means the i -th class contains about $\frac{k_{max}}{i}$ objects where k_{max} is the size of the largest class.

The potential reduction rate of all blocking-based scaling techniques depends on the number of true pairs – that means pairs of records referring to the same entity. In Zipf distributed problems, the number of true pairs correlates to the size of the largest class. Thus, we want to estimate the complexity of the largest class k_{max} . Because all class sizes have to sum up to the number of records n , we can state with Zipf's law:

¹ Mentasys GmbH, Karlsruhe, Germany, <http://www.mentasys.de/>

² For the sake of simplicity, we use an exponent of 1 in all Zipf formulas.

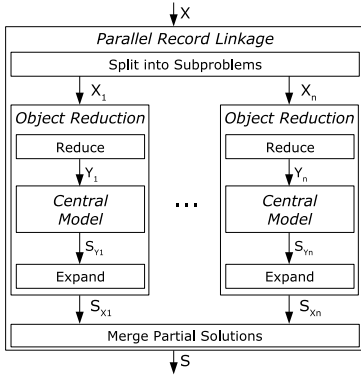


Fig. 2. Combining parallel record linkage with object reduction for scaling a central model

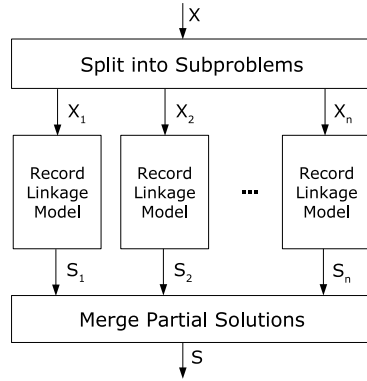


Fig. 3. Parallel Record Linkage: A problem X is split into overlapping subproblems X_1, \dots, X_n . The subproblems are solved independently in parallel and afterwards are merged to a global solution S

$$n = \sum_{i=1}^m \frac{k_{max}}{i} = k_{max} \cdot \sum_{i=1}^m \frac{1}{i} \approx k_{max} \cdot (\ln(m) + \gamma)$$

and thus $k_{max} \approx \frac{n}{\ln(m) + \gamma}$

Where m is the number of all classes, which is unknown in advance and γ is the Euler-Mascheroni constant ($\gamma \approx 0.577$). Now, we can estimate the complexity of k_{max} . As $m \leq n$ also $\ln(m) \leq \ln(n)$ and so we can conclude, that k_{max} grows approximately linear in n . This means, that the size of the largest class k_{max} is in $\Omega(n/\ln n)$. One can conclude that there are $\Omega(n^2/\ln^2 n)$ true pairs in a Zipf distributed record linkage problem.

4 Method

4.1 Scalable Framework

The objective of our framework is to scale up arbitrary record linkage models. In general a record linkage model is a function f_{RL} that generates a partition $f_{RL}(X) \subseteq \mathcal{P}(X)$ of a set of objects X .

We provide two meta algorithms for record linkage that decrease complexity by splitting a problem X in many subproblems X_1, \dots, X_n and by reducing the number of objects within a problem X . Both meta models need a record linkage model for solving the modified problems. Basically, our parallelization technique targets problems with many classes whereas our object reduction method targets

Algorithm 1. Parallelizing by Canopy-Clustering

```

1: procedure CANOPYCLUSTERING( $X$ )
   outputs a set  $P$  of subproblems for objects  $X$ 
2:    $P \leftarrow \emptyset$ 
3:    $C \leftarrow X$  ▷  $C$  is the set of possible centers
4:   while  $C \neq \emptyset$  do
5:      $x \leftarrow \text{random } C$ 
6:      $\text{Canopy}(x) \leftarrow \{y \in X \mid \text{sim}(x, y) > \theta_{\text{loose}}\}$ 
7:      $P \leftarrow P \cup \{\text{Canopy}(x)\}$ 
8:      $C \leftarrow C \setminus \{y \in X \mid \text{sim}(x, y) > \theta_{\text{tight}}\}$ 
9:   end while
10:  return  $P$ 
11: end procedure

```

problems with large classes. Although both algorithms can be used separately, we recommend to combine them so that both aspects are regarded. A useful combination (see figure 2) would be to use parallelizing as outer model, then use object reduction in each parallelized subproblem and solve each reduced subproblem with the record linkage model of your choice – e.g. a classifier based approach like we use in the evaluation chapter.

4.2 Parallel Record Linkage

In general one of the most popular approaches for scaling up systems is parallelizing. Instead of solving one big problem at once, the problem is split in many small problems. These small problems are solved separately and afterwards are merged to a global solution. Our meta model for parallelizing record linkage models work the same way (see figure 3). First, the problem is split into overlapping subproblems. Then each subproblem is solved by another record linkage model and finally the solutions are merged.

Split into Subproblems. Parallelizing is a function f_P that generates a partition of overlapping sets, s.t.:

$$\bigcup_{X_i \in f_P(X)} X_i = X, \quad \emptyset \notin f_P(X) \quad (1)$$

An optimal parallelizing function should generate a large number of subproblems, that have few overlaps, and all objects of the same class should share at least one subproblem.

For splitting a problem into a set of subproblems, clustering with a cheap distance metric can be applied. We suggest to use soft-clustering instead of hard-clustering. This way an object can be placed in several subproblems. The main reason for using soft clustering is that parallelizing has to be fast such that decisions have to be as simple as possible. Especially at the borders of clusters, the degree of uncertainty is high. These serious decisions should not be made

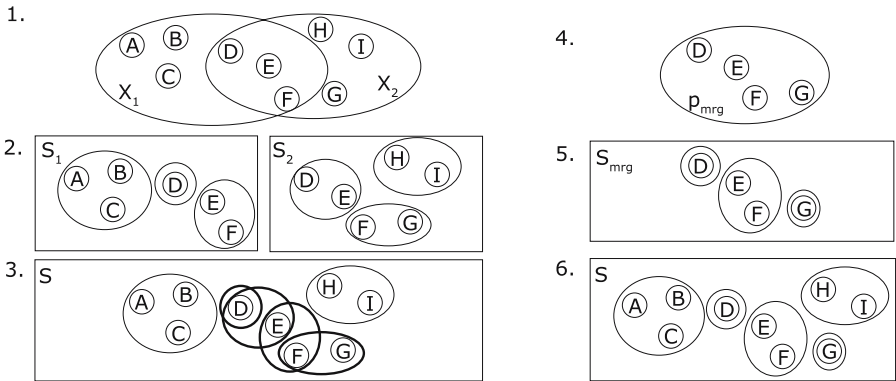


Fig. 4. Merge partial solutions: (1) A problem X was parallelized into overlapping sets X_1 and X_2 . (2) Solutions S_1 for X_1 and S_2 for X_2 are predicted in parallel. (3) The combined solution S overlaps for objects D, E, F, G . (4) A new problem p_{mrg} is created for the overlaps and (5) a solution S_{mrg} is predicted. (6) Now the overall solution S is hard clustered.

by a fast and approximative algorithm. With soft-clustering the algorithm can defer this decision to the more powerful central model.

A possible algorithm for parallelizing is clustering by canopies (see Algorithm 1). The design of this algorithm is inspired by the canopy blocker of McCallum et al. [10]. In contrast to the canopy blocker of McCallum et al., our CANOPY-CLUSTERING algorithm returns overlapping sets of objects. This way, the space complexity is $O(n)$ instead of $O(n^2)$. For CANOPY-CLUSTERING a cheap distance-function like TFIDF-cosine-similarity can be used. An efficient implementation should use an inverted index so that $Canopy(x)$ can be calculated quickly. When training data is available, optimal values for θ_{loose} and θ_{tight} can be found by maximizing both recall and reduction rate.

Merge Partial Solutions. Using soft-clustering for parallelizing comes to the price, that solutions of subproblems may overlap and have to be merged. An example can be found in figure 4. Here, parallelizing puts the object D, E, F both in problem X_1 and X_2 . The two central models predict different equivalences, i.e. the predicted clusters overlap. To solve these overlaps, we suggest to identify overlapping parts and collectively reestimate the class memberships of many overlapping objects. In step 3 of figure 4 the solution S is unsure about the equivalences of D, E, F and G . Thus, a new problem $p_{mrg} = \{D, E, F, G\}$ is created (figure 4, step 4) and is solved by an expensive central model (figure 4, step 5).

Our method for merging subsolutions iteratively eliminates overlaps until a hard clustered solution – that is a partition – is found. Inside each iteration, first of all overlapping regions are identified and subproblems are generated. A new problem p is generated by picking a random object cluster c with overlaps and by

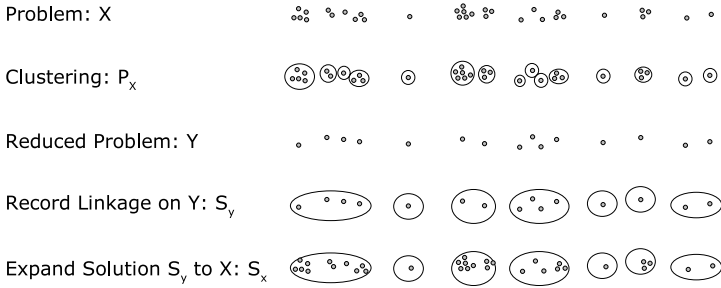


Fig. 5. Object reduction: Problem X is reduced to Y by clustering P_X and creating representatives for each cluster. Record linkage is performed on Y to create a solution S_Y . At last S_Y is expanded to S_X .

extending it with other overlapping object clusters c' . Enlarging the problem p is stopped as soon as no other overlaps with this problem are found or the size of the problem extends a threshold θ_{mrg} . This threshold prevents the new problem to become too large and ensures that it can be solved by the central record linkage model. The set of subproblems P is extended until no more subproblems can be found. Then the subproblems are solved separately and afterwards are merged with the current solution. For finding maximal overlapping clusters, the overlap coefficient can be used:

$$ov(c_1, c_2) = \frac{|c_1 \cap c_2|}{\min\{|c_1|, |c_2|\}} \tag{2}$$

It is easy to show that the proposed algorithm terminates and outputs a hard clustered solution. In each iteration at least one subproblem is generated out of two overlapping object clusters. After solving the subproblem with the central record linkage model, the solution to this subproblem contains no overlaps. So after each iteration the number of overlapping objects decreases.

4.3 Meta Model for Object Reduction

The second scaling technique targets large classes. If we look at problems with lots of classes, parallelizing is an efficient way to generate many subproblems, containing only a few different classes. But in problems with large classes, like in problems with Zipf distributed class sizes, another reduction step is necessary. The reason is that all objects of the largest class will be completely inside one subproblem – under the assumption that parallelizing was effective. In a problem with Zipf-distributed class sizes, the size k_{max} of the largest class is in $\Omega(n/\ln n)$, so this subproblem will have $\Omega(n^2/\ln^2 n)$ true pairs. Thus we suggest to reduce the problem size by eliminating pairs that are obviously identical. This will be done by merging these identical objects before applying an expensive model.

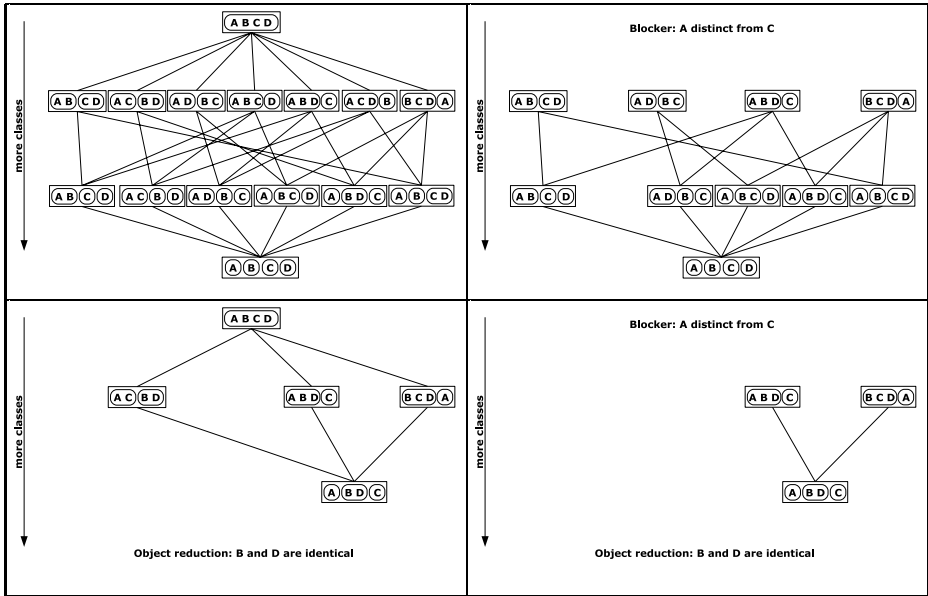


Fig. 6. Hypothesis space for a problem with four objects: A , B , C and D . A blocker reduces the space top-down – here the combination (A, C) is eliminated. Object reduction reduces the space bottom-up – here B and D are identified as identical. Combining object reduction and blocking results in a much smaller hypothesis space – here only three possibilities.

Method. The overall method for solving a problem using object reduction is shown in figure 5. First we start with objects X . These objects are reduced to a subset Y . For this task one can use standard clustering techniques. In our experiments, we choose a HAC algorithm with complete-linkage and a very high threshold. As distance measure, we use the overlap coefficient over 2-grams. The objective of the reduction process is to produce a partition with perfect precision.

After having clustered X to a partition P_X , each cluster of P_X is regarded as an “object”. The reduced object set Y composes of these objects. When sets of objects (= clusters of P_X) should be used as objects in a record linkage problem Y , the question arises how to represent each cluster by a single object. We propose to randomly pick one of the objects of each cluster in P_X , use it as a representative and put it into Y . Normally a random object of a cluster might not be a good representative because clusters might be diverse, but in our case clusters only contain very similar objects. Another approach would be to build prototypes, e.g. cluster centers and use them to build up Y .

Afterwards, the reduced problem Y is solved by an arbitrary record linkage model f_{OI} , that returns a solution on S_Y . The reduced solution S_Y is then expanded to all objects in X , so that a solution S_X results.

Object Reduction and Blockers. When combining our object reduction method with a blocker, the set of potential hypotheses is reduced in two directions (see figure 6). The unrestricted hypothesis space contains all possible solutions, that is all partitions of X . A blocker reduces this space top-down by eliminating object combinations that are obviously different. Object reduction works bottom-up and searches for pairs that are very likely identical. Combining both reduction methods of blocking and object reduction results in a hypothesis space that only contains non-trivial hypotheses. Only these hypotheses have to be regarded by an expensive decision model.

Object reduction is effective particularly with regard to large classes, that appear in problems with non-uniformly distributed class sizes. In this case, it is very likely that many objects of a large class are very similar. This will result in many reductions and a smaller hypothesis space. With this reduction an expensive model can be applied to such problems.

It is important to note, that our proposed model for object reduction only performs the bottom-up step of figure 6. The blocking of obviously false pairs should be done by the central model of your choice. The reason is that blocking is already a standard technique in most models.

5 Evaluation

5.1 Dataset and Model Setup

In our experiments, we evaluate methods for scaling a state-of-the-art record linkage model. We evaluate on the *Cora* and the *Camera* dataset which are described in section 3. As expensive central model, we use the popular approach of training a probabilistic classifier and use it as a learned similarity measure for clustering the objects into sets of equal objects [6,3,16,5]. Analogous to [5], the model uses constrained hierarchical agglomerative clustering with average linkage for collective decisions and as classifier a SVM (for *Cora*) and logistic regression (for *Camera*), respectively. As pairwise features over the textual attributes, this model uses several heuristic similarity measures, that are TFIDF-cosine-similarity, Overlap-coefficient over tokens, 2-grams and 3-grams; the model for the *Camera* dataset additionally uses some domain specific measures.

In each experiment, we randomly label 50% of the objects with their true class label and predict the whole dataset. We report the runtime and the F-Measure on the pairs between unlabeled objects. All experiments were run on a single standard PC. The parallel scaling method would considerably benefit from using more machines because each subproblem could be solved in parallel on another machine. Even though, also with the parallel scaling technique we only use one PC and solve all subproblem sequentially one after another on the same machine.

5.2 Comparison of Scaling Techniques

In the first experiment, we compare our two novel scaling techniques to the popular CANOPY-BLOCKER [10]. As both parallelizing and object reduction are meta models, they can be used in compound models. In all we have five different

Table 2. Runtime and quality results for several scaling methods on the Cora dataset

Scaling Method	Cora	
	F-Measure	Runtime (min)
None	0.948 ± 0.008	206
Blocking	0.954 ± 0.011	121
Object Reduction + Blocking	0.948 ± 0.011	52
Parallelizing + Blocking	0.936 ± 0.011	20
Parallelizing + Object Reduction + Blocking	0.944 ± 0.009	8

scaling setups: (1) no scaling, (2) scaling by blocking, (3) scaling by parallelizing with blocking, (4) scaling by object reduction with blocking and (5) scaling by parallelizing and object reduction with blocking (see figure 2). We run all experiments five times with random train/ test splits.

Table 2 shows the average F-Measure quality with standard deviation and the average runtime for the five scaling approaches on the *Cora* dataset. As one can see, the runtime decreases from 121 minutes to 8 minutes when adding parallelizing and object reduction to a blocker based model. This corresponds to a speedup of 15 or in other words the runtime decreases by 93%. It is interesting, that parallelizing is so effective even though all subproblems are solved sequentially on the same machine. The reason is, that the cost of solving k small problems of the size n/k is much less than solving one problem of the size n .

On the other hand, our proposed scaling methods are also effective in terms of quality. Scaling the blocker based model with both parallelization and object reduction decreases the F-Measure only little from 95.4% to 94.4%. This difference is not statistically significant.

5.3 Scaling a Large Dataset with Parallelizing and Object Reduction

In the second experiment, we examine the components of the scaling framework in more detail on the *Camera* dataset with 15,481 objects and 956,957 true pairs.

CANOPY-CLUSTERING returned 90 subproblems and achieves a recall of 98%. Because a lot of these subproblems are very small, we automatically merged the smallest subproblems, so that each subproblem contains at least 200 objects. This is done, because we have to assure that in each subproblem is enough labeled data for training a pairwise model. Subproblems with more than 200 objects were not modified. In total 50 subproblems remain. In each subproblem object reduction is applied. Reduction achieved a precision of at least 98% in each subproblem. Afterwards the central model is applied on each reduced problem.

As CANOPY-CLUSTERING produces a lot of overlaps, merging the subproblems is no trivial task. The 50 subproblems contain in total 37,780 objects, that means on average each of the 15,481 objects is mentioned in 2.4 subproblems. The local models predict 3098 distinct classes in total for all subproblems. Still there are lots of overlaps, that have to be resolved by the merging process described in section

4.2. Merging needs 3 iterations to resolve all these overlaps and outputs a consistent solution. The F-Measure for the overall solution after all merging iterations is 93%.

The overall execution time was 8 hours and 10 minutes on a single machine. The runtime for splitting the problem into subproblems using CANOPY-CLUSTERING was about 5 minutes. Solving all 50 subproblems took 3 hours and 10 minutes. The largest subproblem was solved in less than 30 minutes. We run all parallelized subproblems one after another on a single machine, so runtime would decrease a lot if multiple machines were used in parallel. Theoretically the runtime for solving the subproblems in parallel can be lowered to 30 minutes using 7 machines of this type.

6 Conclusion and Future Work

We have shown that some important record linkage problems have Zipf-like distributed class sizes and that the standard technique of blocking does not scale with such problems. Thus we have proposed two techniques for scaling arbitrary record linkage models to large problems with non-uniformly distributed class sizes. The first one parallelizes a problem in many overlapping subproblems, so that each subproblem can be solved independently with an arbitrary record linkage model. Afterwards the solutions are merged by iteratively reestimating regions with uncertainty. The second scaling technique reduces the number of objects when dealing with large class sizes. By combining both techniques, record linkage models scale to problems having many classes as well as large classes. We have shown by experiments that our scaling techniques can scale a state-of-the-art record linkage model to challenging datasets and is efficient both in runtime and quality. As far as we know, our framework is the first approach that is able to efficiently solve large record linkage problems with both many classes and large class sizes.

One promising point for future work would be to develop other domain-independent parallelization methods that generate less overlaps than CANOPY-CLUSTERING. This might be achieved by transferring work on adaptive blocking [11,12] to parallelizing.

Acknowledgements

This work was funded by the X-Media project (www.x-media-project.org) sponsored by the European Commission as part of the Information Society Technologies (IST) programme under EC grant number IST-FP6-026978.

References

1. Fellegi, I.P., Sunter, A.B.: A theory for record linkage. *Journal of the American Statistical Association* 64, 1183–1210 (1969)
2. Jin, L., Li, C., Mehrotra, S.: Efficient record linkage in large data sets. In: *Proceedings of the 8th International Conference on Database Systems for Advanced Applications (DASFAA)* (2003)

3. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003), Washington, DC (2003)
4. Culotta, A., McCallum, A.: Joint deduplication of multiple record types in relational data. In: CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 257–258. ACM Press, New York (2005)
5. Rendle, S., Schmidt-Thieme, L.: Object identification with constraints. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM-2006), Hong Kong (2006)
6. Cohen, W.W., Richman, J.: Learning to match and cluster large high-dimensional data sets for data integration. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2002), pp. 475–480. Edmonton, Alberta (2002)
7. Singla, P., Domingos, P.: Entity resolution with markov logic. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM-2006), Hong Kong (2006)
8. Newcombe, H., Kennedy, J., Axford, S., James, A.: Automatic linkage of vital records. *Science* 130, 954–959 (1959)
9. Hernández, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data (SIGMOD-1995), San Jose, CA, pp. 127–138 (1995)
10. McCallum, A.K., Nigam, K., Ungar, L.: Efficient clustering of high-dimensional data sets with application to reference matching. In: Proceedings of the 6th International Conference On Knowledge Discovery and Data Mining (KDD-2000), Boston, MA, pp. 169–178 (2000)
11. Bilenko, M., Kamath, B., Mooney, R.J.: Adaptive blocking: Learning to scale up record linkage. In: Proceedings of the 6th IEEE International Conference on Data Mining (ICDM-2006), Hong Kong (2006)
12. Michelson, M., Knoblock, C.A.: Learning blocking schemes for record linkage. In: Proceedings of AAAI 2006 (2006)
13. Baxter, R., Christen, P., Churches, T.: A comparison of fast blocking methods for record linkage. In: Proceedings of the 2003 ACM SIGKDD Workshop on Data Cleaning, Record Linkage, and Object Consolidation, Washington, DC, pp. 25–27 (2003)
14. Christen, P., Churches, T., Hegland, M.: A parallel open source data linkage system. In: Dai, H., Srikant, R., Zhang, C. (eds.) PAKDD 2004. LNCS (LNAI), vol. 3056, Springer, Heidelberg (2004)
15. Karypis, G., Kumar, V.: Parallel multilevel graph partitioning. In: Proceedings of the 10th International Parallel Processing Symposium (IPPS 1996) (1996)
16. Bilenko, M., Basu, S., Sahami, M.: Adaptive product normalization: Using online learning for record linkage in comparison shopping. In: Proceedings of the 5th IEEE International Conference on Data Mining (ICDM 2005) (2005)

Large-Scale k-Means Clustering with User-Centric Privacy Preservation

Jun Sakuma and Shigenobu Kobayashi

Tokyo Institute of Technology,
4259 Nagatsuta, Midori-ku, Yokohama, Japan

Abstract. A k-means clustering with new privacy-preserving concept, *user-centric privacy preservation*, is presented. In this framework, users can conduct data mining using their private information with storing them in their local storages. After the computation, they obtain only mining result without disclosing private information to others. The number of parties that join conventional privacy-preserving data mining has been assumed to be two. In our framework, we assume large numbers of parties join the protocol, therefore, not only scalability but also asynchronism and fault-tolerance is important. Considering this, we propose a k -mean algorithm combined with a decentralized cryptographic protocol and a gossip-based protocol. The computational complexity is $O(\log n)$ with respect to the number of parties n and experimental results show that our protocol is scalable even with one million parties.

Keywords: privacy, data mining, clustering, k-means, peer-to-peer.

1 Introduction

With the rapid growth of services on the Internet, a large amount of personal information is being stored and exploited for personalized online services. For example, online bookshops suggest that "customers who bought this book also bought these books". As another example, search engines offer personalized search services that reorder search results based on the history of past searches to give more weight to topics that interest each searcher.

If such distributed personal information is integrated among numerous users, variable knowledge for users would be extracted. However, it is pointed out that the combination of personal information can identify individuals with high probability, even when identifiers are removed from personal information [1]. As a technology to extract valuable knowledge from distributed private data sources without disclosure of them, privacy-preserving data mining (PPDM) has attracted attention. Many well-known data mining algorithms have been modified to preserve the privacy of distributed datasets, for example, decision-tree learning [2], association rule mining [3], and k -means clustering [4].

Conventionally, public or private organizations that collect a large amount of personal information are assumed to be PPDM participants. They are responsible for privacy preservation and the PPDM is conducted only when these

organizations reach an agreement. We designate this framework as *server-centric privacy preservation*. Server-centric privacy preservation mainly assumes computations among relatively small number of participants and is well-suited to mining between enterprises. Nevertheless, problems persist, mainly from the perspective of intrinsic owners of personal data.

Consider a PPDM using two databases managed by an online book shop and an online music store. They independently manage personalized recommendation systems based on clients' personal preferences extracted from personal purchase histories. Customers might want to know "customers who bought these books and music also bought these books and music". However, these two shops might not reach agreement for the PPDM because it might not benefit each other through integration of their databases (imagine these two shops are in a competitive relationship). This indicates that customers miss opportunities to enjoy more sophisticated personalized services because individuals cannot lead PPDM using their own personal information at their own initiative.

As a contrasting privacy preservation concept, we specifically investigate *user-centric privacy preservation*. This framework assumes that users store personal information in their local storages not in enterprise databases. If users believe that valuable knowledge would be extracted from the collection of their personal information, they can freely establish or join a session for PPDM and can enjoy data mining without disclosing personal information. For this framework, we present a novel protocol for k -means clustering in this paper. The cross-domain personalized recommendation service is one of motivating applications of this framework. Clustering from combinations of various personal information, such as histories of geographical movement, purchase, web search, and web browsing, is expected to provide more sophisticated personalized services.

Little difference separates server-centric and user-centric privacy preservation. However, the number of parties to be processed is considered to be much larger in user-centric model than in server-centric model. The number of parties is typically assumed to be two in server-centric model; however, we assume $10 \sim 10^6$ parties in user-centric model. In such a large-scale network, both the scalability and the treatment of uncertain networking environment, such as asynchronism and fault-tolerance, are important. To overcome these difficulties, our protocol assumes a network in which users can directly communicate with each other like Peer-to-Peer (P2P) network. Our contribution is mainly on following two protocols:

1. private Asynchronous Average Computation (AAC): This protocol is a cryptographic extension of a gossip-based protocol and computes cluster centers privately.
2. private Nearest Cluster-center Determination (NCD): This protocol privately computes cluster labels and the computation is decentralized over binary trees.

Based on these protocols, our protocol is constructed. The computational complexity of our protocol is $O(\log n)$ with respect to the number of parties n . In addition, the computation is decentralized over each node almost asynchronously in a fault-tolerant manner.

The remainder of this paper is organized as follows. In section 2, we survey precedent studies related to privacy preserving k -means. Section 3 describes building blocks for our protocol. In sections 4 and 5, we propose two primitives: private AAC and private NCD. The proofs of security for these primitives are also shown. In section 6, privacy-preserving k -means is designed with these two primitives. Experimental results are also shown. Section 7 presents our concluding remarks.

2 Related Works and Basic Strategy

Let $X = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} (\mathbf{x}_j \in \mathbb{R}^d)$ be a dataset and $x_{j\ell}$ be the ℓ -th attribute of vector \mathbf{x}_j . k -means clustering partitions the data into k clusters, represented by cluster centers $\boldsymbol{\mu}_i$. Let $Z = \{z_{ij}\}, z_{ij} \in \{0, 1\}$ be a cluster label set. If data \mathbf{x}_j belongs to i -th cluster, cluster label $z_{ij} = 1$. Otherwise, $z_{ij} = 0$. Cluster labels and cluster centers are updated alternately and repeatedly until convergence as follows:

$$\boldsymbol{\mu}_i \leftarrow \frac{\sum_{j=1}^n z_{ij} \mathbf{x}_j}{\sum_{j=1}^n z_{ij}} \quad (\text{updating cluster center}), \tag{1}$$

$$z_{ij} = \begin{cases} 1 & \text{If } i = \arg \min_k (\mathbf{x}_j - \boldsymbol{\mu}_k)^T (\mathbf{x}_j - \boldsymbol{\mu}_k) \\ 0 & \text{otherwise.} \end{cases} \quad (\text{updating cluster label}) \tag{2}$$

Private information in k -means clustering is defined by how the dataset X is partitioned among nodes. Vaidya et al. have proposed a privacy-preserving k -means for the vertically partitioned model where X_j corresponds to a subset of attributes of all data entries [4]. Jha et al. have proposed a privacy-preserving k -means for two parties in the horizontally partitioned model where X_j corresponds to a subset of attributes of all data entries [5]. Jagannathan et al. proposed a privacy-preserving k -means for two parties in the arbitrarily partitioned model where there is not necessarily a simple pattern of how data are shared among parties [6].

In k -means clustering with user-centric privacy preservation, we can naturally assume that the horizontally partitioned model with numerous parties. In this model, privacy-preserving k -means clustering is ideally stated as follows:

Statement 1. Let n parties $P_j (j = 1, \dots, n)$ hold P_j attributes of X . $\cup_j X_j = X$. For any $x \in X_j$, P_j holds x .

Jha’s protocol functions in the horizontally partitioned model. However, this allows each party to learn cluster centers and cannot be extended to a multi-party protocol. Jagannathan’s protocol also functions in the horizontally partitioned model. However, the secure circuit evaluation must be repeated synchronously $O(n^2)$ times in n -party case; consequently, it is not scalable and asynchronous. The description above reveals that conventional protocols are not available or are insufficiently scalable for user-centric privacy preservation.

Our protocol is basically divided into two steps: private Asynchronous Average Computation (AAC) which computes cluster centers privately among all

nodes and private Nearest Cluster-center Determination (NCD) which privately computes cluster labels at each node. Private AAC is a cryptographic extension of a gossip-based protocol [7]. For private NCD, we use a protocol for private comparison of random shares based on Yao’s secure circuit evaluation [8] as a primitive. The same protocol is used in [6] for the same purpose.

3 Building Blocks

Gossip-based Protocol. Gossip-based protocol has been emerging as an approach that achieves scalable and fault-tolerant statistical aggregation [9]. Kowalczyk et al. have proposed a simple gossip-based protocol called ”newscast”, which computes the average of values distributed over P2P networks without transferring all data to a central repository [7]. Let there be n parties $\mathcal{P} = \{P_1, \dots, P_n\}$ and $x_j \in \mathbb{R}$ be the input of P_j . Then, the newscast protocol is described as follows:

1. $\mu_j \leftarrow x_j, t = 1,$
2. Contact a node $P_{j'} \in_r \mathcal{P}$ and receives $\mu_{j'},$
3. $\mu_j \leftarrow \frac{\mu_j + \mu_{j'}}{2}, t \leftarrow t + 1.$ If $t > T,$ terminate the protocol. Else, go to step 2.

\in_r denotes an uniform randomly selection from a set. We call μ_j a local estimate and the number of messages of each node a local degree. Let the average $\sum_j x_j/n$ be $\mu.$ After the asynchronous execution of this protocol, it is proved that the local estimate μ_j converges to μ as cycle $t \rightarrow \infty;$ the variance of μ_i drops on the average by factor $\lambda,$ with $\lambda \leq \frac{1}{2\sqrt{e}}.$ See [7] for details of theoretical properties.

Homomorphic Public-key Cryptosystem. The homomorphic property of public-key cryptosystem is exploited for the computation of encrypted values without decrypting them. The key generation algorithm generates a valid pair (s_k, p_k) of private and public keys. $\mathbb{Z}_N (= [0, \dots, N - 1])$ denotes the domain of data. The encryption of an integer $t \in \mathbb{Z}_N$ is denoted as $c = Enc_{p_k}(t; r),$ where r is a random integer. The decryption is denoted as $t = Dec_{s_k}(c).$ With a valid key pair $(p_k, s_k), Dec_{s_k}(Enc_{p_k}(t; r)) = t$ is required for any t and $r.$ In addition, a public key cryptosystem with homomorphic property satisfies

$$Enc(t_1; r_1) \cdot Enc(t_2; r_2) = Enc(t_1 + t_2; r_1 + r_2), \tag{3}$$

$$Enc(t_1; r_1)^{t_2} = Enc(t_1 t_2; r_1), \tag{4}$$

where $t_1, t_2 \in \mathbb{Z}_N.$ r_1, r_2 are random numbers in \mathbb{Z}_N and changed for every encryption for security reasons. In what follows, $Enc(\cdot; r)$ is described as $Enc(\cdot)$ for simplicity. These properties enable the addition of any two encrypted integers and the multiplication of an encrypted integer by an integer. Paillier cryptosystem is known as a semantically secure cryptosystem [1] with homomorphism [10].

Private Comparison of Random Shares. Let $\mathbf{x} \in \mathbb{Z}_N^d$ be an integer vector that appears in the middle of a computation. For example, \mathbf{x} corresponds to cluster centers at each step in our protocol.

¹ A public-key cryptosystem is semantically secure when a probabilistic polynomial-time adversary cannot distinguish between random encryptions of two elements chosen by herself.

Assume that Alice knows $\mathbf{x}^A = (x_1^A, \dots, x_d^A)$ and Bob knows $\mathbf{x}^B = (x_1^B, \dots, x_d^B)$ where $x_i = x_i^A + x_i^B \pmod N$ and $x_i^A, x_i^B \in_r \mathbb{Z}_N$ for all i . Then, we say that Alice and Bob have a value x_i shared between Alice and Bob while x_i itself is unknown to both of them. Through the protocol of Yao's secure circuit evaluation [8], Alice learns an index i^* such that $i^* = \arg \max_i (x_i^A + x_i^B)$ and nothing else. Bob learns nothing.

A few protocols for private comparison of random shares are known. One of standard solutions is Yao's secure circuit evaluation [8].

4 Private Asynchronous Average Computation

4.1 Protocol for Private AAC

Private AAC is a cryptographic extension of newscast. In this section, we assume one-dimensional value x_j is given to P_j . Although data x_j might be rational number, x_j can be treated as positive integers by adding and multiplying some constant without loss of generality. We also assume that the nodes and the server behave as semi-honest parties [2]. Private AAC is described in figure 1. After the computation, all nodes learn encrypted local estimates $Enc_{pk}(2^{T+1}\mu_j)$ and nothing else, where T is the maximum of cycles. The server learns nothing.

How this protocol correctly computes $Enc_{pk}(2^{T+1}\mu)$ is explained. At step 2, values in each node are encrypted by the server's public key. At step 3, values are sent to $P_{j'}$ which is chosen randomly. At step 4, $P_{j'}$ updates the encrypted local estimate of P_j .

As shown before, the update of newscast is described as $\mu_j \leftarrow \frac{\mu_j + \mu_{j'}}{2}$. This includes division which is not allowable in the homomorphic cryptosystem. Therefore, this update is modified. Assume that each node works synchronously such that all nodes keep identical cycle and updates the local estimate with an equation $\mu_j \leftarrow \mu_j + \mu_{j'}$. In this setting, the local estimate converges to $2^{T+1}\mu$ after T updates because this update is equivalent to that of newscast except that the local estimate is doubled at each update.

In actual case, two local estimates with different cycles might be exchanged. The consistency between the cycle and the local estimate is retained if update equations are modified as

$$\mu_j \leftarrow \begin{cases} \mu_j + 2^{t_j - t_{j'}} \mu_{j'} & (\text{if } t_j \geq t_{j'}) \\ \mu_{j'} + 2^{t_{j'} - t_j} \mu_j & (\text{o.w.}) \end{cases}$$

t_j and $t_{j'}$ are the number of updates in P_j and $P_{j'}$, respectively. This update is also equivalent to that of asynchronous newscast except that the local estimate is doubled at each update. Using the homomorphic property, these update equations are rewritten with encrypted values $c_j = Enc_{pk}(x_j)$ and $c_{j'} = Enc_{pk}(x_{j'})$ as shown:

² A semi-honest party is one who follows the protocol properly with the exception that it maintains a record of all intermediate computations. From accumulated records, semi-honest parties seek to learn other parties' privacy [11].

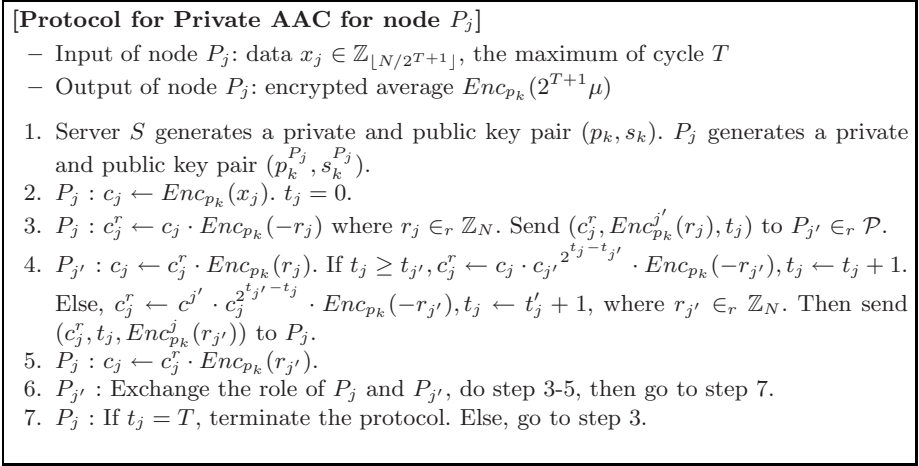


Fig. 1. Private AAC protocol: $Enc_{p_k}(\cdot)$ denotes a message is encrypted by the server’s public key p_k . $Enc_{p_k}^j(\cdot)$ denotes a message is encrypted by the public key of P_j , $p_k^{P_j}$.

$$c_j \leftarrow \begin{cases} c_j \cdot c_{j'}^{2^{t_j-t_{j'}}} & (\text{if } t_j \geq t_{j'}) \\ c_{j'} \cdot c_j^{2^{t_{j'}-t_j}} & (\text{o.w.}) \end{cases}$$

In the protocol, randomizations are introduced to prevent the server from eavesdropping on local estimates. The local estimate is randomized by r_j at step 3 and $r_{j'}$ at step 4: these randomization are resolved at step 4 and step 5, respectively.

Convergence Property: The convergence property of newscast is inherited with these update equations. The domain of x_j is set to $\mathbb{Z}_{\lfloor N/2^{T+1} \rfloor}^d$ such that $2^{T+1}\mu$ exists in the message space \mathbb{Z}_N . This protocol can be modified such that the squared and the weighted mean are estimated, which are used in next section. See Appendix in detail.

Gossip-based averaging guarantees $\mu_j - \mu \leq \epsilon$ for any ϵ with probability $1 - \delta$ after $\lceil 0.581(\log n + 2 \log \sigma + 2 \log \frac{1}{\epsilon} + 1 \log \frac{1}{\delta}) \rceil$ cycles of newscast and data variance σ^2 [7]. It follows that the computational complexity of this protocol is $O(\log n)$ with fixed ϵ, δ , and σ .

Even when some node leaves the network suddenly, the protocol is still processed fault-tolerant because communication between two nodes does not affect communication between two other nodes: the theoretical convergence property will not be followed in such a situation; however, it would give an estimate close to the average if the number of disappeared nodes is not very large.

Security and Privacy: Finally, a theorem for the security of this protocol is presented.

Theorem 1. *Let \mathcal{P} be a set of nodes in a network. Let S be a server. Let P_j and $P_{j'}$ be two nodes in \mathcal{P} . Let x_j and $x_{j'}$ be the data of P_j and $P_{j'}$ respectively. Let μ_j and $\mu_{j'}$ be the local estimates of P_j and $P_{j'}$ respectively. Let μ be the average of μ_j and $\mu_{j'}$. Let ϵ and δ be two positive real numbers. Let T be a positive integer. Let \mathcal{P} and S execute the Private AAC protocol for T cycles. Then, the probability that the local estimate of P_j is within ϵ of the average μ is at least $1 - \delta$.*

The proof should be stated following the standardized proof methodology of secure multi-party computation as shown in [11]; however, because of the limitation of the space we briefly explain why this protocol does not reveal private information as follows:

Because all exchanged local estimates are all encrypted by the server’s public key in this protocol, no nodes can decrypt messages related to local estimates and no nodes can know other nodes’ private values. The server can decrypt exchanged local estimates, however, all local estimates are randomized by r_j, r'_j at each node at each step. Random values r_j and r'_j are exchanged with encrypted by public key of each node. It follows that the server cannot resolve randomizations by itself. Thus, the sever never learns any knowledge related to nodes’ private value, x_j , either.

5 Private Determination of the Nearest Cluster Center

We present a protocol by which a node privately determines the nearest cluster center by taking encrypted cluster centers as inputs. Let the i -th cluster center be $\mu_i = (\mu_{i1}, \dots, \mu_{id})$. After the execution of private AAC, all nodes share encrypted estimates of cluster centers and squared cluster centers (see appendix) for all i as follows:

$$c_i = (c_{i1}, \dots, c_{id}) = (Enc_{p_k}(2^{T+1}\mu_{i1}), \dots, Enc_{p_k}(2^{T+1}\mu_{id})), \tag{7}$$

$$c_i^{(2)} = (c_{i1}^{(2)}, \dots, c_{id}^{(2)}) = (Enc_{p_k}(2^{T+1}\mu_{i1}^2), \dots, Enc_{p_k}(2^{T+1}\mu_{id}^2)), \tag{8}$$

Now the problem for node P_j is to determine i^* such that $i^* = \arg \min_i d(\mathbf{x}_j, \mu_i)$ (the index of the nearest cluster center) without disclosing \mathbf{x}_j and knowing μ_i . If two parties have random shares of $d(\mathbf{x}_j, \mu_i)$ for all i , the nearest cluster center can be privately determined by private comparison of random shares (fig. 2, top).

First, the way to prepare random shares of these distances is described. Let d_{ij}^A and d_{ij}^B denote random shares of $2^{T+1}d(\mathbf{x}_j, \mu_i)$. Given \mathbf{x}_j, c_i and $c_i^{(2)}$ for all i at step 1a, node P_j computes

$$c_i = \prod_{\ell=1}^d Enc_{p_k}(2^{T+1}x_{j\ell}^2) \cdot \prod_{\ell=1}^d c_{i\ell}^{-2x_{j\ell}} \cdot \prod_{\ell=1}^d c_{i\ell}^{(2)} \cdot Enc_{p_k}(-d_{ij}^B)$$

$$= Enc_{p_k}(2^{T+1}(\mathbf{x}_j^T \mathbf{x}_j - 2\mu_i^T \mathbf{x}_j + \mu_i^T \mu_i) - d_{ij}^B) = Enc_{p_k}(2^{T+1}d(\mathbf{x}_j, \mu_i) - d_{ij}^B).$$

Decrypting c_i , the server obtains $d_{ij}^A = 2^{T+1}d(\mathbf{x}_j, \mu_i) - d_{ij}^B$ at step 1b : both have random shares of $d(\mathbf{x}_j, \mu_i)$. Then at step 2, P_j learns the nearest cluster center using private comparison of random shares.

Assuming private comparison of random shares is secure, this protocol securely and correctly determines the nearest cluster center. However, the computational time of the server is seriously large. The server must reply requests from all nodes: the computational complexity is $O(n)$ with respect to the number of nodes. This would be apparently a bottleneck because we assume n is very large in user-centric privacy-preservation.

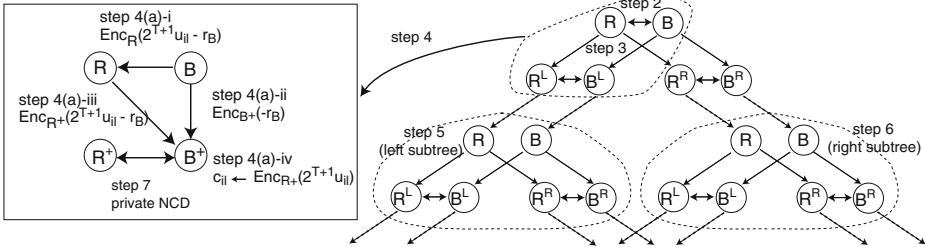
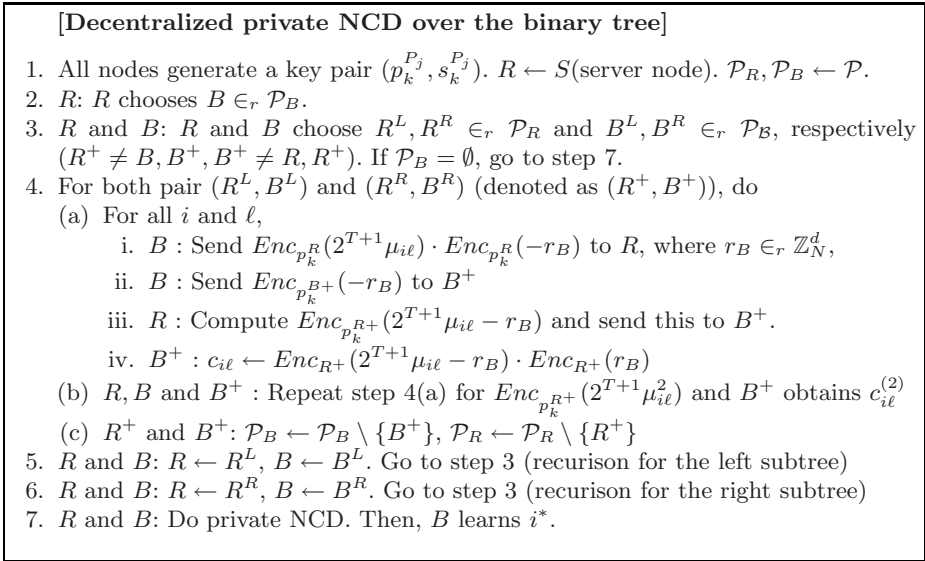
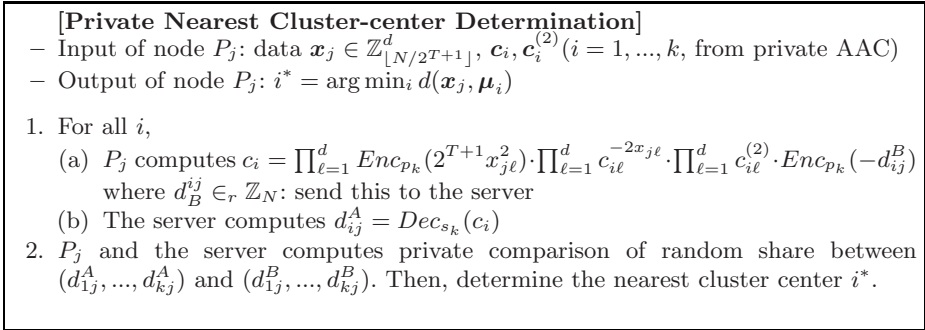


Fig. 2. Top: Non-decentralized private NCD; middle and bottom: Decentralized private NCD. For simplicity, $\text{Enc}_{p_k, X}$ is described as Enc_X in the bottom figure.

Messages exchanged during the protocol execution except ones related to private comparison of random shares are denoted as

$$\left(Enc_{p_k^R}(2^{T+1}\mu_{i\ell} - r_B), Enc_{p_k^{B+}}(-r_B), Enc_{p_k^{R+}}(2^{T+1}\mu_{i\ell} - r_B), Enc_{p_k^{R+}}(d_{iB^+}^A) \right).$$

Assume that $R, R^+, B,$ and B^+ observes all messages exchanged for protocol execution. During step 2 in which R is engaged, R observes a randomized value $2^{T+1}\mu_{i\ell} - r_B$: the remainder of the messages are encrypted in the form that R cannot decrypt. R^+ observes two randomized values $2^{T+1}\mu_{i\ell} - r_B$ and $d_{iB^+}^A$; the remainder of the messages are encrypted in the form that R^+ cannot be decrypt. B observes a random value r_B : the remainder of the messages are encrypted in the form that B cannot decrypt. B^+ observes two random values r_B and $d_{iB^+}^A$: the remainder of the messages are encrypted in the form that B^+ cannot decrypt. Thus, all messages transferred in the network are randomized or encrypted in the form that they cannot decrypt. Consequently, they learn only the result of private comparison and nothing else.

6 k-Means Clustering Using Private AAC and NCD

k -means clustering with user-centric privacy preservation is designed with two primitives presented in previous sections. The outline of the protocol is summarized as follows:

1. Server S and node P_j generate a key pair (p_k^S, s_k^S) and $(p_k^{P_j}, s_k^{P_j})$.
2. All nodes join private AAC and obtain c_i for $i = 1, \dots, k$.
3. All nodes and the server join decentralized private NCD : encrypted squared mean $c_i^{(2)}$ are propagated and z_{ij} for $i = 1, \dots, n, j = 1, \dots, k$ are determined.
4. If termination conditions are satisfied, terminate the protocol. Else, go to step 2.

At step 2, encrypted cluster centers c_i and encrypted squared cluster centers are privately computed. Then, at step 3, distributed private NCD is conducted: $c_i^{(2)}$ are propagated and cluster label are determined by taking c_i and $c_i^{(2)}$ as inputs. Step 4 judges the convergence. The simplest termination condition is to stop the protocol after a fixed number of iterations. As an alternative, the convergence can be judged by sharing whether cluster labels are changed or not among nodes. These can be computed using private AAC and the termination protocol will be discussed in the longer version of this paper.

Here, we briefly show what the nodes and the server learn from execution of the protocol. At step 2, nodes just obtains a sequence of encrypted cluster centers, c_i and $c_i^{(2)}$, which cannot be decrypted by nodes. As shown in theorem 1, all nodes and the server learn nothing about other nodes' private data from private AAC. At step 3, both nodes learn a sequence of z_{ij} that includes which cluster center is (or was) the nearest to the node's data. Unfortunately, this result slightly deviates form what statement 1 in section 2 describes because statement 1 does not allow to disclose cluster labels at intermediated steps. Nevertheless, P_j does not learn any from the protocol execution other than a series of z_{ij} .

After the execution of the protocol, computed results, such as cluster centers, number of nodes belonging to each cluster, nodes belonging to the same cluster,

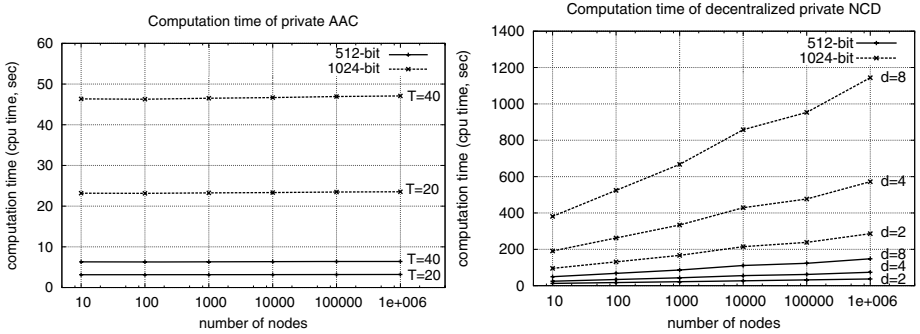


Fig. 3. Computational time of private AAC ($d = 1$, left) and private NCD ($d = 2, 4, 8, k = 2$, right). 1024, 512 bit-key were used.

can be shared among nodes if they reach an agreement. Private sharing of these results among nodes requires additional protocols. These issues will be discussed in other paper, too.

Computational Analysis: To investigate the scalability, we show experimental results of these protocols. As homomorphic cryptosystem, Paillier cryptosystem with 512/1024-bit key was used. The server and the node program were implemented by J2SE ver. 1.5.0. The number of nodes was varied from $n = 10$ to 10^6 . Actually, the execution in the real P2P network with a million nodes is unrealistic. Instead, we simulated the P2P network environment on a personal computer; both the server and nodes program were run on a Xeon2.8GHz(CPU), 2GB(RAM) Windows PC.

We simulated private AAC and measured the computation time per node. In private AAC, the computation time of encryption and decryption is much larger than the communication time because the unit message length is at most the length of the cipher, 512 or 1024 bit. Therefore, we only measured the overall computation time without communications overhead.

For a unit variance dataset and a network with $n = 10^6$ nodes, 25 cycles are required to guarantee that the local estimate μ_j in each node is within 10^{-6} from the correct average μ with 95% probability. Considering this, we set the maximum cycle to $T = 20, 40$ and the dimension $d = 1$ in experiments. The results are shown in figure 3 left. As shown, private AAC with a million of nodes is completed within 50 (sec) at most.

Next, we evaluated the computation time of decentralized private NCD. In this experiment, we did not construct a complete binary trees but partial binary trees whose width was 2 and depth was $\lceil \log_2 n \rceil + 4$ because of the limitation of the memory. This setting is sufficient for the evaluation of the computation time because the number of recursions is at most $\lceil \log_2 n \rceil + 4$ as shown in section 5 and the computation is executed in parallel in the P2P network environment. We set $k = 2$: the computation time from the step 2 to the termination of the

protocol was measured. In the largest setting (1024-bit key, $d = 8$, $n = 10^6$), it costs 1200 (sec)(=20 min) at most. This result includes the propagation of $e^{(2)}$. Without this, the computation time is decreased by half.

Finally, the computation time of privacy-preserving k -means was evaluated. Here, we assume two cases. Case 1 is a small-scale setting ($d = 2$, $k = 2$, $n = 1000$). Case 2 is a large-scale setting ($d = 4$, $k = 4$, $n = 10^6$). Both cases assume $T = 40$ and 1024-bit key. For a single iteration of k -means in case 1, step 2 costs about 180 (sec) and step 3 costs 660 (sec). From those, k -means clustering is expected to be converged within a few hours. In case 2, step 2 costs 740 (sec) and step 2 costs 9100 (sec). For single iteration, it costs about 2.7 hour. In this case, clustering is expected to be completed within a couple of days.

7 Conclusion

We propose a protocol for k -means clustering with user-centric privacy preservation based on two novel protocols: private AAC and private NCD. Our protocol is implemented totally asynchronous and fault-tolerant and is scalable even with a million users. Computation time is dependent on that of encryption and decryption strongly. With more sophisticated implementation of the cryptosystem would improve the computation time drastically. Nevertheless, considering a million users participate in the protocol, we can conclude that experimental results are remarkably scalable. Other data mining algorithms with user-centric privacy preservation is our future work.

References

1. Sweeney, L.: k -anonymity: a model for protecting privacy. *International Journal on Uncertainty, Fuzziness and Knowledge-based Systems* 10(5), 557–570 (2002)
2. Lindell, Y., Pinkas, B.: Privacy Preserving Data Mining. In: Bellare, M. (ed.) *CRYPTO 2000*. LNCS, vol. 1880, pp. 20–24. Springer, Heidelberg (2000)
3. Evfimievski, A., Srikant, A., Agrawal, R., Gehrke, J.: Privacy Preserving Mining of Association Rules. In: *ACM SIGKDD Int'l conf. on Knowledge discovery in data mining*, pp. 217–228 (2002)
4. Vaidya, J., Clifton, C.: Privacy-preserving k -means clustering over vertically partitioned data. In: *ACM SIGKDD Int'l conf. on Knowledge discovery in data mining*, pp. 206–215 (2003)
5. Jha, S., Kruger, L., McDaniel, P.: Privacy Preserving Clustering. In: *European Symposium on Research in Computer Security*, pp. 397–417 (2005)
6. Jagannathan, G., Wright, R.: Privacy-preserving distributed k -means clustering over arbitrarily partitioned data. In: *ACM SIGKDD Int'l conf. on Knowledge discovery in data mining*, pp. 593–599 (2005)
7. Kowalczyk, W., Vlassis, N.: Newscast EM. In: *NIPS 17*, MIT Press, Cambridge (2005)
8. Yao, A.C.-C.: How to Generate and Exchange Secrets. In: *IEEE Symposium on FOCS*, pp. 162–167 (1986)
9. Kempe, D., Dobra, A., Gehrke, J.: Computing aggregate information using gossip. In: *IEEE Symposium on FOCS*, pp. 482–491 (2003)

10. Paillier, P.: Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 223–238. Springer, Heidelberg (1999)
11. Goldreich, O.: Foundations of Cryptography II: Basic Applications. Cambridge University Press, Cambridge (2004)

Appendix: Private AAC for Weighted Average

Cluster centers are computed as weighted averages as $\mu_j = \sum_j z_{ij} \mathbf{x}_j / \sum z_{ij}$, where $z_{ij} \in \{0, 1\}$. If weights z_{ij} are not mutually private, this is easily solved by execution of private AAC only between nodes whose z_{ij} is one. However, if the weight z_{ij} is private, this violates the privacy. Then we modify updates of private AAC as follows:

- Step 1,2, and, 3 is the same with private AAC
- When $z_{ij} = 1$ and $z_{ij'} = 1$, updates are executed normally
- When $z_{ij} = 1$ and $z_{ij'} = 0$, P_j updates the local estimate of $P_{j'}$ normally. However, $P_{j'}$ merely increments the cycle of P_j and the local estimate of P_j is merely multiplied by $2^{t_j - t_{j'} + 1}$ or 2 to retain the consistency between the cycle and the local estimate of P_j .
- When $z_{ij} = 0$ and $z_{ij'} = 0$, both parties merely increment the cycle of the local estimate for each other.

The behavior of each node is controlled by z_{ij} . As a consequence of updates show above, the update progresses only between parties with $z_{ij} = z_{ij'} = 1$ without disclosure of z_{ij} . Let the probability that two nodes whose z_{ij} is both 1 be p . Then, T/p cycles are required to guarantee the same convergence property with private AAC protocol.

Appendix: Private AAC for Squared Average

A protocol to compute the squared average is described. We need to use Yao’s secure circuit evaluation for this protocol. Let x^A and x^B inputs of two parties, Alice and Bob, respectively. x^A and x^B are random shares of x . Then, consider a Yao’s protocol to compute y^A and y^B as outputs of Alice and Bob, respectively, in which y^A and y^B are random shares of x^2 . By Yao’s protocol, squared averages are obtained through the computation between the sever and node P_1 as follows (arbitral node is available instead of P_1).

1. P_1 generates a random number $r_1 \in \mathbb{Z}_N$, computes $c_{i\ell} \cdot Enc_{p_k}(-r_1)$ and send this to the sever.
2. Server decrypts this and obtains $r_S = \mu_{i\ell} - r_1$ (r_1 and r_S are random shares of $\mu_{i\ell}$).
3. Between the server and P_1 , run the Yao’s protocol. Then, P_1 and the server obtains y_1 and y_S , respectively, in which y_1 and y_S are random shares of $\mu_{i\ell}^2$.
4. The server compute $c_S \leftarrow Enc_{p_k}(y_S)$ and send to P_1 .
5. P_1 computes $c_{i\ell}^{(2)} \leftarrow Enc_{p_k}(y_1) \cdot c_S = Enc_{p_k}(y_S + y_1)$

After P_1 obtains $c_{i\ell}^{(2)}$ for all i and ℓ , P_1 become the first blue node. Then, $c_{i\ell}^{(2)}$ can be propagated thorough the binary tree as done in figure 2.

Semi-Supervised Local Fisher Discriminant Analysis for Dimensionality Reduction

Masashi Sugiyama¹, Tsuyoshi Idé², Shinichi Nakajima³, and Jun Sese⁴

¹ Tokyo Institute of Technology, Tokyo, Japan
sugi@cs.titech.ac.jp

<http://sugiyama-www.cs.titech.ac.jp/~sugi/>

² IBM Research, Kanagawa, Japan
goodidea@jp.ibm.com

³ Nikon Corporation, Saitama, Japan
nakajima.s@nikon.co.jp

⁴ Ochanomizu University, Tokyo, Japan
sesejun@is.ocha.ac.jp

Abstract. When only a small number of labeled samples are available, supervised dimensionality reduction methods tend to perform poorly due to overfitting. In such cases, unlabeled samples could be useful in improving the performance. In this paper, we propose a semi-supervised dimensionality reduction method which preserves the global structure of unlabeled samples in addition to separating labeled samples in different classes from each other. The proposed method has an analytic form of the globally optimal solution and it can be computed based on eigendecompositions. Therefore, the proposed method is computationally reliable and efficient. We show the effectiveness of the proposed method through extensive simulations with benchmark data sets.

1 Introduction

The goal of dimensionality reduction is to obtain a low-dimensional representation of high-dimensional data samples while preserving most of ‘intrinsic information’ contained in the original data. Once dimensionality reduction is carried out appropriately, the compact representation of the data can be used for various succeeding tasks such as visualization and classification.

In supervised learning scenarios where data samples are accompanied with class labels, Fisher Discriminant Analysis (FDA) [1] is a popular dimensionality reduction method. FDA seeks an embedding transformation such that between-class scatter is maximized and within-class scatter is minimized. FDA works very well if samples in each class are Gaussian with the common covariance structure. However, it tends to give undesired results if samples in a class form several separate clusters or there exist outliers [1]. To overcome this drawback, Local Fisher Discriminant Analysis (LFDA) [2] has been proposed [2], which localizes the between-class and within-class scatter matrices. LFDA works well even when within-class multimodality or outliers exist. Furthermore, LFDA overcomes critical limitation of original

FDA in dimensionality reduction—the dimension of the FDA embedding space should be less than the number of classes [1], while LFDA does not suffer from this restriction in general.

However, the performance of LFDA (and all other supervised dimensionality reduction methods) tend to be degraded when only a small number of labeled samples are available. Thus, the supervised methods overfit embedding spaces to the labeled samples. In such cases, it is effective to make use of unlabeled samples which are often available abundantly, i.e., semi-supervised learning. The book [3] showed through extensive simulations that PCA, which is an unsupervised dimensionality reduction method for preserving the global data structure, works moderately well in semi-supervised learning scenarios.

Although PCA is reported to work well, it may not be the best choice in semi-supervised learning due to its unsupervised nature. In this paper, we propose a new semi-supervised dimensionality reduction method which smoothly bridges LFDA and PCA so that we can control our reliance on the global structure of unlabeled samples and information brought by (a small number of) labeled samples. We experimentally show that the proposed method, which we refer to as SELF, compares favorably with other methods. Note that SELF maintains the same computational advantage of LFDA and PCA, i.e., a global solution can be analytically computed based on eigendecompositions. Therefore, SELF is still computationally efficient and reliable.

2 Preliminaries

In this section, we formulate the linear dimensionality reduction problem and give some mathematical backgrounds.

2.1 Formulation

Let $\mathbf{x}_i \in \mathbb{R}^d$ ($i = 1, 2, \dots, n$) be d -dimensional samples and let $\mathbf{X} \equiv (\mathbf{x}_1 | \mathbf{x}_2 | \dots | \mathbf{x}_n)$. Let $\mathbf{z} \in \mathbb{R}^r$ ($1 \leq r \leq d$) be a low-dimensional representation of a high-dimensional sample $\mathbf{x} \in \mathbb{R}^d$, where r is the dimensionality of the reduced space. We focus on linear dimensionality reduction, i.e., using a $d \times r$ transformation matrix \mathbf{T} , an embedded representation \mathbf{z} of a sample \mathbf{x} is obtained as

$$\mathbf{z} = \mathbf{T}^\top \mathbf{x}, \quad (1)$$

where $^\top$ denotes the transpose of a matrix or a vector.

Many dimensionality reduction techniques developed so far involve an optimization problem of the following form:

$$\mathbf{T}_{OPT} \equiv \operatorname{argmax}_{\mathbf{T} \in \mathbb{R}^{d \times r}} \left[\operatorname{tr} \left(\mathbf{T}^\top \overline{\mathbf{C}} \mathbf{T} (\mathbf{T}^\top \underline{\mathbf{C}} \mathbf{T})^{-1} \right) \right]. \quad (2)$$

Let $\{\varphi_k\}_{k=1}^d$ be the generalized eigenvectors associated with the generalized eigenvalues $\{\lambda_k\}_{k=1}^d$ of the following generalized eigenvalue problem:

$$\overline{\mathbf{C}} \varphi = \lambda \underline{\mathbf{C}} \varphi. \quad (3)$$

We assume that the generalized eigenvalues are sorted as $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_d$ and the generalized eigenvectors are normalized as $\varphi_k^\top \underline{\mathbf{C}} \varphi_k = 1$ for $k = 1, 2, \dots, d$. Note that this normalization is often automatically carried out by an eigensolver. Then a solution \mathbf{T}_{OPT} is analytically given as $(\varphi_1 | \varphi_2 | \dots | \varphi_r)$ (e.g., [1]):

When addressing dimensionality reduction problems, we often face with a matrix of the following pairwise form [2]:

$$\mathbf{S} \equiv \frac{1}{2} \sum_{i,j=1}^n W_{i,j} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \tag{4}$$

where \mathbf{W} is some n -dimensional matrix. Let \mathbf{D} be the n -dimensional diagonal matrix with $D_{i,i} \equiv \sum_{j=1}^n W_{i,j}$, and let $\mathbf{L} \equiv \mathbf{D} - \mathbf{W}$. Then \mathbf{S} is expressed as $\mathbf{S} = \mathbf{X} \mathbf{L} \mathbf{X}^\top$, which is positive semi-definite.

2.2 Principal Component Analysis (PCA)

A fundamental unsupervised dimensionality reduction method is Principal Component Analysis (PCA).

Let $\mathbf{S}^{(t)}$ be the scatter matrix:

$$\mathbf{S}^{(t)} \equiv \sum_{i=1}^n (\mathbf{x}_i - \boldsymbol{\mu})(\mathbf{x}_i - \boldsymbol{\mu})^\top, \tag{5}$$

where $\boldsymbol{\mu} \equiv \frac{1}{n} \sum_{i=1}^n \mathbf{x}_i$. The PCA transformation matrix \mathbf{T}_{PCA} is defined as

$$\mathbf{T}_{PCA} \equiv \operatorname{argmax}_{\mathbf{T} \in \mathbb{R}^{d \times r}} \left[\operatorname{tr} \left(\mathbf{T}^\top \mathbf{S}^{(t)} \mathbf{T} (\mathbf{T}^\top \mathbf{T})^{-1} \right) \right]. \tag{6}$$

That is, PCA seeks a transformation matrix \mathbf{T} such that scatter in the embedding space is maximized. A solution \mathbf{T}_{PCA} is given with $\overline{\mathbf{C}} = \mathbf{S}^{(t)}$ and $\underline{\mathbf{C}} = \mathbf{I}_d$, where \mathbf{I}_d is the identity matrix on \mathbb{R}^d .

2.3 Locality-Preserving Projection (LPP)

Another useful unsupervised dimensionality reduction technique is Locality Preserving Projection (LPP) [4].

Let \mathbf{A} be the affinity matrix, i.e., the n -dimensional square matrix with $A_{i,j}$ being the affinity between \mathbf{x}_i and \mathbf{x}_j . We assume that $A_{i,j} \in [0, 1]$; $A_{i,j}$ is large if \mathbf{x}_i and \mathbf{x}_j are ‘close’ and $A_{i,j}$ is small if \mathbf{x}_i and \mathbf{x}_j are ‘far apart’. There are several different manners of defining \mathbf{A} , e.g., based on nearest neighbors or the heat kernel. Through the paper, we use the definition of the affinity matrix \mathbf{A} , i.e.,

$$A_{i,j} = \exp \left(- \frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_i \sigma_j} \right). \tag{7}$$

σ_i is the local scaling around \mathbf{x}_i defined by $\sigma_i = \|\mathbf{x}_i - \mathbf{x}_i^{(k)}\|$, where $\mathbf{x}_i^{(k)}$ is the k -th nearest neighbor of \mathbf{x}_i . A heuristic choice of $k = 7$ has been shown to be useful through extensive simulations [5, 2].

Let $\mathbf{S}^{(n)}$ and $\mathbf{S}^{(l)}$ be the $n \times n$ and the $l \times l$ scatter matrices defined by

$$\mathbf{S}^{(n)} \equiv \mathbf{X}\mathbf{D}^{(n)}\mathbf{X}^\top, \quad \mathbf{S}^{(l)} \equiv \frac{1}{2} \sum_{i,j=1}^n W_{i,j}^{(l)}(\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \quad (8)$$

where $\mathbf{D}^{(n)}$ is the n -dimensional diagonal matrix with $D_{i,i}^{(n)} \equiv \frac{1}{n} \sum_{j=1}^n A_{i,j}$ and $W_{i,j}^{(l)} \equiv \frac{1}{n} A_{i,j}$. The LPP transformation matrix \mathbf{T}_{LPP} is defined as

$$\mathbf{T}_{LPP} \equiv \operatorname{argmax}_{\mathbf{T} \in \mathbb{R}^{d \times r}} \left[\operatorname{tr} \left(\mathbf{T}^\top \mathbf{S}^{(n)} \mathbf{T} (\mathbf{T}^\top \mathbf{S}^{(l)} \mathbf{T})^{-1} \right) \right]. \quad (9)$$

That is, LPP seeks a transformation matrix \mathbf{T} such that (\mathbf{x}_i, y_i) data pairs in the original space \mathbb{R}^d are kept close in the embedding space \mathbb{R}^r . Thus, LPP tends to preserve the local structure of the data. A solution \mathbf{T}_{LPP} is given with $\overline{\mathbf{C}} = \mathbf{S}^{(n)}$ and $\underline{\mathbf{C}} = \mathbf{S}^{(l)}$.

2.4 Fisher Discriminant Analysis (FDA)

A popular supervised dimensionality reduction technique is Fisher Discriminant Analysis (FDA) [1]. When discussing supervised learning problems, we suppose that we have n' labeled samples $\{(\mathbf{x}_i, y_i)\}_{i=1}^{n'}$, where $y_i \in \{1, 2, \dots, c\}$ is a class label associated with the sample \mathbf{x}_i and c is the number of classes. Let n'_m be the number of labeled samples in class $m \in \{1, 2, \dots, c\}$.

Let $\mathbf{S}^{(b)}$ and $\mathbf{S}^{(w)}$ be the between-class scatter matrix and the within-class scatter matrix:

$$\mathbf{S}^{(b)} \equiv \sum_{m=1}^c n'_m (\boldsymbol{\mu}_m - \boldsymbol{\mu})(\boldsymbol{\mu}_m - \boldsymbol{\mu})^\top, \quad \mathbf{S}^{(w)} \equiv \sum_{m=1}^c \sum_{i:y_i=m} (\mathbf{x}_i - \boldsymbol{\mu}_m)(\mathbf{x}_i - \boldsymbol{\mu}_m)^\top, \quad (10)$$

where $\boldsymbol{\mu}_m \equiv \frac{1}{n'_m} \sum_{i:y_i=m} \mathbf{x}_i$. The FDA transformation matrix \mathbf{T}_{FDA} is defined as

$$\mathbf{T}_{FDA} \equiv \operatorname{argmax}_{\mathbf{T} \in \mathbb{R}^{d \times r}} \left[\operatorname{tr} \left(\mathbf{T}^\top \mathbf{S}^{(b)} \mathbf{T} (\mathbf{T}^\top \mathbf{S}^{(w)} \mathbf{T})^{-1} \right) \right]. \quad (11)$$

That is, FDA seeks a transformation matrix \mathbf{T} such that between-class scatter is maximized and within-class scatter is minimized in the embedding space \mathbb{R}^r . A solution \mathbf{T}_{FDA} is given with $\overline{\mathbf{C}} = \mathbf{S}^{(b)}$ and $\underline{\mathbf{C}} = \mathbf{S}^{(w)}$.

The between-class scatter matrix $\mathbf{S}^{(b)}$ has at most rank $c - 1$ [1]. This implies that FDA allows us to obtain at most $c - 1$ meaningful features; the remaining features found by FDA are arbitrary in the null space of $\mathbf{S}^{(b)}$. This is an essential limitation of FDA in dimensionality reduction.

2.5 Local Fisher Discriminant Analysis (LFDA)

Local Fisher Discriminant Analysis (LFDA) is a supervised dimensionality reduction method [2] which overcomes vulnerability of original FDA against within-class multimodality or outliers [1].

Let $\mathbf{S}^{(lb)}$ and $\mathbf{S}^{(lw)}$ be the between-class scatter matrix and the within-class scatter matrix defined by

$$\mathbf{S}^{(lb)} \equiv \sum_{i,j=1}^{n'} \frac{W_{i,j}^{(lb)}}{2} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \quad \mathbf{S}^{(lw)} \equiv \sum_{i,j=1}^{n'} \frac{W_{i,j}^{(lw)}}{2} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \tag{12}$$

where $\mathbf{W}^{(lb)}$ and $\mathbf{W}^{(lw)}$ are the n' -dimensional matrices with

$$W_{i,j}^{(lb)} \equiv \begin{cases} A_{i,j}(1/n' - 1/n'_{y_i}) & \text{if } y_i = y_j, \\ 1/n' & \text{if } y_i \neq y_j, \end{cases} \quad W_{i,j}^{(lw)} \equiv \begin{cases} A_{i,j}/n'_{y_i} & \text{if } y_i = y_j, \\ 0 & \text{if } y_i \neq y_j. \end{cases} \tag{13}$$

The LFDA transformation matrix \mathbf{T}_{LFDA} is defined as

$$\mathbf{T}_{LFDA} \equiv \operatorname{argmax}_{\mathbf{T} \in \mathbb{R}^{d \times r}} \left[\operatorname{tr} \left(\mathbf{T}^\top \mathbf{S}^{(lb)} \mathbf{T} (\mathbf{T}^\top \mathbf{S}^{(lw)} \mathbf{T})^{-1} \right) \right]. \tag{14}$$

$A_{i,j}(1/n' - 1/n'_{y_i})$ is negative while $A_{i,j}/n'_{y_i}$ and $1/n'$ are non-negative. Thus, LFDA seeks a transformation matrix \mathbf{T} such that nearby data pairs in the same class are made close and the data pairs in different classes are made apart; far apart data pairs in the same class are not imposed to be close. Samples in different classes are separated from each other irrespective of their affinity values. A solution \mathbf{T}_{LFDA} is given with $\overline{\mathbf{C}} = \mathbf{S}^{(lb)}$ and $\underline{\mathbf{C}} = \mathbf{S}^{(lw)}$.

When $A_{i,j} = 1$ for all i, j (i.e., no locality), $\mathbf{S}^{(lw)}$ and $\mathbf{S}^{(lb)}$ are reduced to $\mathbf{S}^{(w)}$ and $\mathbf{S}^{(b)}$ [2]. Thus, LFDA could be regarded as a localized variant of FDA. The between-class scatter matrix $\mathbf{S}^{(b)}$ has at most rank $c - 1$, while its local counterpart $\mathbf{S}^{(lb)}$ usually has full rank (given $n' \geq d$). Therefore, LFDA can be applied to dimensionality reduction into r dimensional spaces.

3 Semi-Supervised LFDA (SELF)

In this section, we propose a new dimensionality reduction method for semi-supervised learning scenarios. From here on, we consider the case where, among all samples $\{\mathbf{x}_i\}_{i=1}^n$, only $\{\mathbf{x}_i\}_{i=1}^{n'}$ ($1 \leq n' \leq n$) are labeled and the rest are unlabeled.

3.1 Basic Idea

When only a small number of labeled samples are available, supervised dimensionality reduction methods tend to find embedding spaces which are overfitted to the labeled samples. In such situations, using unlabeled samples is often effective—indeed, the book [3] showed through extensive simulations that PCA works well on the whole; our experimental results in Section 4 also show that PCA is sometimes better than LFDA. This means that preserving the global structure of all samples in an unsupervised manner can be better than strongly relying on class information provided by a small number of labeled samples.

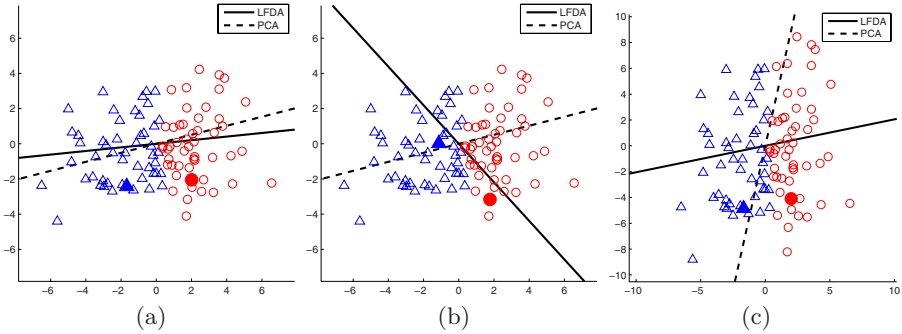


Fig. 1. Illustrative examples of LFDA and PCA for toy data sets. Circle/triangle symbols denote samples in positive/negative classes and filled/unfilled symbols denote labeled/unlabeled samples; solid and dashed lines denote 1-dimensional embedding spaces found by LFDA and PCA, respectively (onto which data samples will be projected).

Figure 1 depicts 2-dimensional 2-class examples; circle/triangle symbols denote samples in positive/negative classes and filled/unfilled symbols denote labeled/unlabeled samples; solid and dashed lines denote 1-dimensional embedding spaces found by LFDA and PCA, respectively (onto which data samples will be projected). For the data set in Figure 1(a), both LFDA and PCA can find good embedding spaces which well separate unlabeled samples in different classes from each other. However, for the data set in Figure 1(b), LFDA finds an embedding space that is overfitted to the labeled samples. On the other hand, in the case of Figure 1(c), PCA does not work well due to its unsupervised nature.

The above result implies that LFDA and PCA can compensate for the weakness of each other, i.e., LFDA can utilize label information, while PCA can avoid overfitting. Our simulation results with benchmark data sets in Section 4 also show that LFDA and PCA work in a complementary manner. Motivated by these facts, we propose a method that smoothly controls our reliance on the global structure of unlabeled samples and class information brought by labeled samples. We refer to the proposed method as *Self-Adaptive LFDA* (SELF).

The embedding transformations of LFDA and PCA can be analytically computed based on the eigendecompositions. So we combine the eigenvalue problems of LFDA and PCA and solve them together. This allows us to maintain the computational efficiency and reliability of LFDA and PCA.

3.2 Definition

More specifically, we propose solving the following generalized eigenvalue problem:

$$S^{(rlb)}\varphi = \lambda S^{(rlw)}\varphi, \tag{15}$$

where $\mathbf{S}^{(rlb)}$ and $\mathbf{S}^{(rlw)}$ are local between-class scatter matrix and local within-class scatter matrix defined by

$$\mathbf{S}^{(rlb)} \equiv (1 - \beta)\mathbf{S}^{(lb)} + \beta\mathbf{S}^{(t)}, \quad \mathbf{S}^{(rlw)} \equiv (1 - \beta)\mathbf{S}^{(lw)} + \beta\mathbf{I}_d. \quad (16)$$

$\beta \in [0, 1]$ is a trade-off parameter—SELF is reduced to LFDA when $\beta = 0$, and SELF is reduced to PCA when $\beta = 1$. In general, SELF inherits characteristics of both LFDA and PCA (this will be discussed in detail in Section 3.3). The solution of SELF can be computed in the same way as LFDA or PCA.

3.3 Properties

First, we give an interpretation of $\mathbf{S}^{(rlb)}$. The matrix $\mathbf{S}^{(rlb)}$ can be expressed as

$$\mathbf{S}^{(rlb)} \equiv \frac{1}{2} \sum_{i,j=1}^n W_{i,j}^{(rlb)} (\mathbf{x}_i - \mathbf{x}_j)(\mathbf{x}_i - \mathbf{x}_j)^\top, \quad (17)$$

where $\mathbf{W}^{(rlb)}$ is the n -dimensional matrix with

$$W_{i,j}^{(rlb)} \equiv \begin{cases} (1 - \beta)A_{i,j}(1/n' - 1/n'_{y_i}) + \beta/n & \text{if } y_i = y_j, \\ (1 - \beta)/n' + \beta/n & \text{if } y_i \neq y_j, \\ \beta/n & \text{otherwise.} \end{cases} \quad (18)$$

The first case in Eq. (18) is negative if $\beta < \frac{A_{i,j}n(n' - n'_{y_i})}{A_{i,j}n(n' - n'_{y_i}) + n'n'_{y_i}} (< 1)$. This implies that SELF tries to make sample pairs in the same class close if β is small, while it separates them from each other if β is large. Thus the local data structure in the same class tends to be preserved when β is small, but it is no longer preserved when β is large. The second case in Eq. (18) is always positive for any $\beta \in [0, 1]$, implying that SELF always tries to make sample pairs in different classes apart for any β . This would be natural in semi-supervised learning scenarios. The third case in Eq. (18) is always non-negative, implying that unlabeled samples are separated from each other for preserving the global data structure.

Next, we give an interpretation of $\mathbf{S}^{(rlw)}$. When $\beta = 0$, $\mathbf{S}^{(rlw)}$ ($= \mathbf{S}^{(lw)}$) could be ill-conditioned—this is crucial particularly when the dimension d of the original data space is larger than the number n' of labeled samples. In such situations, $\beta\mathbf{I}_d$ included in $\mathbf{S}^{(rlw)}$ works as a regularizer and SELF can avoid overfitting to the labeled samples. Therefore, SELF is regarded as a regularized variant of LFDA and would be more stable and reliable than original LFDA particularly when the number of labeled samples is small. Note that unlike Eq. (17), $\mathbf{S}^{(rlw)}$ does not have a pairwise expression since \mathbf{I}_d can not be expressed in a pairwise form.

3.4 Numerical Examples

For illustrating how SELF behaves, let us use the [MNIST](http://www.cs.toronto.edu/~roweis/data.html) data set¹. The data set consists of 400 gray-scale face images (40 people, 10 images per person);

¹ <http://www.cs.toronto.edu/~roweis/data.html>

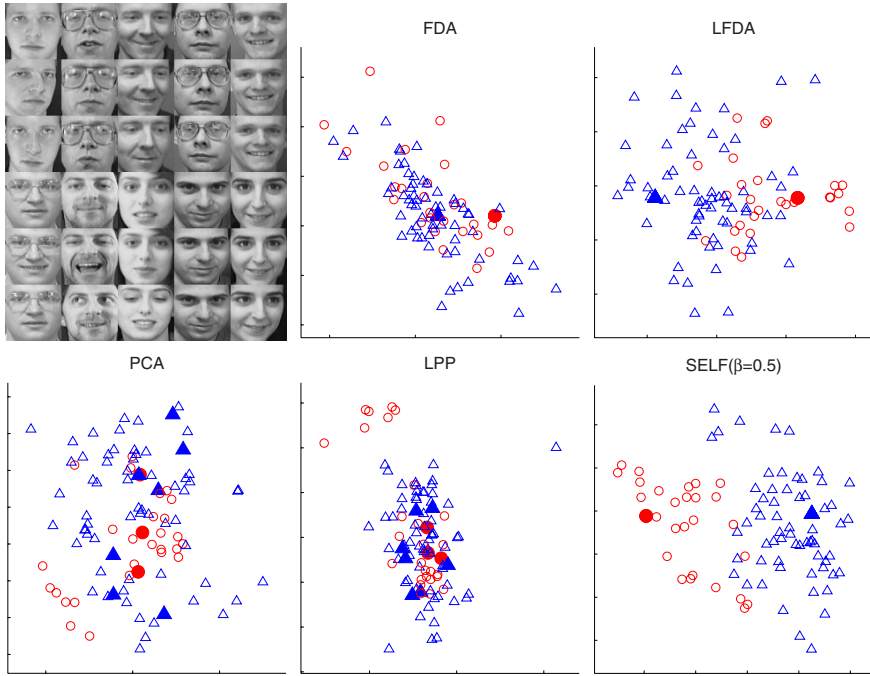


Fig. 2. Embedded face samples (glasses vs. non-glasses). Circle/triangle symbols are faces with/without glasses and filled/unfilled symbols are labeled/unlabeled samples.

each image consists of 4096 ($= 64 \times 64$) pixels and each pixel takes an integer value between 0 and 255 as the intensity level. In this simulation, we use the image samples of only 10 subjects (i.e., totally 100 images) for making the visualization results clear. We note that the result does not change essentially (but visually denser) when all 400 images are used.

Among 10 people used for the experiments, 3 subjects are with glasses and other 7 are without glasses (see the top-left pictures of Figure 2). Our task is to embed the face images into a two-dimensional space so that the subjects with and without glasses are separated from each other. We treat 1 image per person as labeled (i.e., totally 3 faces with glasses and 7 faces without glasses) and the rest are treated as unlabeled. Since each class contains several different subjects, this data set is thought to possess within-class multimodality.

The embedded results are shown in Figure 2, where circle/triangle symbols are faces with/without glasses and filled/unfilled symbols are labeled/unlabeled samples. The figure shows that FDA and LFDA perfectly separate the labeled samples in different classes from each other. However, unlabeled samples tend to be mixed due to an overfitting phenomenon. PCA and LPP tend to mix the labeled samples in different classes due to the unsupervised nature. Consequently, unlabeled samples in different classes are also mixed. On the other hand, SELF with $\beta = 0.5$ clearly separates the labeled samples in different classes from each

other, and at the same time, it also nicely separates the unlabeled samples in different classes from each other. We note that, in this visualization simulation, the result of SELF is not sensitive to the choice of the trade-off parameter β ; the results are almost unchanged for $0.01 \leq \beta \leq 0.99$.

4 Simulations

In this section, we experimentally evaluate the performance of relevant dimensionality reduction methods using standard classification benchmark data sets.

The book [3] conducted systematic experiments for comparing semi-supervised learning methods. The results showed that each method performs very well for a particular type of data sets. However, at the same time, it tends to be poor for other kinds of data sets. Thus, the performance of semi-supervised learning methods is highly dependent on the type of data sets and there seems to be no single best method. On the other hand, $1 - \text{misclassification rate}$ is shown to be stable for various data sets, although it may not be the best possible method in semi-supervised classification. For avoiding the bias caused by the choice of the learning methods, we decided to use the 1-nearest neighbor classifier in our experiments.

The misclassification rate is sometimes monotone increasing as the dimensionality is reduced [2]. In such cases, if the best dimensionality is chosen, e.g., by cross-validation, the largest dimension is mostly chosen (i.e., no dimensionality reduction). Then we may not be able to compare the performance of dimensionality reduction methods in a meaningful way. Prefixing the reduced dimensionality r to some number is a possible option for avoiding the above problem, but the evaluation results can significantly depend on the choice of the dimensionality. Based on this argument, we decided to use the $\int_0^r \text{misclassification rate}$ over reduced dimensions (or equivalently the area under the classification error curve) as our error metric, which we believe to be reasonable in the current experiments.

First, we employ the benchmark data sets taken from the book [3], which consist of 9 semi-supervised data sets. We refer to them as the `ss` data sets. We did not test the `ss1` and `ss2` data sets since they are too huge. Note that the `ss1` data set contains 6 classes, while the other data sets have 2 classes. Table 1 describes the mean and standard deviation of the misclassification rate over repetitions. Since we had a numerical problem when computing LFDA, we slightly regularized it and consider SELF with $\beta = 0.001$ as LFDA. The fulfillment of the `cluster assumption` [3] is described as ‘CA’, which is the correct classification rate by the 1-nearest-neighbor classifier when both training and test labels are used for classifying all the training and test samples. Note that CA is computed $\int_0^r \text{CA}$ dimensionality reduction is applied, so it represents the fulfillment of the cluster assumption of the original data samples. The larger the value of CA is, the more reliable the cluster assumption would be (although the values are coarse).

² Even so, dimensionality reduction is still useful since a compact representation of the data can yield faster computation in the test phase.

Table 1. Misclassification rate for the SSL data sets. The numbers in the bracket are the standard deviation over repetitions. For each data set, the best method and comparable ones based on the *t*-test at the significance level 5% are described in bold face. ‘CA’ denotes the fulfillment of the cluster assumption. SELF(CV) denotes SELF with β chosen by cross validation. SELF’ denotes the combination of LFDA and LPP in a similar manner. The upper and lower halves of the table correspond to the cases with the number of labeled samples 100 and 10, respectively.

Data	CA	LFDA	SELF ($\beta = 0.5$)	PCA	SELF (CV)	LPP	SELF’ (CV)
SSL1	0.98	14.9(1.8)	6.0(1.3)	6.2(1.1)	6.0(1.4)	27.4(1.4)	28.4(2.6)
SSL2	0.97	15.7(0.9)	9.6(1.1)	11.2(0.8)	10.3(2.4)	24.1(2.2)	21.9(1.9)
SSL3	1.00	21.1(3.9)	14.3(1.8)	15.5(1.0)	14.1(1.4)	18.0(2.4)	18.5(2.4)
SSL4	0.58	33.4(3.5)	36.6(2.4)	48.7(2.4)	33.4(3.7)	46.7(1.7)	36.0(4.7)
SSL5	0.64	27.5(2.3)	27.2(2.3)	31.0(1.9)	27.3(2.9)	37.0(1.3)	35.3(1.9)
SSL6	0.98	38.1(1.5)	35.4(2.4)	27.3(2.7)	27.0(2.7)	35.2(1.7)	36.9(3.2)
SSL7	0.68	29.4(2.4)	29.1(2.4)	29.3(1.6)	27.7(1.4)	32.0(0.9)	32.8(1.5)
# Bests		2	5	2	7	0	1
SSL1	0.98	22.9(5.1)	26.3(6.1)	19.2(4.2)	22.3(5.4)	45.9(2.3)	48.5(2.4)
SSL2	0.97	22.3(3.0)	21.3(2.9)	25.8(4.2)	21.5(2.5)	31.2(7.5)	21.4(0.8)
SSL3	1.00	42.7(2.9)	42.9(3.0)	42.7(4.2)	43.6(3.2)	40.4(4.1)	41.0(5.2)
SSL4	0.58	47.3(2.9)	47.7(2.7)	49.9(2.2)	48.3(3.3)	49.5(2.5)	48.5(1.9)
SSL5	0.64	45.4(4.4)	45.4(4.4)	36.3(5.5)	40.2(6.9)	41.2(3.3)	44.5(3.6)
SSL6	0.98	67.7(4.6)	67.0(4.0)	67.7(4.1)	67.6(4.6)	71.4(4.0)	73.7(2.9)
SSL7	0.68	43.6(5.2)	43.6(5.2)	38.9(5.7)	40.1(7.1)	40.3(4.2)	42.7(5.3)
# Bests		5	4	5	6	3	4

Table 2. Misclassification rate for the IDA data sets. The upper and lower halves of the table correspond to the cases with the number of labeled samples 100 and 30, respectively.

Data	CA	LFDA	SELF ($\beta = 0.5$)	PCA	SELF (CV)	LPP	SELF’ (CV)
banana	0.87	27.0(2.6)	26.6(2.1)	26.4(1.9)	26.5(2.1)	26.4(1.9)	26.5(2.0)
b-cancer	0.68	34.5(4.4)	34.4(4.4)	34.4(4.1)	34.3(4.3)	34.8(4.0)	34.7(4.1)
diabetes	0.70	32.7(2.8)	33.0(2.7)	34.4(2.7)	33.0(2.7)	34.4(2.6)	33.2(2.7)
f-solar	0.63	39.5(5.1)	40.1(5.1)	40.1(5.2)	39.7(5.2)	39.7(5.4)	39.5(5.4)
german	0.69	31.2(2.9)	31.2(3.0)	33.7(2.8)	31.5(2.9)	33.7(2.6)	32.1(3.0)
heart	0.77	22.8(2.9)	22.6(2.8)	24.1(2.7)	23.1(2.8)	23.4(2.9)	23.1(2.8)
image	0.81	17.2(1.3)	18.8(1.3)	19.9(1.5)	17.8(1.7)	18.8(2.1)	16.6(1.3)
ringnorm	0.71	28.1(1.9)	28.9(1.9)	29.1(1.6)	28.1(1.8)	27.1(1.6)	27.6(1.8)
splice	0.71	29.9(3.5)	27.8(3.5)	30.8(2.3)	27.7(3.0)	42.1(1.9)	30.1(4.6)
thyroid	0.96	4.8(2.0)	5.3(2.1)	5.5(2.1)	5.0(1.9)	5.9(2.1)	5.1(2.0)
titanic	0.68	33.2(11.9)	33.2(11.9)	33.2(11.9)	33.2(11.9)	40.0(12.3)	37.4(12.5)
twonorm	0.94	4.8(1.3)	4.5(1.2)	4.1(1.1)	4.3(1.1)	4.0(1.0)	4.5(1.2)
waveform	0.85	15.5(1.4)	14.5(1.5)	14.1(1.4)	14.2(1.7)	13.8(1.4)	14.4(1.9)
# Bests		9	9	6	11	7	9
banana	0.87	31.1(4.0)	30.6(3.5)	30.0(4.1)	29.6(3.4)	30.0(4.1)	30.3(3.6)
b-cancer	0.67	36.1(6.4)	35.4(6.2)	36.1(6.3)	35.6(6.4)	36.1(5.8)	36.0(6.2)
diabetes	0.70	35.0(4.8)	34.7(4.3)	36.0(4.1)	34.9(4.4)	35.9(3.7)	35.1(4.2)
f-solar	0.63	41.5(5.5)	42.6(5.4)	42.7(5.1)	42.0(5.4)	40.6(5.3)	40.4(5.4)
german	0.69	36.6(4.7)	32.8(3.8)	35.6(4.1)	33.9(4.3)	36.0(4.0)	34.5(4.1)
heart	0.76	25.6(5.4)	23.7(4.9)	24.4(4.1)	24.6(4.7)	24.2(4.0)	24.9(4.2)
image	0.81	24.5(3.8)	26.2(3.2)	27.6(3.8)	26.0(3.8)	27.9(4.2)	24.5(3.5)
ringnorm	0.70	35.5(4.2)	34.0(3.7)	33.8(2.8)	33.1(3.2)	31.1(3.3)	32.5(3.8)
splice	0.71	34.0(3.1)	33.1(3.1)	34.6(2.5)	33.2(2.7)	45.2(2.5)	39.9(4.6)
thyroid	0.94	9.9(4.5)	8.3(4.1)	8.4(3.6)	8.7(4.2)	8.2(3.3)	8.9(4.2)
titanic	0.68	33.9(12.1)	34.0(12.2)	34.0(12.1)	33.9(12.1)	40.8(12.3)	37.5(12.9)
twonorm	0.94	15.3(6.5)	6.3(2.0)	4.3(1.3)	6.7(3.9)	4.2(1.3)	6.9(3.8)
waveform	0.85	27.5(4.3)	16.6(3.1)	15.6(2.3)	16.9(3.2)	15.3(2.2)	17.8(3.6)
# Bests		6	9	8	9	8	7

When the number of labeled samples is 100 (see the upper half of the table), LFDA and PCA tend to work well in a complementary way—LFDA works well if CA is small while PCA works well if CA is large. SELF with $\beta = 0.5$ tends to make up the deficit of each method; moreover it can outperform both LFDA and PCA for some cases. We also test ‘SELF(CV)’, where β in SELF is chosen from $\{0, 0.25, 0.5, 0.75, 1\}$ by 10-fold cross validation. The results shown in the table show that SELF(CV) further improves the performance over SELF with $\beta = 0.5$. LPP does not work so well on the whole. The combination of LFDA and LPP in a similar way (indicated by SELF’(CV) in the table) also does not perform as good as SELF(CV). We also tested the combination of LFDA, PCA, and LPP, but this did not further improve the performance over SELF so we omit the detail.

When the number of labeled samples is only 10 (see the lower half of Table II), the difference of the performance among the methods shrinks but SELF(CV) is still slightly better than the other methods.

We also conducted similar experiments using the [MNIST](#) data sets [\[6\]](#), where we randomly extracted labeled and unlabeled samples from the pool of all samples; we tested $n' = 100, 30$. The results are summarized in Table [2](#), showing that SELF(CV) still compares favorably with alternative methods.

Overall, SELFreg is shown to be a useful dimensionality reduction.

5 Conclusions and Future Prospects

Our approach to dimensionality reduction in this paper is called the *flexible* approach, i.e., the dimensionality reduction procedure is independent of subsequent classification algorithms. Our experimental results showed that the proposed method, [SELF](#), works well when it is combined with the 1-nearest-neighbor classifier. An important future direction is to develop a *flexible* method of semi-supervised dimensionality reduction, which explicitly takes properties of subsequent classification algorithms into account. We expect that a wrapper approach is promising in semi-supervised learning since the performance of elaborate semi-supervised learning methods is highly dependent on the reliability of the assumption behind unlabeled samples such as the cluster or manifold structure [\[3\]](#).

In this paper, we focused on linear dimensionality reduction. However, we can show that a non-linear variant of SELF is obtained by employing the standard [kernelized SELF](#). This kernelized variant also allows us to reduce the dimensionality of [structured data](#) such as strings, trees, and graphs [\[7\]](#). However, kernelized SELF shares the common difficulty in kernel methods, i.e., how to choose the kernel functions. This needs to be investigated in the context of semi-supervised dimensionality reduction. In the future work, we will also explore semi-supervised dimensionality reduction of structured data using kernel SELF.

A remaining important issue to be discussed—which is common to all semi-supervised learning techniques—is how to optimize tuning parameters. We may simply employ cross-validation for this purpose, but it has two potential problems. The first problem is that the number of labeled samples is typically small

in semi-supervised learning scenarios and thus cross-validation is not reliable [3]. Fortunately, our experiments showed that SELF is not so sensitive to the trade-off parameter β in small sample cases, but there is still room for further improvement. The second problem is that labeled samples and unlabeled samples can have different (input) distributions. Such a situation is referred to as *covariate shift* in statistics and ordinary cross-validation is known to be significantly biased; *self-cross-validation* is unbiased under covariate shift [8]. In the future work, we will investigate how the covariate shift adaptation techniques could be employed in the context of semi-supervised dimensionality reduction.

Finally, it is important to compare the performance of the proposed method with other related methods, e.g., [9,10].

The authors would like to thank members of T-PRIMAL (Tokyo PRobabilistic Inference and MACHine Learning) for their fruitful comments. MS acknowledges financial support from MEXT (Grant-in-Aid for Young Scientists 17700142 and Grant-in-Aid for Scientific Research (B) 18300057) and Tateishi Science and Technology Foundation.

References

1. Fukunaga, K.: Introduction to Statistical Pattern Recognition, 2nd edn. Academic Press, Inc., Boston (1990)
2. Sugiyama, M.: Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research* 8, 1027–1061 (2007)
3. Chapelle, O., Schölkopf, B., Zien, A. (eds.): *Semi-Supervised Learning*. MIT Press, Cambridge (2006)
4. He, X., Niyogi, P.: Locality preserving projections. In: *NIPS 16*, MIT Press, Cambridge (2004)
5. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: *NIPS 17*, pp. 1601–1608. MIT Press, Cambridge (2005)
6. Rätsch, G., Onoda, T., Müller, K.R.: Soft margins for adaboost. *Machine Learning* 42(3), 287–320 (2001)
7. Kashima, H., Koyanagi, T.: Kernels for semi-structured data. In: *Proceedings of ICML 2002*, pp. 291–298. Morgan Kaufmann, San Francisco (2002)
8. Sugiyama, M., Krauledat, M., Müller, K.R.: Covariate shift adaptation by importance weighted cross validation. *Journal of Machine Learning Research* 8, 985–1005 (2007)
9. Zhang, D., Zhou, Z.H., Chen, S.: Semi-supervised dimensionality reduction. In: *Proceedings of SDM 2007*, Minneapolis, MN, USA, pp. 629–634 (2008)
10. Cai, D., He, X., Han, J.: Semi-supervised discriminant analysis. In: *Proceedings of ICCV 2007*, Rio de Janeiro, Brazil (2008)

An Efficient Algorithm for Finding Similar Short Substrings from Large Scale String Data

Takeaki Uno

National Institute of Informatics
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
uno@nii.jp

Abstract. Finding similar substrings/substructures is a central task in analyzing huge amounts of string data such as genome sequences, web documents, log data, etc. In the sense of complexity theory, the existence of polynomial time algorithms for such problems is usually trivial since the number of substrings is bounded by the square of their lengths. However, straightforward algorithms do not work for practical huge databases because of their computation time of high degree order. This paper addresses the problems of finding pairs of strings with small Hamming distances from huge databases composed of short strings. By solving the problem for all the substrings of fixed length, we can efficiently find candidates of similar non-short substrings. We focus on the practical efficiency of algorithms, and propose an algorithm running in almost linear time of the database size. We prove that the computation time of its variant is bounded by linear of the database size when the length of short strings to be found is constant. Slight modifications of the algorithm adapt to the edit distance and mismatch tolerance computation. Computational experiments for genome sequences show the efficiency of the algorithm. An implementation is available at the author's homepage¹

1 Introduction

These days we have many huge string data such as genome sequences, web documents, log data, etc. Since the size of data is so huge that human cannot grasp them intuitively, they must be computationally analyzed. Finding similar substrings or similar substructures is an important way of analyzing the data. The similarity and distribution of substrings makes it possible to grasp the global or local structures. The number of substrings in a string is at most the square of the string length. Thus, if the distance between two substrings can be computed in polynomial time, similar substrings can be found in polynomial time by comparing all substrings one by one. However, polynomial time algorithms of high degree do not work for huge data, therefore practical fast algorithms are needed.

In the area of algorithms and computation, the problem of finding similar strings has been widely studied. The problem is usually formulated that for two given strings Q and S , find all substrings of S similar to Q . This formulation can

¹ <http://research.nii.ac.jp/~uno/index.html>

be considered as a generalization of string matching problems. When Hamming distance is chosen as a similarity measure, a straightforward algorithm solves the problem in $O(|S||Q|)$ time, thus a research goal is to reduce this time complexity. Here the length of S and Q is denoted by $|S|$ and $|Q|$.

For the problem of finding substrings of S with the shortest Hamming distance to Q , Abrahamson [1] proposed an algorithm running in $O(|S|(|Q| \log |Q|)^{1/2})$ time. If the maximum Hamming distance is k , the computation time can be reduced to $O(|S|(k \log k)^{1/2})$ [4]. Some approximation approaches have been also developed. The Hamming distance of two strings of length l within $(1-\epsilon)$ and $(1+\epsilon)$ approximation ratio with probability δ can be computed in $O(\log l \log(1/\delta)/\epsilon)$ time [6]. For edit distance, which allows insertions and deletions, algorithms proposed by Muthukrishnan and Sahinalp [8,9] approximate the minimum distance substring. Using these algorithms, the problem can be solved in shorter time but may fail with some solutions. These algorithms take more than $O(|S|^2)$ time to find similar substrings even for fixed length strings, Thus direct application of these algorithms does not work in practice.

On the other hand, there are several studies for efficient data structures to find similar substrings. The problem is formulated such that, for a given string S , construct a data structure of not a large size such that for any query string Q , substrings of S similar to Q can be found in short time. For the problem of finding substring of S equal to Q , there are many efficient data structures such as suffix array which make it possible to find all such substrings in almost $O(|Q|)$ time. However, allowing the errors makes the problem difficult. Existing algorithms basically need $\theta(|S|)$ time in the worst case. This difficulty can be observed in many other similarity search problems, such as inner product of vectors, points in Euclidean space, texts and documents. Motivated by practical use, there have been many studies on approximation and heuristic approaches.

Yamada and Morishita [12] proposed an algorithm for computing a lower bound of the shortest Hamming distance from Q to a substring in S . The algorithm constructs a data structure in $O(|S| \log |S|)$ time, then answers a lower bound in $O(|Q|L)$ time for any Q , where L is a constant no greater than $|Q|$. They also proposed an efficient exact algorithm for strings with small alphabet such as genome sequences [13].

In bioinformatics area, the problem of finding substrings of two strings which are similar to each other is called homology search, and has been widely studied. Because of the huge size of genome sequences, developing exact algorithms running in short time is difficult thus many heuristic algorithms have been proposed. BLAST and FASTA [2,3,10] are widely used among these algorithms. The idea of BLAST is to find short substrings of S and Q that are equal and check whether there are similar substrings including them. This idea is based on the observation that two similar substrings may have common short substrings. Actually, if the Hamming distance between two strings is no more than 9% of their length, they always have common string of 10 letters. The disadvantage of this method is that when the strings are long, huge number of substrings are the same, thus a lot of comparisons must be made. Such frequently appearing strings can be

considered as a kind of noise in practice, thus heuristic methods ignore these strings in the interest of practical efficiency. Another method of solving the problem is to partition Q and S into many blocks [11]. Some statistics of the blocks are computed, for example the number of each letter in the blocks, which for pruning blocks will never be similar. Then a dynamic programming connects the blocks and produces candidates of long similar substrings. The idea is that long similar substrings are expected to be not so many.

In this paper, we focus on Hamming distance. For given a set \mathcal{S} of strings of the same length l , our problem is to enumerate all pairs of similar strings in \mathcal{S} . We consider the case in which the length l is small, and propose a practically efficient algorithm. The idea of the algorithm is to classify the strings in several ways so that any two similar strings are in the same group for at least one classification. Only strings in the same group have to be compared, which reduces the cost of the comparison. Each string is partitioned into k blocks, then any two strings with Hamming distance at most d share at least $k - d$ blocks. Thus they are in the same group at least one classification based on combinations of $k - d$ of these blocks. By setting k to l , the Hamming distance of any two strings in the same group is at most d . Using this fact, the time complexity is bounded by $O((\sum_{i=0}^d \binom{l}{i} C_i) \times (|\mathcal{S}| + dN)) = O(2^l (|\mathcal{S}| + dN))$, where N is the number of pairs to be output. Computational experiments show its practical efficiency.

Using the algorithm makes it possible to approach the problem of finding similar non-short substrings. We can observe that two non-short similar strings may have several short substrings with short Hamming distance. Thus, pairs of substrings including several such strings are candidates for similar substrings. This approach has a certain accuracy. For example, any two strings of 3,000 letters with Hamming distance of at most 290 includes at least three substrings of 30 letters with Hamming distance of at most two. Similar observation can be made for edit distance. We propose an algorithm for finding representative pairs of non-short substrings including certain similar short substrings. We compared the human genome and mouse genome by our algorithm. The computation is done in quite short time and we could see the homology structure figured out by the comparison.

2 Preliminary

Let Σ be an alphabet of letters, and a $s_1 s_2 \dots s_l$ be a sequence of letters. The l -length of a string S is the number of letters in S and is denoted by $|S|$. A sequence composed of no letter is also a string and is called an empty string. The length of an empty string is 0. The i th letter of a string S is written $S[i]$, and i is called the i -index of $S[i]$. The substring of S starting from the i th letter and ending at the j th letter is denoted by $S[i, j]$. For example, when string S is $ABCDEFGG$, $S[3] = C$, and $S[4, 6] = DEF$. When $j < i$, we define $S[i, j]$ by the empty string. For two strings S_1 and S_2 , the $|S_2|$ -length of S_2 to S_1 is a string S given by concatenating S_2 to S_1 , i.e., $|S| = |S_1| + |S_2|$, $S[i] = S_1[i]$ if $i \leq |S_1|$, and $S_2[i - |S_1|]$ otherwise. The concatenation of S_2 to S_1 is denoted by $S_1 \cdot S_2$.

For two strings S_1 and S_2 of the same length, the Hamming distance of S_1 and S_2 is defined by the number of positions i satisfying that $S_1[i] \neq S_2[i]$. The Hamming distance is denoted by $HamDist(S_1, S_2)$. Such letters are called mismatches of S_1 and S_2 , and the positions of mismatches are called mismatch positions of S_1 and S_2 . For string S and integers i and k , $i \leq k$, we denote the substring of S starting from $(\lceil |S|(i-1)/k \rceil + 1)$ th letter to $(\lceil |S|i/k \rceil)$ th letter, i.e., $S[\lceil |S|(i-1)/k \rceil + 1, \lceil |S|i/k \rceil]$, by $B(S, k, i)$. $B(S, k, i)$ is called the i -th k -block of S .

For a string S , the i -th k -block of the position i is a string given by $S[1, i-1] \cdot S[i+1, |S|]$. The i -th k -block of letter a to S at position i is a string given by $S[1, i-1] \cdot A \cdot S[i, |S|]$ where A is the string composed of one letter a . The i -th k -block of position i of S to a is a string given by $S[1, i-1] \cdot A \cdot S[i+1, |S|]$. For two strings S_1 and S_2 , the k -block edit distance of S_1 and S_2 is the smallest number of combinations of insertion, deletion and change needed to transform S_1 to S_2 .

The problem we address in this paper is formulated as follows. Let \mathcal{S} be a multi set of strings of the same length. \mathcal{S} is allowed to include more than one same string, and every string has an ID to be distinguished from the others. The problem is formulated as follows.

Short Hamming Distance String Pair Enumeration Problem

Input: A multi set \mathcal{S} of strings of fixed length l , threshold value d
Output: All pairs of strings S_1 and S_2 such that $HamDist(S_1, S_2) \leq d$.

Hereafter we fix the input set \mathcal{S} of strings of length l and a threshold value d .

3 Multi-classification Algorithm

The basic idea of the algorithm is to classify the strings in several ways so that any two similar strings are in the same group at least once. Let $C(k, j)$ be the set of j distinct integers taken from $1, \dots, k$. For example, $C(4, 2) = \{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$. For a string S and a set $C = \{i_1, \dots, i_{k-d}\}$, $i_j < i_{j+1}$ taken from $C(k, k-d)$, we define $Sig(S, C) = B(S, k, i_1) \cdot B(S, k, i_2) \cdot \dots \cdot B(S, k, i_{k-d})$. We suppose that an integer k , $d < k \leq l$ is chosen, and have a look at the following property.

Lemma 1. $HamDist(S_1, S_2) \leq d \implies \exists C \in C(k, k-d)$ s.t. $Sig(S_1, C) = Sig(S_2, C)$

The statement is obvious from the pigeonhole principle. Suppose that $HamDist(S_1, S_2) \leq d$. Observe that if $B(S_1, k, j) \neq B(S_2, k, j)$ holds, it includes at least one mismatch, i.e., $S_1[i] \neq S_2[i]$ holds for some i , $\lceil |S|(i-1)/k \rceil + 1 \leq i \leq \lceil |S|i/k \rceil$. Since S_1 and S_2 have at most d mismatches, at most d integers j satisfy $B(S_1, k, j) \neq B(S_2, k, j)$, thereby at least $k-d$ integers h satisfy $B(S_1, k, h) = B(S_2, k, h)$. Setting C to the set of those integers h satisfying $B(S_1, k, h) = B(S_2, k, h)$ shows that $Sig(S_1, C) = Sig(S_2, C)$. \square

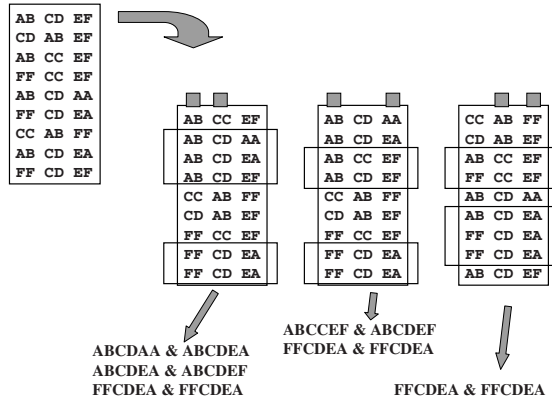


Fig. 1. Example of multi-classification for finding strings with Hamming distance of at most one, by dividing strings in three blocks and classifying them by two blocks

This lemma motivates us to restrict the comparison to those pairs of strings satisfying the condition of the lemma. To efficiently find these pairs, we focus on the combinations of integers. For each $C \in C(k, k - d)$, we classify the strings S in \mathcal{S} according to $Sig(S, C)$ so that two strings S_1 and S_2 satisfy $Sig(S_1, C) = Sig(S_2, C)$ if and only if they are in the same group. In Fig. 1, we show an example of this method, which we call the *multi-classification method*. In the example, there are nine strings and set $d = 1$ and $k = 3$. Each block is composed of two letters, and classifications by two blocks are done three times. For each classification there are several groups represented by rectangles with more than one strings, and some of them contain strings with Hamming distance of at most one, written at the head of the arrows.

ALGORITHM. MultiClassification_Basic (\mathcal{S} :set of strings of length l , d)

1. choose k from $d + 1, \dots, l$
2. **for each** $C \in C(k, k - d)$ **do**
3. classify all strings $S \in \mathcal{S}$ by $Sig(S, C)$
4. **for each** group K of the classification
 - output all pairs S_1 and S_2 in K satisfying $HamDist(S_1, S_2) \leq d$
6. **end for**

The classification for C is done by sorting $Sig(S, C)$ in $O(l(k - d)/k \times |\mathcal{S}|)$ time by a radix sort. We compute the probability that two randomly chosen letters from strings of \mathcal{S} are the same, and choose k such that the expected size of each group in a classification is less than 1. Then the comparisons for a group is not so many, and the bulk of the computation time is for radix sort. Since $l(k - d)/k$ is expected to be relatively small when l is small, it can be expected that the practical performance of the algorithm will be high.

3.1 Reducing the Cost for Radix Sort

Here we present a way to reduce the total computation time for radix sort by unifying the sort of the prefix of *Sig*. Suppose that we repeatedly and recursively add integers one by one to construct $C \in C(k, k - d)$ like a backtrack algorithm. Then, after choosing i in some iteration of the backtracking, $B(S, k, i)$ is common to all C generated in the recursive call, i.e., until i is removed. Thus, the radix sort for $B(S, k, i)$ can be done at the iteration and the result can be used in the recursive calls. As a result, the computation time for each radix sort is reduced to $O(l/k \times |\mathcal{S}|)$. We describe the algorithm in the next subsection.

3.2 Avoiding Duplication without Memory

The multi-classification described above may output duplicates, i.e., output one pair of strings many times. For example, in Fig. 1, the pair FFCDEA and FFCDEA is output three times. A way to avoid such duplication is to store all the pairs found in memory and check the duplication when a new pair is found. Although this is simple, it requires a lot of memory. Here, we present a method that does not store found pairs and thus requires no extra memory.

A pair of strings S_1 and S_2 is output more than once if $B(S_1, k, i) = B(S_2, k, i)$ holds more than $k - d$ integers i . Then, $Sig(S_1, C) = Sig(S_2, C)$ holds for many C 's. For given S_1 and S_2 , let $C^*(S_1, S_2)$ be the lexicographically minimum one among $\{C' | C' \in C(k, k - d), Sig(S_1, C') = Sig(S_2, C')\}$. Our idea is to output an S_1 and S_2 pair only when the current operating C is equal to $C^*(S_1, S_2)$. Since, $C^*(S_1, S_2)$ is the collection of the $k - d$ smallest i 's satisfying $B(S_1, k, i) = B(S_2, k, i)$, the computation is not a heavy task. The algorithm is the following which requires an initial call with \mathcal{S} , d and k , and set $C = \emptyset$.

ALGORITHM. MultiClassification (\mathcal{S} :set of strings of length l , d , k , C)

1. **if** $|C| = k - d$ **then** output all pairs S_1 and S_2 in K satisfying $HamDist(S_1, S_2) \leq d$ and $C = C^*(S_1, S_2)$; **return**
2. **for each** i larger than the maximum integer in C **do**
3. do a radix sort to classify all strings $S \in \mathcal{S}$ according to $B(S, k, i)$
4. **for each** group K of the classification with $|K| > 1$
 call MultiClassification ($K, d, k, C \cup \{i\}$)
5. **end for**

Theorem 1. MultiClassification (\mathcal{S} :set of strings of length l , d , k , C) runs in $O(l/k \times |\mathcal{S}| \times {}_l C_d)$

3.3 A Fixed Parameter Tractable Algorithm

The time complexity of the algorithm presented in the previous subsection is still $O(|\mathcal{S}|^2)$ since the bottle neck of the computation is actually step 1. For example, if all strings in \mathcal{S} are the same, $HamDist(S_1, S_2)$ must be computed ${}_l C_d$ times for every S_1 and S_2 pair in \mathcal{S} , thereby the total computation time is $O(l|\mathcal{S}|(|\mathcal{S}| + {}_l C_d))$. Here we will save the computation time in step 1.

Let $k = l$. Then, for each i , $B(S, k, i)$ is composed of one letter, thus $Sig(S_1, C) = Sig(S_2, C)$ immediately means $HamDist(S_1, S_2) \leq d$. This implies that the Hamming distance does not have to be computed for any pair in each group. Another task in step 1 is avoiding duplications. We do this in another way.

Duplicate outputs occur when $HamDist(S_1, S_2)$ is strictly smaller than d . If $HamDist(S_1, S_2) = d$, exactly one $C \in C(k, k - d)$ satisfies $Sig(S_1, C) = Sig(S_2, C)$. This implies that without any check, we can output pairs with Hamming distance equal to d without duplications. Thus, we change d' from 0 to d and output only pairs with Hamming distance equal to d' , we need no check for duplications. We call this algorithm the *multi-classification algorithm*. For the complete version of our algorithm, we obtain the following theorem. Note that the computation of $HamDist(S_1, S_2)$ is done in $O(d)$ time if $Sig(S_1, C) = Sig(S_2, C)$.

Theorem 2. Let S be a string of length $|S|$ and N be a set of strings of length l . For a given distance threshold d , the number of pairs (S_i, S_j) such that $HamDist(S_i, S_j) \leq d$ is $O((\sum_{i=0}^d \binom{l}{i} C_i) \times (|S| + dN)) = O(2^l (|S| + dN))$.

4 Approach to Long Substrings

In this section, we consider the problem of finding all pairs of substrings of a given string S that are similar to each other in some sense. In the sense of time complexity, the existence of polynomial time algorithms for this kind of problem is trivial since we have to compare only a polynomial number of pairs. However, in a practical sense, this problem is difficult since even if we restrict the pairs to be strings of the same length, $O(|S|^3)$ pairs of substrings must be compared. For huge strings the computation time must be quasi linear time, thus $O(|S|^3)$ time is far from practical efficiency.

Here we approach this problem with our algorithm. For a string S , distance threshold value d and length l , a pair of positions $(p, q), p \neq q$ is an $l-d$ seed if $HamDist(S[p, p + l - 1], S[q, q + l - 1]) \leq d$. We can find all $l-d$ seeds by giving all the substrings of S of length l to our multi-classification algorithm. One typical approach to capturing the similarity structures by using such seeds is as follows. We partition S into non-short blocks, for example, partition a string of 1,000,000 letters into 1,000 strings of 1,000 letters. We define the similarity measure of blocks $S[k_1, h_1]$ and $S[k_2, h_2]$ by the number of $l-d$ seeds (p, q) satisfying $k_1 \leq p \leq h_1$ and $k_2 \leq q \leq h_2$. We can visualize the similarity structure of this measure by a figure such that the intensity of the color of the pixel (x, y) is given by the number of $l-d$ seeds in x th block and y th block. The left of Figure 2 shows an example of pictures obtained by this method. If the blocks are large, any two blocks have a sufficiently large number of seeds, thus all pixels will be the same color. For large scale data, we need more precise method of deleting such noise.

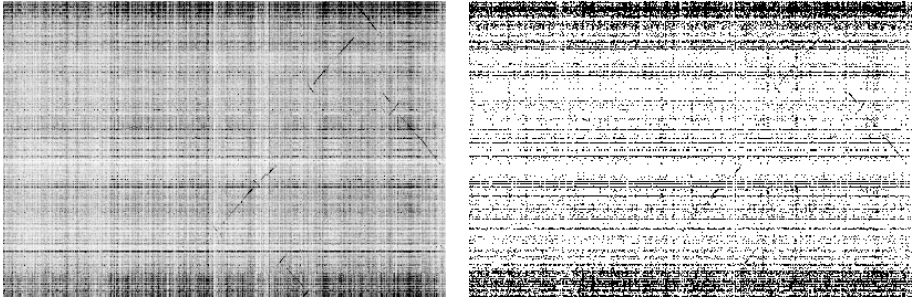


Fig. 2. Matrix showing similarity of mouse 11 chromosomes (X-axis) and Human 17 chromosome (Y-axis), with black cells on similar parts; we can see similar substructures as diagonal lines, but the figure is noisy because of the low resolution

A pair of positions $(p, q), p \neq q$ is a (w, c, l, d) candidate if (p, q) is an l - d seed and p is a multiple of l . The normal l - d seeds can also be enumerated by our algorithm with shorter time than the usual l - d seeds. For a width threshold w and count threshold c , we say the pair of substrings $S[k_1, h_1]$ and $S[k_2, h_2]$ is a normal (w, c, l, d) candidate if there are distinct c normal l - d seeds (p, q) satisfying $k_1 \leq p \leq h_1, k_2 \leq q \leq h_2$ and $|p - q| \leq w$. A pair of similar substrings can be considered to be a normal (w, c, l, d) candidate for non-trivial w, c, l , and d . Especially if the Hamming distance of two substrings is short, they must be a normal (w, c, l, d) candidate for a certain (w, c, l, d) . For example, if the Hamming distance of two substrings of length 3000 is at most 290, they have to be a normal $(0, 3, 30, 2)$ candidate. If the edit distance of two substrings of length 3000 is at most 190 and has at most 50 of insertions and deletions, then they have to be a normal $(50, 3, 30, 2)$ candidate. Thus, we are motivated to enumerate all normal (w, c, l, d) candidates. However, for a set of c normal l - d seeds, there would be many normal (w, c, l, d) candidates including these seeds. Thus, the number of enumerated candidates can be large. Recall that the aim here is to find candidates of similar substrings, or to capture the similarity structures. Not many similar candidates are needed to represent one similar structure. Thus, here we propose a simple algorithm to output a set of pairs of substrings such that any normal (w, c, l, d) candidate is obtained by a slight modification of one of the pair.

For an integer z , we consider a slit of width $2w$. An l - d seed (p, q) is included in the slit of z if $z \leq p - q \leq z + 2w$. For each multiple z of w , we find all integers i such that there are at least c l - d seeds (p, q) included in the slit of z such that $i \leq p + q < i + a$ where a is a given length, and one of them satisfies $i = p + q$. For such integers i , the pair of substrings $S[(i + z + w)/2, (i + z + w)/2 + a]$ and $S[(i - z - w)/2, (i - z - w)/2 + a]$ is a desired pair. We output all such pairs. This requires sorting of all l - d seeds, but remaining process is very light and simple. We display a figure made by this approach in the right of Figure 2.

5 Applications and Extensions

In the practical applications there are many variants of similar string finding problems. In the following subsections we present several problems to which we can apply our multi classification algorithm.

5.1 Computing Mismatch Tolerance

In real world applications, we often need to find several unique short strings which are similar to no other strings. Such unique strings can be used as characterizations, invariants of string databases, or markers of substructures. A typical application is in microarray. A microarray is a tool for biological experiments that can detect the existence of short strings, say 25 letters, in the genome sequence of a species or organizations. If a unique short substring in a gene sequence is known, the existence of the substring indicates the existence of the gene. To allow for experimental error, the substring has to have no similar substring.

When the Hamming distance is used, one of the uniqueness measure is called mismatch tolerance. The mismatch tolerance is the shortest Hamming distance to the other string. More precisely, for a set \mathcal{S} of strings of the same length l , the mismatch tolerance of string S , denoted by $mis(S, \mathcal{S})$ is defined by $\min\{HamDist(S, S') \mid S' \in \mathcal{S} \setminus \{S\}\}$. If $mis(S, \mathcal{S})$ is large, S has no similar string in \mathcal{S} in the sense of Hamming distance, thus our aim is to find the strings having not so small mismatch tolerance. Here we define our problem.

All Mismatch Tolerance Computing Problem

Input: for a set \mathcal{S} of strings of the same length l , distance threshold d

Output: all $S \in \mathcal{S}$ such that $mis(S, \mathcal{S}) \leq d$

This problem can be solved by solving the short Hamming distance string pair enumeration problem. Actually, we do not have to output pairs, thus we do not check the duplications. Moreover, in the complete version of our algorithm, we have to execute the algorithm only for $d' = d$, and omit the computation of Hamming distance. Thus we obtain the following theorem.

Theorem 3. For a set \mathcal{S} of strings of the same length l , distance threshold d , $O(tC_d|\mathcal{S}|) = O(2^l|\mathcal{S}|)$.

5.2 General Edit Distance

In many studies and real world applications, the distance between two strings, genomes, and documents is evaluated by edit distance. The multi classification algorithm proposed above fails for edit distance since the position of the block shifts by the preceding insertions and deletions. For example, the edit distance between $S_1 = \text{ABCDEFGHG}$ and $S_2 = \text{ACDEFGHI}$ is 2, obtained by deleting the second letter of S_1 and the eighth letter of S_2 . By setting $k = 4$, the strings

are partitioned into substrings of two letters. Although there are only two positions edited, no substrings in the partitions of S_1 and S_2 are the same, since the substrings in the middle are shifted by the deletion of the second letter.

For adapting to edit distance, we consider $\hat{C}(k, d)$ instead of $C(k, k-d)$ where $\hat{C}(k, d)$ is the set of $k-d$ signed or unsigned integers taken from 1 to d , i.e., $\hat{C}(k, d) = \{C \mid |C| = d, C \subseteq \{1, 1^+, 1^-, 2, 2^+, 2^-, \dots, k, k^+, k^-\}\}$. i^+ , i^- and i means an insertion, a deletion and a change at the i th block. For $C \in \hat{C}(k, d)$, let $\text{sft}(C, i) = |\{j^+ \mid j < i, j^+ \in C\}| - |\{j^- \mid j < i, j^- \in C\}|$, and $\text{Eq}(C) = \{i \mid i, i^+, i^- \notin C\}$. We denote $S[\lceil |S|(i-1)/k \rceil + 1 + j, \lceil |S|i/k \rceil + j]$ by $\hat{B}(S, i, j)$. Then, for string S and $C \in \hat{C}(k, d)$, we define $\hat{S}ig(S, C)$ by $\hat{B}(S, i_1, \text{sft}(C, i_1)) \cdot \hat{B}(S, i_2, \text{sft}(C, i_2)) \cdot \dots \cdot \hat{B}(S, i_{k-d}, \text{sft}(C, i_{k-d}))$ where $\text{Eq}(C) = \{i_1, \dots, i_{k-d}\}, i_j < i_{j+1}$. By using the terminology, we obtain the following lemma.

Lemma 2. $C \in \hat{C}(k, d) \implies \hat{S}ig(S_1, \text{Eq}(C)) = \hat{S}ig(S_2, C)$

The proof is omitted by the page limit. Based on the lemma, we are motivated to classify all strings by $\hat{S}ig(S, \text{Eq}(C))$ and $\hat{S}ig(S, C)$ for all $C \in \hat{C}(k, d)$ to obtain all the pairs of strings satisfying the condition of the lemma. By checking the edit distance for all pairs in each group classified, we can find all pairs of strings with edit distance at most d .

Theorem 4. $O(3^d l/k \times |S| \times_l C_d)$

Theorem 5. $O(2^l 3^d (|S| + l^2 N)) \times (\sum_{i=0}^d l C_i) \times (|S| + l^2 N)$

6 Computational Experiments

This section shows the results of computational experiments of the basic version of our algorithm. The code was written in C, and compiled with gcc. We used a note PC with a Pentium M 1.2GHz processor with 768 MB of memory, with cygwin which is a Linux emulator on Windows. The implementation is available at the author’s homepage; <http://research.nii.ac.jp/~uno/index.html>.

The instance is the set of substrings of fixed length taken from the Y chromosome of Homo sapiens. The length is set to 20, 50 and 300. Figure 3 shows the results. Each line corresponds to one threshold value d . The X-axis is the number of input substrings, and Y-axis is the computation time. Both axes use log scales. We can see that the computation time increases slightly higher than linear, but smaller than the square. Figure 4 shows the number of executed radix sorts. The number increases drastically if the number of mismatches increases, but does not increase much as the increase of input size.

We also show the increase in computation time against the increase of l with fixed d/l . The instance is fixed to that with 2.1 million strings, and the result is shown in the right-lower figure of Figure 3. From these results, at least for genome sequences our algorithm is quite scalable for the increase of input string.

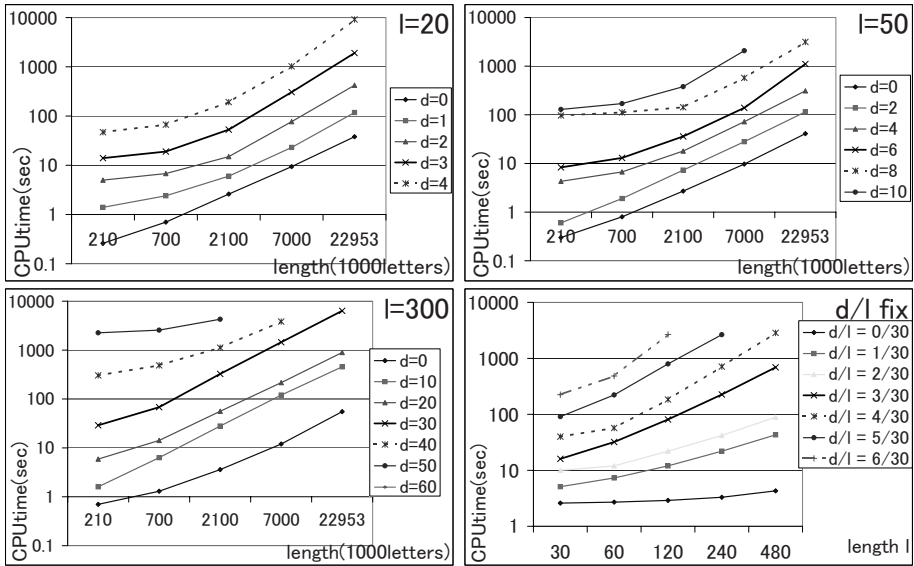


Fig. 3. Increase in computation time against the increase in database size with fixed l and d : the right-lower figure is for fixed d/l inputting a string of 2.1 million letters

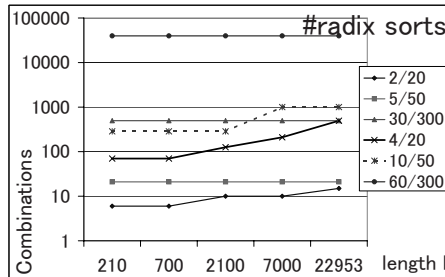


Fig. 4. Number of radix sorts performed

7 Conclusion

We proposed an efficient algorithm for enumerating all pairs of strings with Hamming distance at most given d from string set \mathcal{S} . We focused on the practical efficiency of algorithms, and proposed an algorithm based on multiple classifications according to combinations of blocks of each string. We proved that the computation time of its variant is bounded by linear of the string length when the length of strings in the string set is constant. A simple modification of the algorithm adapts the edit distance, and computation of mismatch tolerance.

We also proposed a method of finding similar non-short substrings from huge strings. We modeled similar non-short strings by two non-short strings including

several short similar substrings. We presented an efficient algorithm for finding those strings from huge strings. By the computational experiments for genome sequences, we demonstrated the practical efficiency of the algorithm. On the comparison of genome sequences, we could find similar long substrings from human and mouse genomes in a practically short time.

Acknowledgments

We gratefully thank to Professor Asao Fujiyama of National Institute of Informatics of Japan, Professor Shinichi Morishita of Tokyo University Doctor Takehiko Itoh of Mitsubishi Research Institute, and Professor Hidemi Watanabe of Hokkaido University, for their valuable comments. We would also like to thank to Professor Tsuyoshi Koide and Doctor Juzo Umemori of National Institute of Genetics for their contribution to the evaluation of the algorithm on practical genome problems.

References

1. Abrahamson, K.: Generalized String Matching. *SIAM J. on Comp.* 16(6), 1039–1051 (1987)
2. Altschul, S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J.: Basic local alignment search tool. *J. Mol. Biol.* 215, 403–410 (1990)
3. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, Z.W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* 25, 3389–3402 (1997)
4. Amir, A., Lewenstein, M., Porat, E.: Faster Algorithms for String Matching with k Mismatches. In: *Symposium on Disc. Alg.*, pp. 794–803 (2000)
5. Brown, P., Botstein, D.: Exploring the New World of the Genome with DNA Microarrays. *Nature Genetics* 21, 33–37 (2000)
6. Feigenbaum, J., Kannan, S., Strauss, M., Viswanathan, M.: An Approximate l_1 -difference Algorithm for Massive Data Streams. In: *Proc. FOCS 1999* (1999)
7. Manber, U., Myers, G.: Suffix Arrays: A New Method for On-line String Searches. *SIAM J. on Comp.* 22, 935–948 (1993)
8. Muthukrishnan, S., Sahinalp, S.C.: Approximate Nearest Neighbors and Sequence Comparison with Block Operations. In: *Proc. 32nd annual ACM symposium on Theory of Computing*, pp. 416–424 (2000)
9. Muthukrishnan, S., Sahinalp, S.C.: Simple and Practical Sequence Nearest Neighbors under Block Edit Operations. In: Apostolico, A., Takeda, M. (eds.) *CPM 2002. LNCS*, vol. 2373, Springer, Heidelberg (2002)
10. Pearson, W.R.: Flexible sequence similarity searching with the FASTA3 program package. *Methods in Molecular Biology* 132, 185–219 (2000)
11. Yamada, S., Gotoh, O., Yamana, H.: Improvement in Accuracy of Multiple Sequence Alignment Using Novel Group-to-group Sequence Alignment Algorithm with Piecewise Linear Gap Cost. *BMC Bioinformatics* 7, 524 (2006)
12. Yamada, T., Morishita, S.: Computing Highly Specific and Mismatch Tolerant Oligomers Efficiently. In: *Bioinformatics Conference* (2003)
13. Yamada, T., Morishita, S.: Accelerated Off-target Search Algorithm for siRNA. *Bioinformatics* 21, 1316–1324 (2005)

Ambiguous Frequent Itemset Mining and Polynomial Delay Enumeration

Takeaki Uno¹ and Hiroki Arimura²

¹ National Institute of Informatics,
2-1-2, Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan
uno@nii.jp

² Graduate School of Information Science and Technology, Hokkaido University, Kita
14 Nishi 9, Sapporo 060-0814, Japan
arim@ist.hokudai.ac.jp

Abstract. Mining frequently appearing patterns in a database is a basic problem in recent informatics, especially in data mining. Particularly, when the input database is a collection of subsets of an itemset, called transaction, the problem is called the frequent itemset mining problem, and it has been extensively studied. The items in a frequent itemset appear in many records simultaneously, thus they can be considered to be a cluster with respect to these records. However, in this sense, the condition that every item appears in each record is quite strong. We should allow for several missing items in these records. In this paper, we approach this problem from the algorithm theory, and consider the model that can be solved efficiently and possibly valuable in practice. We introduce ambiguous frequent itemsets which allow missing items in their occurrence records. More precisely, for given thresholds θ and σ , an ambiguous frequent itemset P has a transaction set \mathcal{T} , $|\mathcal{T}| \geq \sigma$, such that on average, transactions in \mathcal{T} include ratio θ of items of P . We formulate the problem of enumerating ambiguous frequent itemsets, and propose an efficient polynomial delay polynomial space algorithm. The practical performance is evaluated by computational experiments. Our algorithm can be naturally extended to the weighted version of the problem. The weighted version is a natural extension of the ordinary frequent itemset to weighted transaction databases, and is equivalent to finding submatrices with large average weights in their cells. An implementation is available at the author's homepage¹.

1 Introduction

The frequent pattern mining problem is to find patterns frequently appearing in a given database. It is one of the central tasks in data mining, and has been a focus of recent informatics studies. Particularly, when the database \mathcal{D} is a collection of \mathcal{T} , where a transaction is a subset of an $I = \{1, \dots, n\}$, and

¹ <http://research.nii.ac.jp/~uno/index.html>

² In the literature, a transaction is often defined by a pair of an item subset and its ID. However, we omit the ID since it has no effect on the arguments in this paper.

the patterns to be found are also subsets of itemsets, the problem is called the [frequent pattern mining](#) [[14](#),[11](#),[2](#)].

Precisely, a transaction of \mathcal{D} including P is called an occurrence of P , and the set of occurrences of P is denoted by $Occ(P)$. We define the frequency of an itemset by $|Occ(P)|$, and say an itemset is a frequent itemset if its frequency is no less than the given threshold value σ , called the support threshold. The frequency is often called the support, and σ is called the support threshold.

Frequent pattern mining is often used for data analysis. For data so huge that humans can not get any intuition from an overview of it, the frequent pattern mining is a useful way to capture the features of the data's features, both in a global sense and in a local sense. However, we often encounter difficulties in trying to use the frequent pattern mining on real-world data. One difficulty is that data is often incorrect or has missing parts. Such errors mean that some records that should include a pattern P do not include it, thus P may be overlooked because its frequency appears to be too low. A way to deal with this difficulty is to consider an ambiguous inclusion relation whereby we consider that a transaction T includes a pattern P if most items of P are included in T .

There are several studies on the frequent pattern mining with ambiguous inclusions. In some contexts, these patterns are called fault-tolerant frequent itemsets [[5](#),[7](#),[8](#),[9](#),[16](#)]. In some of these studies [[16](#)], ambiguous inclusion is defined such that an itemset P is included in a transaction T if $|P \cap T|/|P| \geq \theta$. Given this definition, the family of frequent itemsets is not always anti-monotone, thus the usual apriori based algorithms are not output sensitive in the sense of time complexity. On the other hand, the authors introduced an inclusion allowing a constant number of missing items, i.e., $|P \setminus T| \leq \theta$. This does not violate the monotonicity, thereby admits both apriori and backtrack with many related techniques developed for frequent itemset mining. However, it has a disadvantage that a transaction can miss only few items of large itemsets whereas almost all small itemsets will be frequent.

In some studies [[5](#),[7](#),[8](#),[9](#)], they considered that it is a fault if an item of the itemset is not included in a transaction, and treat mining pairs of an itemset and a transaction set such that there are few faults between their elements. Their algorithms find pairs with few faults, but they are not always minimum solutions.

In this paper, we address the problem from the algorithmic point of view, and model the problem in a different way. In the other words, the goal of this paper is to investigate the most simple and useful model of ambiguous frequency which allows fast computation. In the existing practice-based approach, the designed model often allows no fast algorithm. Heuristic approaches lose the completeness and exactness of the algorithm. For developing fast algorithms, we consider another model for ambiguous frequency.

For an itemset P and a transaction T , the inclusion ratio of items of P included in T , i.e., $|T \cap P|/|P|$. For an itemset P and a transaction set \mathcal{T} , the average inclusion ratio of \mathcal{T} for P is defined by the average of the inclusion ratio of transactions in \mathcal{T} , i.e., $(\sum_{T \in \mathcal{T}} |T \cap P|)/(|\mathcal{T}||P|)$. By representing the inclusion between transactions and items by matrix, the average inclusion

transaction database D

A: 1,2,4
B: 1,2
C: 1,3
D: 2,3

Inclusion ratio of A for itemset {1,3,4,5} = 1/2
 inclusion ratio of B for itemset {1} = 1
 average inclusion ratio of {A,B,C} for itemset {1} = 1
 for density threshold $\theta = 0.65$,
 $AmbiOcc(\{1,3\}) = \{A,B,C\}$
 for density threshold $\theta = 0.65$,
 a maximum co-occurrence set of {1,2,3} = {A,B,C,D}

Fig. 1. Examples of average inclusion ratio and maximum occurrence sets

ratio corresponds to the density of the submatrix induced by the items and transactions. When the average inclusion ratio is high, the items co-occur in the transactions, or the transactions co-occur in the items. For a density threshold θ , a transaction set of the largest size having average inclusion ratio no less than θ is called the *maximum co-occurrence set* for P . Note that any maximum co-occurrence set can be obtained by choosing transactions in decreasing order of their inclusion ratio. The size of the maximum co-occurrence set is called the *co-occurrence number* of P , and is denoted by $cov(P)$. We denote the lexicographical minimum maximum co-occurrence set by $AmbiOcc(P)$. Some examples are shown in Figure 1.

For the minimum support threshold σ , an itemset is called an *ambiguous frequent itemset* if its maximum co-occurrence size is no less than σ . The problem in this paper is formulated as follows.

Ambiguous Frequent Itemset Enumeration Problem

Input: transaction database \mathcal{D} , minimum support σ , density threshold θ

Output: all ambiguous frequent itemsets in \mathcal{D}

We propose a polynomial delay polynomial space algorithm, and show the practical performance by computational experiments. Note that an algorithm is *polynomial delay* if the computation time between any two consecutive solutions is polynomial in the input size.

If we represent the inclusion relation by a bipartite graph, an ambiguous frequent itemset and its corresponding transaction set corresponds to a vertex set inducing a dense bipartite subgraph, which is a quasi bipartite clique. Enumerating dense subgraphs whose edge density is no less than a threshold value can be done in polynomial delay polynomial space [14]. However, since an ambiguous frequent itemset has a lower bound for transaction sets, and identifies the same itemsets with different transaction sets, a direct application of the algorithm to our problem is not polynomial delay.

The existence of a polynomial delay algorithm for the enumeration problem of ambiguous frequent itemset is not trivial, since as we will show, simple algorithms involve an NP-complete problem in each iteration. The framework of the algorithm in this paper is motivated from the enumeration algorithm for pseudo cliques [14]. We introduce an adjacency relation with respect to a removal of an item between ambiguous frequent itemsets, and implicitly construct

a tree-shaped traversal route on the relation. Our algorithm searches traverses the tree in a depth-first search manner, so that the computation time is polynomial delay. To best of our knowledge, this is the first result of even an output polynomial time algorithm for this problem. The ambiguous frequency and our algorithm can be naturally extended to a weighted version, in a straightforward manner.

2 Polynomial Delay Algorithm

The frequent itemset enumeration problem is, from the viewpoint of complexity theory, an easy problem. The reason is that the frequency has a monotone property, thus obviously any frequent itemset can be obtained by iteratively adding items to the emptyset by passing through only frequent itemsets. The repeated addition admits any ordering of items, hence we can efficiently avoid the duplications by adding items only in increasing order of their indices. Thus, we can construct a backtrack algorithm of a polynomial delay polynomial space. Precisely, the computation time for each frequent itemset is linear in the size of the database, i.e., $O(|\mathcal{D}|)$, where $|\mathcal{D}|$ is the size of database \mathcal{D} , i.e., $|\mathcal{D}| = |\mathcal{I}| + \sum_{T \in \mathcal{D}} |T|$. The space complexity is optimal, that is, $O(|\mathcal{D}|)$.

However, the family of ambiguous frequent itemsets does not have this monotone property. For the database \mathcal{D} in Figure 1, $\theta = 65\%$ and $\sigma = 4$, we can see that $cov(\{1, 2, 3\}) = 4$, as obtained by transaction set $\{A, B, C, D\}$, thereby $\{1, 2, 3\}$ is an ambiguous frequent itemset. However, the maximum co-occurrence size of its subset $\{1, 3\}$ is 3, obtained by transaction set $\{A, B, C\}$, thereby $\{1, 3\}$ is not an ambiguous frequent itemset. Since the monotonicity is not held, a straightforward backtrack algorithm is not applicable to the enumeration.

We approach the problem in another way. For itemset $P \neq \emptyset$, we define $e^*(P)$ by the item $e \in P$ that minimizes $|AmbiOcc(P) \cap Occ(\{e\})|$. Ties are broken by choosing the minimum index one. Using e^* , we introduce an adjacency relation among ambiguous frequent itemsets, and construct an implicit traversal route.

Lemma 1. *Let $P \neq \emptyset$ be an ambiguous frequent itemset. Then, $cov(P \setminus \{e^*(P)\}) \geq cov(P)$.*

First we observe that the average inclusion ratio of $AmbiOcc(P)$ for P is given by the average of $|AmbiOcc(P) \cap Occ(\{e\})|/|AmbiOcc(P)|$, since the average inclusion ratio of $AmbiOcc(P)$ for P is $\frac{\sum_{e \in P} |AmbiOcc(P) \cap Occ(\{e\})|}{(|P|-1) \times |AmbiOcc(P)|}$. From the observation, the average inclusion ratio of $AmbiOcc(P)$ for $P \setminus \{e^*(P)\}$ is the average of $|AmbiOcc(P) \cap Occ(\{e\})|/|AmbiOcc(P)|$ among $P \setminus \{e^*(P)\}$, thus it is no less than the average inclusion ratio of $AmbiOcc(P)$ for P . It means that $cov(P \setminus \{e^*(P)\})$ is no less than $cov(P)$, thus $e^*(P)$ satisfies the condition to be the item e in the statement. \square

For an itemset $P \neq \emptyset$, we define the parent $Prt(P)$ of P by $P \setminus \{e^*(P)\}$. From Lemma 1, $P \setminus \{e^*(P)\}$ is an ambiguous frequent itemset. Particularly, $cov(Prt(P)) \leq cov(P)$. The cardinality of $Prt(P)$ is exactly one smaller than P ,

$\theta = 66\%, \sigma = 4$

- A: 1,3,4,7
- B: 2,4,5,
- C: 1,2,7
- D: 1,4,5,7
- E: 2,3,6
- F: 3,4,6

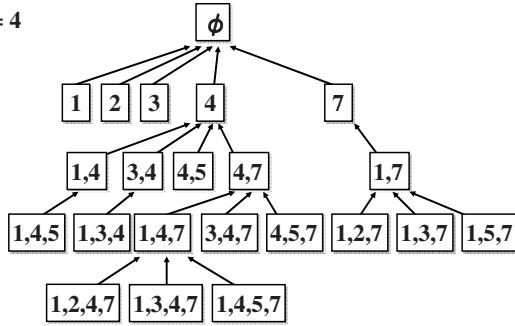


Fig. 2. Example of an enumeration tree

thus the parent child relation induced by Prt is acyclic. Since every ambiguous frequent itemset other than the emptyset has a parent, the relation induces a rooted tree spanning all ambiguous frequent itemsets. We call this tree the Prt -tree of ambiguous frequent itemsets. An example of the enumeration tree is shown in Figure 2. By traversing the tree, we can find all ambiguous frequent itemsets without duplication.

To perform a depth-first search on the enumeration tree, we need to find all children of the current visiting ambiguous frequent itemset. By recursively finding the children, we can perform a depth-first search without using additional memory on visited vertices. This ensures the polynomiality of the memory complexity. The algorithm can be written as follows.

ReverseSearch(P)

1. Output P
2. for each $e \notin P$
 - 2-1. if $P \cup \{e\}$ is an ambiguous frequent itemset then
 - 2-2. if $Prt(P \cup \{e\}) = P$ then
 - 2-3. call ReverseSearch ($P \cup \{e\}$)

The computation of the average inclusion ratio and the parent of $P \cup \{e\}$ in 2-1 and 2-2 can be done in $O(\|\mathcal{D}\|)$ time. They are executed at most n times in an iteration, thus the computation in an iteration except for those in the recursive calls is bounded by $O(\|\mathcal{D}\| \times n)$. This algorithm outputs an ambiguous frequent itemset in each iteration, thus the computation time per ambiguous frequent itemset is $O(\|\mathcal{D}\| \times n)$. The depth of the enumeration tree is bounded by n , thus we obtain the following theorem.

Theorem 1. Let \mathcal{D} be a database with n items and m transactions. Let θ and σ be the minimum support and the minimum average inclusion ratio, respectively. Then the number of ambiguous frequent itemsets is $O(n\|\mathcal{D}\|)$.

3 Improvements for Efficient Practical Computation

For practical huge databases, the computation time $O(|\mathcal{D}| \times n)$ is quite long. It is not easy to reduce the time complexity, but possible to improve the practical efficiency by using at typical structures of actual datasets. The heavy tasks in each iteration with respect to itemset P is the computation of $cov(P \cup \{e\})$ and $e^*(P \cup \{e\})$ for each e . Both need $O(n|\mathcal{D}|)$ time, and we will describe techniques to reduce the computation time for each. Note that the computation of $cov(P)$ is maybe heavier since we have to compute $e^*(P \cup \{e\})$ only when $P \cup \{e\}$ is an ambiguous frequent itemset.

We define $Occ_{=h}(P)$ by the set of transactions not including exactly h items in P , i.e., $Occ_{=h}(P) = \{T \mid T \in \mathcal{D}, |P \setminus T| = h\}$. Similarly, $Occ_{\leq h}(P) = \{T \mid T \in \mathcal{D}, |P \setminus T| \leq h\}$. For the computation of $cov(P \cup \{e\})$, we have to obtain $Occ_{=h}(P \cup \{e\})$ for each e and h , in increasing order of h . We can use the following property and lemma for efficient computation.

Prop. 13 For a transaction T included in $Occ_{=h}(P)$ for some $h, 0 \leq h \leq k$, $T \in Occ_{=h}(P \cup \{i\})$ holds if T includes i . Otherwise, $T \in Occ_{=h+1}(P \cup \{i\})$.

Lemma 2. $Occ_{=0}(P \cup \{i\}) = Occ_{=0}(P) \cap Occ(\{i\})$, $Occ_{=h}(P \cup \{i\}) = (Occ_{=h}(P) \cap Occ(\{i\})) \cup (Occ_{=h-1}(P) \setminus Occ(\{i\}))$, $h \geq 1$

From these, we can see that $Occ_{=h}(P \cup \{e\})$ for all h are obtained by moving transactions of $Occ_{=h}(P) \cap Occ(\{e\})$ to $Occ_{=h+1}(P)$. This takes $O(|Occ(\{e\})|)$ time, which is expected to be small when the input database is sparse. To compute $Occ_{=h}(P) \cap Occ(\{e\})$, a method called [Delivery](#) is efficient [\[11\]\[12\]\[13\]](#).

We briefly explain the framework of Delivery. An example is shown in Fig. [3](#). First, we prepare an empty bucket for each item e . Next, for each transaction T in $Occ_{=h}(P)$, we “insert T into the bucket of e for each item $e \in T$ ”. After performing this operation for all transactions in $Occ_{=h}(P)$, the content of the bucket of e is equal to $Occ_{=h}(P \cup \{e\})$. The pseudo code of occurrence deliver is described as follows. The code inputs a transaction set \mathcal{S} , then sets $bucket[e]$ to $\mathcal{S} \cap Occ(\{e\})$ for all e . We suppose that the bucket of any item e is initialized, and thus is empty at the beginning.

Delivery(\mathcal{S})

1. for each $T \in \mathcal{S}$ do
2. for each $i \in T$, insert T into $bucket[i]$

Lemma 3. $Delivery(\mathcal{S}) = \mathcal{S} \cap Occ(\{e\})$, $e \in \mathcal{S}, O(|\mathcal{S}|)$

Let $k^*(P)$ be the smallest h satisfying $AmbiOcc(P) \subseteq Occ_{\leq h}(P)$.

Lemma 4. $P \cup \{e\}, P, k^*(P \cup \{e\}) \leq k^*(P) + 1, Occ_{\leq k^*(P)}(P) \subseteq Occ_{\leq k^*(P)}(P \cup \{e\})$

From Lemma [2](#), we have $Occ_{\leq k^*(P)}(P \cup \{e\}) \subseteq Occ_{\leq k^*(P)}(P) \subseteq Occ_{\leq k^*(P)+1}(P \cup \{e\})$. This means that $Occ_{\leq k^*(P)}(P)$ is constructed by taking $|Occ_{\leq k^*(P)}(P)|$

transactions from $Occ_{\leq k^*(P)+1}(P \cup \{e\})$ in decreasing order of the inclusion ratio for $P \cup \{e\}$. If $P \cup \{e\}$ is a child of P , we have $cov(P \cup \{e\}) \leq cov(P) \leq |Occ_{\leq k^*(P)}(P)|$. Thus, $Occ_{\leq k^*(P)}(P)$ includes a maximum co-occurrence set of $P \cup \{e\}$, and $k^*(P \cup \{e\}) \leq k^*(P) + 1$. \square

This lemma implies that for the computation of $cov(P \cup \{e\})$, we have to look only at the transactions included in $Occ_{\leq k^*(P)+1}(P)$. This reduces the computation time to $O(|Occ_{\leq k^*(P)+1}(P)|)$, where $|Occ_{\leq k^*(P)+1}(P)|$ is the sum of the sizes of transactions in $Occ_{\leq k^*(P)+1}(P)$.

We next state a lemma to determine $k^*(P \cup \{e\})$ efficiently. Let $Th(P, k) = \theta \times (|P| + 1) \times |Occ_{\leq k}(P)| - \sum_{T \in Occ_{\leq k}(P)} |T \cap P|$. If and only if $|Occ_{\leq k}(P) \cap Occ(\{e\})| \geq Th(P, k)$, the average inclusion ratio of $Occ_{\leq k}(P)$ for $P \cup \{e\}$ is no less than θ . For any transactions $T \in Occ_{=h}(P)$ and $T' \in Occ_{=h+1}(P)$, the inclusion ratio of T for $P \cup \{e\}$ is always no less than that of T' for $P \cup \{e\}$. Thus, we have the following property.

Lemma 5. $|Occ_{\leq k}(P) \cap Occ(\{e\})| \geq Th(P, k) \implies |Occ_{\leq k-1}(P) \cap Occ(\{e\})| < Th(P, k) \implies |Occ_{\leq k-1}(P) \cap Occ(\{e\})| < cov(P \cup \{e\}) \leq |Occ_{\leq k}(P)|$

Note that the statement holds for a unique k since $Th(P, k)$ is monotone decreasing for the increase of k . From the above lemma, we compute $|Occ(\{e\}) \cap Occ_{\leq k-1}(P)|$ in increasing order of k from $k = 1$, then find each item e satisfying the condition of Property 5 with k , and check whether $P \cup \{e\}$ is a child of P or not by computing $AmbiOcc(P \cup \{e\})$. The algorithm based on this method is as follows.

ALGORITHM FINDALLCHILDREN(P)

1. compute $Th(P, k)$ for each $k = 0, \dots, k^*(P) + 1$
2. **for** $k = 0$ **to** $k^*(P)$
3. compute $Occ_{=k}(P) \cap Occ(\{e\})$ for each e
4. **for each** e s.t. $|Occ(\{e\}) \cap Occ_{\leq k-1}(P)| \geq Th(P, k-1)$ **do** (for each e if $k=0$)
5. **if** $|Occ(\{e\}) \cap Occ_{\leq k}(P)| < Th(P, k)$ **then**
6. **if** $\sigma \leq |AmbiOcc(P \cup \{e\})| \leq |AmbiOcc(P)|$ **then**
7. **if** $Prt(P \cup \{e\}) = P$ **then** $P \cup \{e\}$ is a child
8. **end for**
9. **end for**

Step 6 computes $AmbiOcc(P \cup \{e\})$, then obtain $Prt(P \cup \{e\})$ by computing $AmbiOcc(P \cup \{e\}) \cap Occ\{e'\}$ for each $e' \in P \cup \{e\}$. At the time of computing these values, we already have computed $Occ_{\leq k^*(P \cup \{e\})-1}(P) \cap Occ_{\leq k^*(P \cup \{e\})}(P \cup \{e\})$, thus we have to compute only $Occ_{=k^*(P \cup \{e\})}(P) \cap Occ(\{e\})$.

Now the computation time in each iteration with respect to P is (a) $O(|Occ_{\leq k^*(P)+1}(P)|)$ for computing $cov(P \cup \{e\})$ for all e , and (b) $O(|Occ_{=k^*(P \cup \{e\})}(P)|)$ for each e such that $P \cup \{e\}$ is an ambiguous frequent itemset. In practical datasets, it is expected that $P \cup \{e\}$ is an ambiguous frequent itemset only for few e 's. Otherwise the number of ambiguous frequent itemsets is huge so that we can not enumerate them in a practically short time,

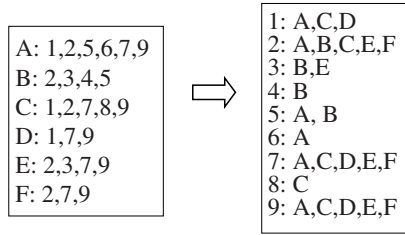


Fig. 3. Example of delivery

and we can not deal with huge output itemsets. Therefore, we can expect that (b) is not larger so much than (a), thus the computation time for an iteration is $O(|Occ_{\leq k^*(P)+1}(P)|)$, which is relatively shorter than $O(n||D||)$.

4 Weighted Ambiguous Frequent Itemset

In practical transaction databases, items of each transaction often has several different weights. For example, POS data includes the number or the price of each item purchased by a customer. In experiments in industry or natural science, each cell or item may have a kind of intensity. Such a database can be regarded as a matrix of item columns and transaction rows such that each cell has a value. One may be naturally motivated to find submatrices with a large average weight of cells. These locally heavy submatrices correspond to important objects such as clusters, and have applications in knowledge discovery and data engineering.

We define the problem as follows. We suppose that each item e of a transaction T has a weight $w(T, e)$. For an itemset P and a transaction T , we define the $w(T, P)$ of T with respect to P by $(\sum_{e \in P \cap T} w(T, e))/|P|$. For a set \mathcal{T} of weighted transactions, we define the $w(\mathcal{T}, P)$ by $(\sum_{T \in \mathcal{T}} w(T, P))/|\mathcal{T}|$. When we are given a weight threshold θ , we define the $w_{\theta}(P)$ of P by the maximum size of a transaction set having average weight no less than θ . For a given support threshold σ , an itemset is called a σ -weighted ambiguous frequent itemset if its weighted maximum co-occurrence size is no less than σ . The weighted version of the ambiguous frequent itemset enumeration problem is to output all weighted ambiguous frequent itemsets. Given these definitions, we obtain a similar neighboring relation between weighted ambiguous frequent itemsets.

Theorem 2. Let \mathcal{D} be a set of weighted transactions, θ be a weight threshold, and σ be a support threshold. Then, the number of σ -weighted ambiguous frequent itemsets is $O(|\mathcal{D}||n|)$.

The method described in the above sections is not directly applicable to improve the practical efficiency of the weighted version of our algorithm. The reason is

that Properties 1 and 2 are not valid for the weighted version. To compute the weighted maximum co-occurrence size of $P \cup \{e\}$, we need to get the transactions T in the order of $w(T, P \cup \{e\})$. If $w(T, P \cup \{e\})$ is large, then either $w(T, P)$ or $w(T, \{e\})$ has to have a large value. Thus, by getting transactions having large average weights with respect to either P or $\{e\}$, we can efficiently compute the weighted maximum co-occurrence size.

5 Hardness Result for Branch-and-Bound Approaches

We show that a hardness result for simple approaches to answer the question that why we need a sophisticated enumeration scheme. In a typical branch-and-bound algorithm, we may choose an item e and divide the enumeration problem into two subproblems; the enumeration problem of ambiguous frequent itemsets including e , and the problem for itemsets not including e . The division of the problem is done recursively until the problem includes a unique solution (ambiguous frequent itemset). In this approach we have to know the existence of solutions to the restricted problem, otherwise we will divide problems having no solution recursively, thereby exponentially many times.

The following theorem states that this problem is NP-complete. Therefore, we observe that it is hard to get a polynomial delay algorithm by typical branch-and-bound since we have to solve an NP-complete problem in each iteration.

Theorem 3. *Given a transaction database \mathcal{D} , a minimum support threshold θ , a constant number k , and a set of items S , it is NP-complete to check if there exists a frequent itemset of size at least k that is included in at least σ transactions and does not contain any item from S .*

Suppose that we are given a transaction database \mathcal{D} , a minimum support threshold σ , and a constant number k , and going to check for the existence of an itemset of size at least k that is included in at least σ transactions. This is known to be NP-complete [17]. Let I be the set of items included in transactions in \mathcal{D} , and I' be a set of items of size $|\mathcal{D}| \times |I|$ satisfying $I \cap I' = \emptyset$. We choose an item e^* from I' .

We now construct a transaction database $\mathcal{D}' = \{T \cup (I' \setminus \{e^*\}) \mid T \in \mathcal{D}\}$. Let X be a subset of I , \mathcal{T} be a transaction set of \mathcal{D} , and \mathcal{T}' be the transaction set of \mathcal{D}' corresponding to \mathcal{T} . Then, X is a frequent itemset of \mathcal{D} and $\mathcal{T} = Occ(X)$ if and only if the average inclusion ratio of \mathcal{T}' for $X \cup I'$ is strictly larger than $(|\mathcal{D}| \times |I| - 1) / (|\mathcal{D}| \times |I|)$. In particular, when $|X| = k$, the average inclusion ratio is $(|\mathcal{D}| \times |I| + k - 1) / (|\mathcal{D}| \times |I| + k)$. Here we set $\theta = (|\mathcal{D}| \times |I| + k - 1) / (|\mathcal{D}| \times |I| + k)$. Then, X is a frequent itemset of \mathcal{D} of size at least k if and only if $X \cup I'$ is an ambiguous frequent itemset of \mathcal{D}' . Therefore we have the theorem. \square

6 Computational Experiments

In general, the practical computation time of an algorithm often differs from the theoretical upper bound. The reason is that the computation time is dominated

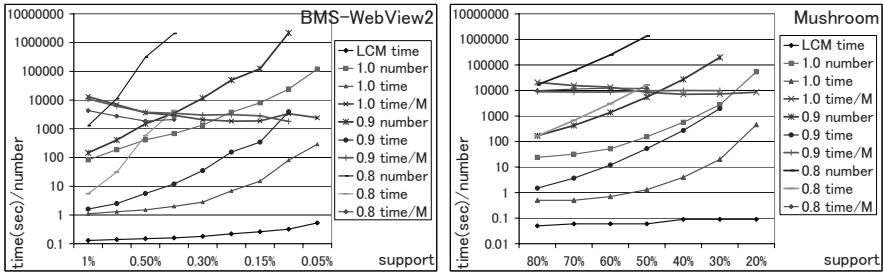


Fig. 4. Computation time and #solutions on BMS-WebView and Mushroom

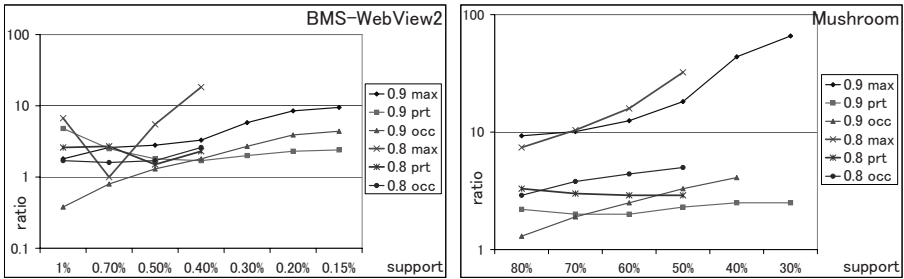


Fig. 5. Comparison of accessed items on BMS-WebView and Mushroom

by the “average”, but the theoretical upper bound looks only at the worst case. To see the gap and to be a help for the practical use, we show the results of some computational experiments.

The C code is used for the implementation. The computer used in the experiments was a notebook PC with a Pentium M 1.1GHz processor with 768MB memory. The experiments were done on cygwin which is an emulator of Linux environments on Windows. The implementation is a simpler version of our algorithm, which compute the parent in a straightforward way. The reason is to choose a simpler version is to see the performance of a simple implementation, which would help for coding. The implementation is available at the author’s homepage; <http://research.nii.ac.jp/~uno/index.html>.

We examined two practical datasets taken from FIMI repository [6]. The first is BMS-WebView2 with about 3,300 items and 77,000 transactions. The average size of transactions is 4.6, thus the dataset is quite sparse. The second is Mushroom with about 120 items and 8,000 transactions. The average size of transactions is 23, thus the dataset is not sparse.

We run the implementations with the thresholds $\theta = 0.8, 0.9$ and 1.0 . Since we could not find any implementation for the ambiguous frequent itemset enumeration, we have no comparison to other implementations. Instead of this, we compare the performance to that of an ordinary frequent itemset enumeration algorithm LCM [12, 11]. Since the frequent itemset enumeration is a special case

of our problem, it can be considered as a kind of upper bound of the performance of the ambiguous frequent itemset enumeration.

The results are shown in Fig. 4. The left is BMS-WebView2, and the right is Mushroom. The horizontal axis is for minimum support threshold, and the vertical axis is for computation time, computation time for 1 million (ambiguous) frequent itemsets, and the number of output itemsets, written in log scales.

The computation time of our algorithm increases as the decrease of minimum support, but the computation time per one million itemsets does not change drastically. It seems to change as the change of average size of $Occ(\{e\})$. Comparing to the ordinary frequent itemset mining algorithm, the performance of our algorithm is not so good. One of the reason is that the cost for computing parents of the candidate children. A simple duplication check by storing the discovered itemsets in memory will accelerate the computation when the output itemsets are few. The other reason is that in the ordinary frequent itemset mining, we can use the conditional database for the current operating itemset, which includes only items larger than the maximum item in the current operating itemset and are frequent in the database induced by the occurrence of the current operating itemset. Usually the number of items in the conditional database is much smaller than the number of items in the original database, thus the computation is faster. To reduce the difference on the computation time, further techniques for the efficient computation are still needed. The number of ambiguous frequent itemsets increases drastically by the decrease of density threshold. In practice, we should use a threshold slightly smaller than 1.0.

We also looked at several statistics on the experiments in Figure 5. “max” means the ratio of ambiguous frequent itemsets and the number of maximal ambiguous frequent itemsets to which no item addition yields an ambiguous frequent itemset. “prt” shows the ratio of the number of accessed items between a straightforward algorithm and the sophisticated algorithm proposed in this paper, and “occ” indicates that between delivery for computing the frequencies of all additions of items, and delivery for computing the parent. As we can see, these ratio increase as the increase the number of solutions. Thus, we can expect the decrease of the number of solutions by outputting only maximal ones. The speedup is also expected by introducing our sophisticated parent computation, but the effect will be limited. The big ratio of “occ” implies that the big gap between computation time of our algorithm and ordinary frequent itemset mining. It also implies that more practical improvements are needed.

7 Conclusion and Future Work

We formulated the enumeration problem of ambiguous frequent itemsets, and proposed a polynomial delay polynomial space algorithm. The algorithm is naturally extended to a weighted version. The experimental performance for practical datasets is acceptable, but improvements on practical performance is a crucial future work. Another interesting research topic is extending the technique to other frequent pattern mining problems.

Acknowledgments

This research was supported by Grant-in-Aid for Scientific Research of Japan, “Developing efficient and accurate algorithms for large-scale data processing in genome science”, and a joint-research fund of National Institute of Informatics.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast Discovery of Association Rules. In: *Advances in Knowledge Discovery and Data Mining*, pp. 307–328 (1996)
2. Asai, T., Abe, K., Kawasoe, S., Arimura, H., Sakamoto, H., Arikawa, S.: Efficient Substructure Discovery from Large Semi-structured Data. In: *SDM 2002* (2002)
3. Avis, D., Fukuda, K.: Reverse Search for Enumeration. *Disc. App. Math.* 65, 21–46 (1996)
4. Bayardo Jr., R.J.: Efficiently Mining Long Patterns from Databases. In: *SIGMOD 1998*, pp. 85–93 (1998)
5. Besson, J., Robardet, C., Boulicaut, J.F.: Mining Formal Concepts with a Bounded Number of Exceptions from Transactional Data. In: Goethals, B., Siebes, A. (eds.) *KDID 2004*. LNCS, vol. 3377, pp. 33–45. Springer, Heidelberg (2005)
6. Goethals, B.: The FIMI repository (2003), <http://fimi.cs.helsinki.fi/>
7. Liu, J., Paulsen, S., Wang, W., Nobel, A., Prins, J.: Mining Approximate Frequent Itemsets from Noisy Data. In: *ICDM 2005*, pp. 721–724 (2005)
8. Seppanen, J.K., Mannila, H.: Dense Itemsets. In: *SIGKDD 2004* (2004)
9. Shen-Shung, W., Suh-Yin, L.: Mining Fault-Tolerant Frequent Patterns in Large Databases. In: *ICS 2002* (2002)
10. Takeda, M., Inenaga, S., Bannai, H., Shinohara, A., Arikawa, S.: Discovering Most Classificatory Patterns for Very Expressive Pattern Classes. In: Grieser, G., Tanaka, Y., Yamamoto, A. (eds.) *DS 2003*. LNCS (LNAI), vol. 2843, pp. 486–493. Springer, Heidelberg (2003)
11. Uno, T., Asai, T., Uchida, Y., Arimura, H.: An Efficient Algorithm for Enumerating Closed Patterns in Transaction Databases. In: Suzuki, E., Arikawa, S. (eds.) *DS 2004*. LNCS (LNAI), vol. 3245, pp. 16–31. Springer, Heidelberg (2004)
12. Uno, T., Kiyomi, M., Arimura, H.: LCM ver. 2: Efficient Mining Algorithms for Frequent/Closed/Maximal Itemsets. In: *IEEE ICDM 2004 Workshop FIMI 2004* (2004)
13. Uno, T., Arimura, H.: An Efficient Polynomial Delay Algorithm for Pseudo Frequent Itemset Mining. In: Corruble, V., Takeda, M., Suzuki, E. (eds.) *DS 2007*. LNCS (LNAI), vol. 4755, pp. 219–230. Springer, Heidelberg (2007)
14. Uno, T.: An Efficient Algorithm for Enumerating Pseudo Cliques. In: Tokuyama, T. (ed.) *ISAAC 2007*. LNCS, vol. 4835, pp. 402–414. Springer, Heidelberg (2007)
15. Wang, J.T.L., Chirn, G.W., Marr, T.G., Shapiro, B., Shasha, D., Zhang, K.: Combinatorial pattern discovery for scientific data: some preliminary results. In: *SIGMOD 1994*, pp. 115–125 (1994)
16. Yang, C., Fayyad, U., Bradley, P.S.: Efficient Discovery of Error-Tolerant Frequent Itemsets in High Dimensions. In: *SIGKDD 2001* (2001)
17. Zaki, M.J., Ogihara, M.: Theoretical foundations of association rules. In: *3rd ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery* (1998)

Characteristic-Based Descriptors for Motion Sequence Recognition^{*}

Liang Wang, Xiaozhe Wang, Christopher Leckie, and Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering
The University of Melbourne, Parkville, Victoria 3010, Australia
{lwwang, catwang, caleckie, rao}@csse.unimelb.edu.au

Abstract. This paper proposes an approach based on characteristic descriptors for recognition of articulated and deformable human motions from image sequences. After extracting human movement silhouettes from motion videos, we apply Tensor Subspace Analysis to embed normalized dynamic silhouette sequences into low-dimensional forms of multivariate time series. Structure-based statistical features are then extracted from such multivariate time series to summarize motion patterns (as descriptors) in a compact manner. A multi-class Support Vector Machine classifier is used to learn and predict the motion sequence categories. The proposed method is evaluated on two real-world state-of-the-art video data sets, and the results have shown the power of our method for recognizing human motion sequences with intra- and inter-person variations on both temporal and spatial scales.

Keywords: motion sequence recognition, multivariate time series, tensor subspace analysis, characteristic-based descriptor, SVM.

1 Introduction

There has recently been growing interest in algorithms that can extract useful information from non-traditional data such as images and videos [1].

[2] aims to discover and understand patterns of human movements from video sequences, . . ., determining typical and anomalous motion patterns, classifying motions into known categories (. . ., walking or riding), and discovering unknown motion patterns by clustering. Human motion analysis has a wide range of applications such as video surveillance (. . ., finding suspicious events such as a person wandering around in a parking lot), human-machine interface (. . ., gesture-driven control) and video understanding and summarization (. . ., interpretation of sport events).

This paper focuses on the analysis of short video clips consisting of individual One of the key challenges in the interpretation of human motions is how to transform semantically agnostic video signals to meaningful

^{*} This work was supported by the Australian Research Council (ARC) Discovery Projects DP0663196 and DP0663979.

feature representations (i.e., low-level feature extraction steps) that provide a sufficient encoding of different motion structures. The resulting outputs can be used as inputs to higher-level recognition processes. There are several issues in this context that raise challenges for human motion analysis [3]: 1) Repeated performance of the same motion by the same person in different instances can vary. 2) The same motion performed by different people can vary because different people have different physical structures or perform motions in different ways. 3) The same motion may have different temporal durations because of the difference in motion speeds. 4) Different motions may have significantly different temporal durations. We consider these variations in motions (due to different instances, different persons with different body types and motion styles, and different motion speeds) as

. The objective of this paper is to develop an approach to represent and recognize articulated and deformable human motions while accounting for the above spatio-temporal variations in motion execution.

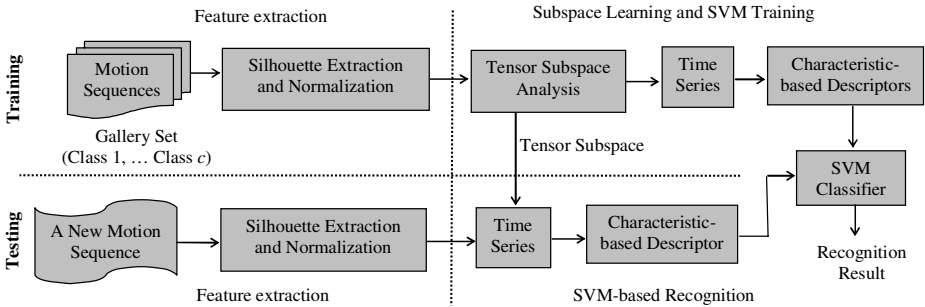


Fig. 1. Framework of characteristic-based descriptors for motion sequence recognition

To this end, this paper proposes a method based on characteristic descriptors for recognizing human motion sequences, as shown in Figure 1. The proposed method consists of the following steps: 1) Extract silhouettes of the moving human from the input sequence. We use normalized raw silhouettes as visual cues, because they are simple but informative, and easy to obtain from original video data. 2) Transform high-dimensional silhouette inputs into a low-dimensional feature space. In particular, we use computationally efficient Tensor Subspace Analysis (TSA) [4] for dimensionality reduction while preserving information of silhouette images. To the best of our knowledge, no previous work has investigated its use in this context. 3) Map each motion sequence into a form of Time Series in the learned embedding space, from which we extract structure-based statistical features to construct a vector-based pattern representation (i.e., Time Series), which naturally converts our motion sequence classification into a Time Series classification problem. 4) Learn a multi-class Support Vector Machine (SVM) classifier [5] using labeled data, and then use it to classify unknown motion sequences into one of a set of known

motion categories. Experimental results on two real-world video data sets have validated the proposed method.

The remainder of this paper is organized as follows. Section 2 briefly reviews related work. Section 3 details each step of the proposed method. The experimental results are presented in Section 4, prior to a summary in Section 5.

2 Related Work

Motion representation and recognition are central to the interpretation of human motions. Various visual cues have been examined in current studies on human motion analysis, . . ., optical flow [6], local descriptors [7], motion trajectories from feature tracking [8,9], etc. For example, Schuldt . . . [7] constructed video representations in terms of local space-time features. Efros . . . [6] proposed a spatiotemporal descriptor based on blurred optical flow measurements to recognize actions.

Image measurements in terms of optical flow or interest points can be unreliable in cases of smooth surfaces, motion singularities and low-quality videos. Feature tracking is difficult due to the great variability in the appearance and articulation of the human body. Fortunately, human motions can be regarded as over time. The use of features derived from silhouettes has been explored recently. For example, Blank . . . [9] utilized properties of the solution to the Poisson equation to extract features from space-time silhouettes for action recognition and detection. Silhouette extraction from video is relatively easier for current imperfect vision techniques. So the method that we present here uses (probably imperfect) space-time silhouettes as basic cues to derive effective motion feature representations.

There are two major categories of approaches to motion recognition [2]. The approaches based on first convert time-varying features corresponding to a motion sequence into a static pattern, and then compare it to pre-stored motion prototypes during recognition. In contrast, - approaches usually use temporal models such as Hidden Markov Models (HMMs), Conditional Random Fields (CRFs) or their variants [10,8,11] to model and classify motions. For example, Nguyen . . . [8] learned and detected activities from movement trajectories using hierarchical HMMs.

Temporal probabilistic models such as HMMs and CRFs usually require very detailed mathematical and statistical modeling, which involves assumptions about the probability distributions of variables of the dynamical model and development of inference methods and parameter learning algorithms, which have a high computational cost. In contrast, our proposed method converts a sequence of silhouette images associated with a motion video into a form of multivariate time series, from which we extract structural statistical features to summarize motion pattern. This strategy using characteristic-based descriptors reduces our temporal classification problem into a static classification one. Accordingly, any of the existing efficient methods for classification can be applied for learning and predicting motion sequence classes.

3 Methodology

3.1 From Motion Image Sequences to Silhouette Sequences

Informative features are critical to motion characterization. The features should be simple, intuitive and easy to extract automatically. As stated before, our work prefers to use silhouettes as basic cues. How should we segment the moving human region from the background image? This can usually be accomplished by well-established motion detection techniques. Various categories of methods for motion detection have been widely studied in the computer vision community [2] (e.g., background subtraction and temporal differencing). Motion segmentation is not our focus in this paper. As such, the video data sets to be used in our experiments have already contained the segmented silhouette masks.

Given a motion video \mathcal{V} including T image frames $\mathbf{I}_1, \dots, \mathbf{I}_T$, we can obtain an associated sequence of moving silhouettes $\mathcal{S} = [\mathbf{S}_1, \mathbf{S}_2, \dots, \mathbf{S}_T]$. The size and position of the foreground human region in silhouette images vary with the distance of the human from the camera, the size of the human and the motion being performed. The silhouette images are thus of varying sizes and positions on the basis of keeping the aspect ratio property of the silhouette so that the resulting silhouette images $\hat{\mathcal{S}} = [\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_T]$ are of equal dimensions and contain as much foreground information as possible without distorting the motion shape.

3.2 From Silhouette Sequences to Multivariate Time Series

Human silhouettes through the duration of a motion may be generally expected to lie on a low-dimensional manifold embedded in a high-dimensional image space. It is well known that high dimensionality not only slows the algorithmic processing, but also degrades performance. Therefore we are motivated to represent motions in a more compact subspace rather than the ambient space.

Traditional dimensionality reduction algorithms such as Principal Component Analysis (PCA) and Linear Discriminative Analysis (LDA) [12] for image processing usually represent an $n_1 \times n_2$ image by a vector in high-dimensional space \mathcal{R}^n ($n = n_1 \times n_2$), and find a map from \mathcal{R}^n to \mathcal{R}^l ($l < n$). However, an image is intrinsically a matrix (or a tensor). Tensor-based methods have been recently studied in the image processing community [4, 13, 14, 15]. To represent the relationship between the row and column vectors of the image matrix, e.g., to preserve the spatial information of silhouette images, we select TSA to perform subspace learning of the articulated motion space [16], in which an image is represented as a second-order tensor in $\mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ (where \mathcal{R}^{n_1} and \mathcal{R}^{n_2} are two vector spaces and \otimes denotes the tensor product). TSA has been recently proposed and demonstrated for use in static face recognition [4]. Here we extend its application to dynamic silhouette data with highly-varied motion shapes.

Given a set of m points $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_m\}$ in $\mathcal{R}^{n_1} \otimes \mathcal{R}^{n_2}$ (e.g., normalized silhouettes here), TSA aims to find two transformation matrices \mathbf{U} of size $n_1 \times l_1$ and \mathbf{V} of size $n_2 \times l_2$ that map these m points to another set of points $\{\mathbf{Y}_1, \mathbf{Y}_2, \dots, \mathbf{Y}_m\}$ in $\mathcal{R}^{l_1} \otimes \mathcal{R}^{l_2}$ ($l_1 < n_1, l_2 < n_2$), such that $\mathbf{Y}_i = \mathbf{U}^T \mathbf{X}_i \mathbf{V}$.

These m points can build a weighted graph \mathcal{G} to model the local geometrical structure of data manifold \mathcal{M} . Let \mathbf{W} be the weight matrix of \mathcal{G} , and

$$W_{ij} = \begin{cases} e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2/\tau}, & \text{if } \mathbf{X}_i \text{ and } \mathbf{X}_j \text{ are "close"} \\ 0 & \text{, otherwise} \end{cases} \quad (1)$$

where “close” can be defined by the k nearest neighbors, . . ., \mathbf{X}_i is among the k nearest neighbors of \mathbf{X}_j , or \mathbf{X}_j is among the k nearest neighbors of \mathbf{X}_i , and τ is a suitable constant. The function $e^{-\|\mathbf{X}_i - \mathbf{X}_j\|^2/\tau}$ is the so called heat kernel, and $\|\cdot\|$ is the Frobenius norm of a matrix. A reasonable transformation representing the graph structure can be obtained by solving the following optimization problem based on the graph Laplacian [4]:

$$\min_{\mathbf{U}, \mathbf{V}} \sum_{i,j} \|\mathbf{U}^T \mathbf{X}_i \mathbf{V} - \mathbf{U}^T \mathbf{X}_j \mathbf{V}\|^2 W_{ij} \quad (2)$$

It is equivalent to the following simultaneous optimization problem [4]:

$$\min_{\mathbf{U}, \mathbf{V}} \frac{\text{tr}(\mathbf{U}^T (\mathbf{D}_V - \mathbf{W}_V) \mathbf{U})}{\text{tr}(\mathbf{U}^T \mathbf{D}_V \mathbf{U})} \quad \text{and} \quad \min_{\mathbf{U}, \mathbf{V}} \frac{\text{tr}(\mathbf{V}^T (\mathbf{D}_U - \mathbf{W}_U) \mathbf{V})}{\text{tr}(\mathbf{V}^T \mathbf{D}_U \mathbf{V})} \quad (3)$$

where \mathbf{D} is a diagonal matrix with $D_{ii} = \sum_j W_{ij}$, $\mathbf{D}_V = \sum_i D_{ii} \mathbf{X}_i \mathbf{V} \mathbf{V}^T \mathbf{X}_i^T$, $\mathbf{W}_V = \sum_{i,j} \mathbf{X}_i \mathbf{V} \mathbf{V}^T \mathbf{X}_j^T$, $\mathbf{D}_U = \sum_i D_{ii} \mathbf{X}_i^T \mathbf{U} \mathbf{U}^T \mathbf{X}_i$, and $\mathbf{W}_U = \sum_{i,j} \mathbf{X}_i^T \mathbf{U} \mathbf{U}^T \mathbf{X}_j$. An iterative method is suggested in [4] to address this optimization problem. If \mathbf{U} is first fixed, then \mathbf{V} can be computed by solving $(\mathbf{D}_U - \mathbf{W}_U) \mathbf{v} = \lambda \mathbf{D}_U \mathbf{v}$. Once \mathbf{V} is obtained, \mathbf{U} can be updated by solving $(\mathbf{D}_V - \mathbf{W}_V) \mathbf{u} = \lambda \mathbf{D}_V \mathbf{u}$. Thus, the optimal \mathbf{U} and \mathbf{V} can be obtained by iteratively computing the above generalized eigenvector problems.

After learning the tensor subspace including the first $l_1 \times l_2$ principal components, any silhouette sequence \mathcal{V} can be accordingly projected into a trajectory \mathcal{P} in such a parametric space $\mathcal{P} = \{\mathbf{P}_1, \mathbf{P}_2, \dots, \mathbf{P}_T\}$, $\mathbf{P}_i \in \mathcal{R}^{l_1} \otimes \mathcal{R}^{l_2}$, while the temporal order across frames is preserved explicitly. Then, we may easily convert \mathcal{P} into a form of multivariate time series with the number of dimensions $l = l_1 \times l_2$.

3.3 From Multivariate Time Series to Characteristic-Based Descriptor

Now motion sequence recognition can be regarded as a time series classification problem. Several alternative paradigms for time series classification have been proposed [17]. Here we wish to extract the most informative features to summarize multivariate time series so as to turn time series classification into static vector-based classification.

In this study, we investigated various data characteristics from diverse perspectives related to univariate time series. A univariate time series can be represented as an ordered set of n real-valued variables Z_1, \dots, Z_n . We selected

the nine most informative, representative and easily measurable characteristics to summarize the time series structure:

..., and . It can be seen that this set of characteristic metrics to represent univariate time series and their structure-based features not only includes conventional features (. . ., trend), but also covers many advanced features (. . ., chaos) which are derived from research on new phenomena [18]. Based on these identified characteristics, corresponding metrics are calculated for constructing the structure-based feature vectors [19], that form a rich portrait of the nature of a time series.

In time series analysis, decomposition is a critical step for transforming the series into a format for statistical measurement [20]. Therefore, to obtain a precise and comprehensive calibration, some measures need to be calculated on both the raw time series data, Z_t , (referred to as data), as well as the ‘trend and seasonally adjusted’ time series, Z'_t , (referred to as data). Four of the nine selected features, . . ., Serial-correlation, Non-linearity, Skewness, and Kurtosis, are calibrated on both and data, each of which contributes two metrics to our family. The remaining five selected features are calibrated only on data, leading to a total of thirteen metrics.

For each dimension of a l -dimensional multivariate time series, we may obtain 13 statistical features to construct the feature vector. Thus, the multivariate time series can be summarized by a r -dimensional ($r = 13 \times l$) vector \mathbf{f} . We refer to such a feature vector \mathbf{f} as a - .

3.4 From Characteristic-Based Descriptor to Motion Recognition

Motion recognition aims to classify an unknown test sequence into one of c known motion classes. Among many available methods for static classification problems, we adopt a multi-class SVM classifier because its performance surpasses other competing classification methods on many benchmark data sets [21].

We are given a labeled training set $\mathcal{T} = \{(\mathbf{f}_1, y_1), (\mathbf{f}_2, y_2), \dots, (\mathbf{f}_n, y_n)\}$, where $y_i \in \mathcal{Y} = \{1, 2, \dots, c\}$ are the known class labels. The multi-class SVM involves a set of discriminant functions $g_y : \mathcal{F} \subseteq \mathcal{R}^r \rightarrow \mathcal{R}, y \in \mathcal{Y}$ defined as

$$g_y(\mathbf{f}) = \langle \boldsymbol{\alpha}_y \cdot \mathbf{k}_S(\mathbf{f}) \rangle + b_y \tag{4}$$

where $\mathbf{k}_S(\mathbf{f}) = [k(\mathbf{f}, \mathbf{s}_1), \dots, k(\mathbf{f}, \mathbf{s}_v)]^T$ is the vector of evaluations of kernel functions centered at support vectors $S = \{\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_v\}, \mathbf{s}_i \in R^r$ which are usually a subset of the training data, $\mathbf{A} = [\boldsymbol{\alpha}_1, \dots, \boldsymbol{\alpha}_c]$ is composed of all weight vectors, and $\mathbf{b} = [b_1, \dots, b_c]^T$ is a vector of all biases. The multi-class classification rule $q : \mathcal{F} \rightarrow \mathcal{Y} = \{1, 2, \dots, c\}$ is defined as

$$q(\mathbf{f}) = \arg \max_{y \in \mathcal{Y}} g_y(\mathbf{f}) \tag{5}$$

Several methods to train multi-class SVM are compared in [5]. Simple - is adopted here, which transforms a multi-class problem into a series of c binary subtasks that can be trained by binary SVMs. Also, we use the Radial Basis Function (RBF) $k(\mathbf{f}_a, \mathbf{f}_b) = \exp(-0.5 \|\mathbf{f}_a - \mathbf{f}_b\|^2 / \sigma^2)$ as the kernel in our experiments.

4 Experiments

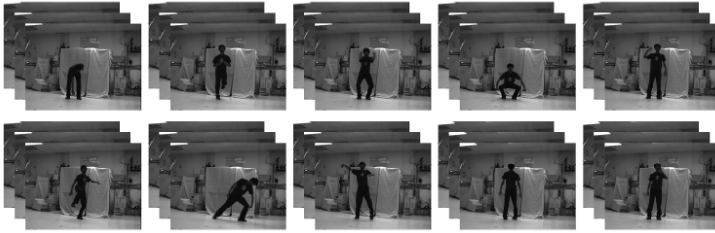
4.1 Evaluation Databases

There is no standard evaluation database in the domain of human motion analysis. We use two state-of-the-art databases in [22] and [9] to evaluate our method. These two databases are appreciably sized (among current databases publicly available), in terms of the number of persons, motions and video sequences.

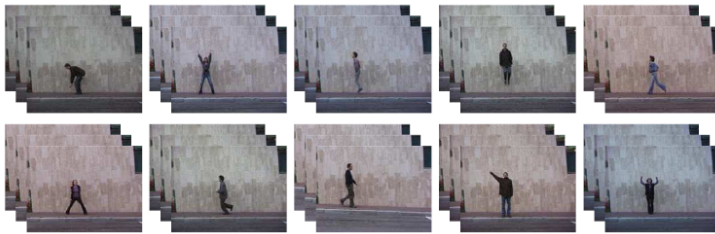
D-I: Data set I consists of 10 different motions performed by one person, each comprising 10 instances, and 100 sequences in total [22]. These motions are

(Pick), (Jog), (Bend-Side), (Turn), and (Phone).

Examples are shown in Figure 2(a). Different instances of the same motion may consist of varying relative speeds. This data set is used to examine the effect of (alone) on motion recognition, as well as slightly different intra-person motion styles among different instances.



(a) D-I



(b) D-II

Fig. 2. Example images of motion data sets

D-II: Data set II consists of 90 low-resolution videos from 9 different people, each performing 10 different motions [9]. These motions are (Jack), (Jump), (Pjump), (Side), (Wave1), and (Wave2). Example images are shown in Figure 2(b). Except for , whether the other motions are in essence periodic or not, people are asked to

¹ <http://www.wisdom.weizmann.ac.il/~vision/SpaceTimeActions.html>

perform those motions multiple times in a continuously repetitive manner. From these 90 videos, we extract 198 motion sequences for our experiments, each of which includes a complete cycle of atomic motion. The number of sequences of each motion is respectively 9, 23, 24, 27, 14, 22, 25, 16, 19, and 19 for bend, jack, jump, pjump, run, side, skip, walk, wave1, and wave2. In addition to temporal execution rates, there are \neq between the same motions since different people have different physical sizes and perform motions in different styles and speeds. Thus this data set is more realistic for testing the method’s robustness to motion variations at both temporal and spatial scales.

4.2 Data Processing and Classification

We adapted the “leave-one-out” cross-validation method for the experiments on both data sets. For D-I, we partition the data set into 10 disjoint sets, each containing one instance of every class of motion. Each time we leave one set out for the test, and use the remaining nine sets for training. This process is repeated 10 times for D-I. For D-II, since different motions are performed by different people in a varied number of repeats, we decided to use the person id to subset the data. That is, we divide the data set into 9 sets, each set including all motions from one person. Each time we leave one set out for the test, and use the remaining sets for training. Thus, if one video in the left-out set is classified correctly, it must show a high similarity to a video from another different person performing the same motion. This process is repeated 9 times for D-II.

For silhouette extraction, we directly use the silhouette masks obtained from [22,9], even though the quality of these silhouette images is not very satisfactory, consisting of leaks and intrusions due to imperfect segmentation. Then, we center and normalize all silhouette images into the same dimension (. . . , 48×32 pixels). Figure 3 illustrates the process “from a motion image sequence to an associated sequence of normalized silhouette images”.

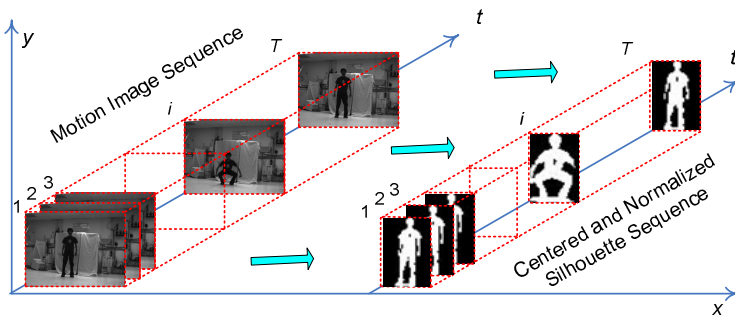


Fig. 3. Illustration to the process of silhouette extraction

When learning the tensor subspace using a given training set, we use the k -nearest neighbors ($k = 20$) to construct the affinity graph. A heat function with $\tau = 1000$ is adopted for the weight matrix. When computing U and V ,

the number of iterations is taken to be 15, and U is initially set to the identity matrix. TSA significantly reduces the dimension number of the input features from 48×32 to 4×4 (. . , $l_1 = l_2 = 4$) (thus leading to lower computational cost), while achieving high accuracy. Note that these parameter settings were found empirically in a series of experiments. Each motion sequence is projected into a 16-dimensional time series in the learned embedding space, from which we extract 13 statistical features corresponding to each univariate time series. Then these features from each univariate time series are joined as one 208-dimensional (16×13) vector. Figure 4 gives several examples of motion sequences in the form of multivariate time series after TSA transformation (in which only the first 7 dimensions, each color per dimension, are shown for simplicity and clarity).

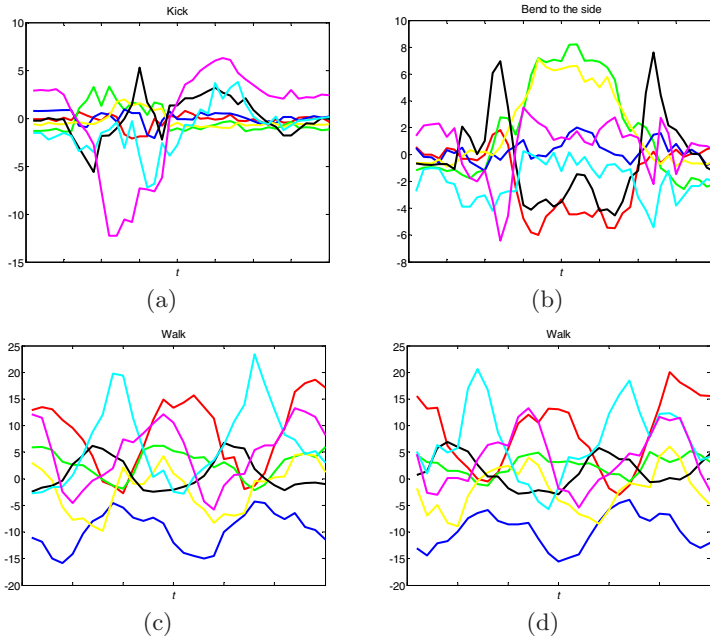


Fig. 4. Examples of multivariate time series forms of motions. Two different motions performed by the same person in D-I (top), and a single motion class performed by two different people in D-II (bottom).

There are two free parameters which need to be tuned for the SVM, namely the regularization constant C and the argument σ of the kernel function. A common method to tune the parameters is to use grid search to select the best parameter from a pre-selected set $\Theta = \{(C_1, \sigma_1), (C, \sigma_2), \dots, (C_a, \sigma_a)\}$. After the best parameter (C^*, σ^*) is tuned, a multi-class SVM classifier is trained using all training data available. Then it may be used to predict the class labels of new test sequences. We use the parameter sets of $(\sigma = 3, C = 10)$ for D-I and $(\sigma = 2.3, C = 10)$ for D-II in our experiments. In addition, we implement the Linear (NN) classifier as a baseline for comparison.

4.3 Results and Analysis

The results of motion sequence recognition are summarized in Tables 1 and 2. Note that the recognition rates reported here are measured in terms of the percentage of correctly classified motion sequences among all test sequences. The results show that: 1) Dynamic silhouettes are indeed informative to encode motion information, and our feature extraction and representation methods are effective; 2) D-I is more easily classified. This is probably because all motion instances are from the same person, thus there are comparatively fewer changes among time-varying silhouette shapes when the same motion is performed; 3) In contrast, D-II is harder to classify because those motions are performed by different people with different body builds and motion styles; and 4) SVM performs better than NN. In summary, our method is demonstrated to be effective for recognition of human motion sequences with temporal and spatial variations due to different people.

Table 1. Recognition Rates for D-I Consisting of 100 Motion Sequences(%)

Motions	Pick	Jog	Push	Squash	Wave	Kick	Bend-Side	Throw	Turn	Phone	Average
NN	80	100	90	90	100	30	100	70	100	80	84
SVM	90	100	100	100	100	70	100	100	100	100	96

Table 2. Recognition Rates for D-II Consisting of 198 Motion Sequences (%)

Motions	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2	Average
NN	88.9	87	79.2	59.3	64.3	95.5	60	100	89.5	68.4	78.3
SVM	100	95.7	87.5	92.6	85.7	86.4	84	100	100	89.4	91.4

To examine and analyze which motion sequences are incorrectly classified (and why), we show confusion matrices with respect to the two data sets in Figure 5. The elements of each row in the confusion matrix represent the probability that a certain kind of motion is classified as other kinds of motions. From Figure 5, it can be seen that most motion sequences have perfect classification, and only a small number of motions (. . ., Kick/Pick in D-I, Skip/Jump/Run, and Wave2/Jack in D-II) are easily confused. In addition to poor silhouette segmentation, high similarities among silhouette shapes in these motions (with locally similar moving patterns) may contribute to these confusions.

4.4 Discussion and Future Work

Although the experiments have demonstrated that our methodology works well, further evaluation on a larger database, with multi-varied motions, persons and scenarios, needs to be examined. Apart from simple silhouette observations, other visual cues could be available from raw videos. Fusion of multiple cues may be

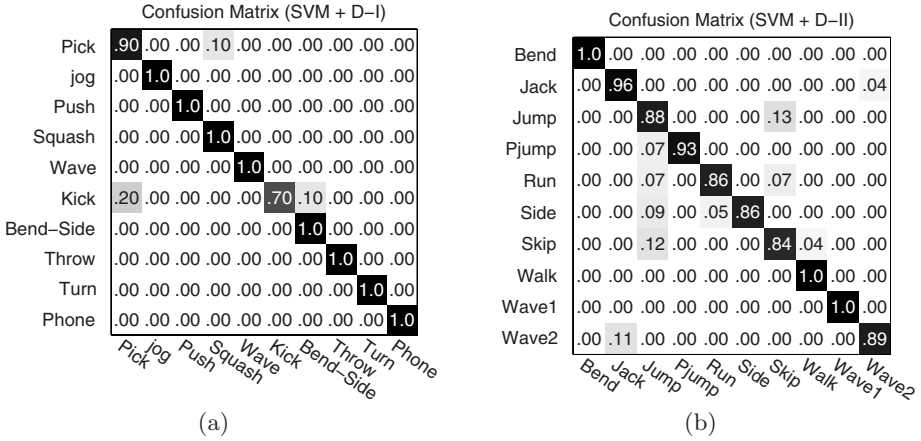


Fig. 5. Confusion matrices of motion sequence classification

preferable for improving accuracy and reliability. When extracting characteristic-based descriptors, we separately process each dimension of the multivariate time series and then simply stack these individual features together. More sophisticated methods that can exploit mutual information among different dimensions of multivariate time series can be useful.

The proposed method currently focuses on the analysis of short video clips consisting of a single individual motion. As long-term goals, we wish to extend our work in several ways: 1) Segmentation and localization in long videos, to find whether a specified action exists in the observed video, and where it is in the video; 2) Behavior profiling, to summarize which basic actions exist in the video, from which the behavior event can be summarized; and 3) Discovering normal and abnormal motion patterns, which may be performed by automatically clustering motion events that frequently occur over a period of time as normal actions, whereas rare actions in comparison can be inferred as being abnormal.

5 Conclusion

This paper has described an effective method for motion sequence recognition. It starts with extracting time-varying silhouettes from image sequences, and then embeds dynamic silhouette sequences into low-dimensional multivariate time series by tensor subspace analysis. Characteristic-based statistical features are obtained from multivariate time series to characterize motion patterns. A multi-class SVM classifier is finally adopted to learn and predict the categories of motion patterns. Our experimental results on two state-of-the-art data sets have validated the proposed method. As a by-product, the multivariate time series for the two video data sets derived from our method provide two dynamic and high-dimensional time series data sets for researchers working on time series analysis in the data mining community.

References

1. Rosenfeld, A., Doermann, D., DeMenthon, D.: Video Mining. Kluwer, Dordrecht (2003)
2. Wang, L., Hu, W.M., Tan, T.N.: Recent developments in human motion analysis. *Pattern Recognition* 36(3), 585–601 (2003)
3. Yacoob, Y., Black, M.J.: Parameterized modeling and recognition of activities. *Computer Vision and Image Understanding* 73(2), 232–247 (1999)
4. He, X., Cai, D., Niyogi, P.: Tensor subspace analysis. In: *Proc. Advances in Neural Information Processing Systems* 18 (2005)
5. Hsu, C.W., Lin, C.J.: A comparison of methods for multiclass support vector machines. *IEEE Trans. Neural Networks* 13(2) (2002)
6. Efros, A., Berg, A., Mori, G., Malik, J.: Recognizing action at a distance. In: *Proc. Int. Conf. Computer Vision* (2003)
7. Schuldts, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: *Proc. Int. Conf. Pattern Recognition.*, vol. 3, pp. 32–36 (2004)
8. Nguyen, N., Phung, D., Venkatesh, S., Bui, H.: Learning and detecting activities from movement trajectories using the hierarchical hidden Markov models. In: *Proc. Int. Conf. Computer Vision and Pattern Recognition* (2005)
9. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: *Proc. Int. Conf. Computer Vision*, pp. 1395–1402 (2005)
10. Brand, M., Oliver, N., Pentland, A.: Coupled hidden Markov models for complex action recognition. In: *Proc. Int. Conf. Computer Vision and Pattern Recognition* (1996)
11. Sminchisescu, C., Kanaujia, A., Li, Z., Metaxas, D.: Conditional models for contextual human motion recognition. In: *Proc. Int. Conf. Computer Vision*, vol. 2, pp. 1808–1815 (2005)
12. Belhumeur, P., Hefanpha, J., Kriegman, D.: Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence* 19(7), 711–720 (1997)
13. Vasilescu, M., Terzopoulos, D.: Multilinear subspace analysis of image ensembles. In: *Proc. Int. Conf. Computer Vision and Pattern Recognition* (2003)
14. Tao, D., Li, X., Hu, W., Maybank, S.J., Wu, X.: Supervised tensor learning. In: *Proc. Int. Conf. Data Mining*, pp. 450–457 (2005)
15. Tao, D., Li, X., Wu, X., Maybank, S.J.: General tensor discriminant analysis and gabor features for gait recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(10), 1700–1715 (2007)
16. Wang, L., Leckie, C., Wang, X., Kotagiri, R., Bezdek, J.: Tensor space learning for analyzing activity patterns from video sequences. In: *Proc. ICDM Workshop on Knowledge Discovery and Data Mining from Multimedia Data and Multimedia Applications* (2007)
17. Kadous, M.W., Sammut, C.: Classification of multivariate time series and structured data using constructive induction. *Machine Learning* 58, 179–216 (2005)
18. Joseph, E.: Chaos Driven Futures. *Future Trends Newsletter* 24(1) (1993)
19. Wang, X., Smith, K., Hyndman, R.: Characteristic-based clustering for time series data. *Data Mining and Knowledge Discovery* 13(3), 335–364 (2006)
20. Hamilton, J.: *Time Series Analysis*. Princeton University Press, Princeton (1994)
21. Cristianini, N., Shawe-Taylor, J.: *Support Vector Machines*. Cambridge University Press, Cambridge (2000)
22. Veeraraghavan, A., Chellappa, R., Roy-Chowdhury, A.: The function space of an activity. In: *Proc. Int. Conf. Computer Vision and Pattern Recognition* (2006)

Protecting Privacy in Incremental Maintenance for Distributed Association Rule Mining

W.K. Wong¹, David W. Cheung¹, Edward Hung², and Huan Liu³

¹ Department of Computer Science, The University of Hong Kong, Pokfulam Road, Hong Kong

{`wkwong2`, `dcheung`}@cs.hku.hk

² Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon

`csehung@comp.polyu.edu.hk`

³ Department of Computer Science and Engineering, Arizona State University Tempe, Arizona, USA

`huan.liu@asu.edu`

Abstract. Distributed association rule mining algorithms are used to discover important knowledge from databases. Privacy concerns can prevent parties from sharing the data. New algorithms are required to solve traditional mining problems without disclosing (original or derived) information of their own data to other parties. Research results have been developed on (i) incrementally maintaining the discovered association rules, and (ii) computing the distributed association rules while preserving privacy. However, no study has been conducted on the problem of the maintenance of the discovered rules with privacy protection when new sites join the old sites. We propose an algorithm SIMDAR for this problem. Some techniques we developed can even further reduce the cost in a normal association rule mining algorithm with privacy protection. Experimental results showed that SIMDAR can significantly reduce the workload at the old sites by up to 80%.

1 Introduction

Protecting privacy is an important element in many database applications. Many countries set up privacy laws to clearly protect privacy, e.g. Australia, United States, United Kingdom. For example, medical records and personal information of patients in a hospital should not be disclosed. Direct public access to private information stored in databases is not allowed. Some traditional algorithms may hence be rendered infeasible in practice.

In a distributed association rule mining process, branches of the same companies or even different companies cooperate together to find out the global association rules. Apart from the privacy concerns about individual records, each party may not be willing to share its own data or even let other parties know any derived information. Most traditional algorithms cannot work without disclosing sensitive information like the counts of itemsets in a particular party.

There are new approaches to handle the privacy-preserving data mining problems. One approach is to modify the database randomly so that other parties can fully access the modified data. However, data mining algorithms can then produce only approximate results using the modified data. Another approach is to develop new algorithm applying cryptographic techniques so that accurate results can be obtained without direct access to the source data. This kind of approach is more expensive but security can be usually proved (with limited information disclosure). [10] gives a solution to the problem of association rule mining with privacy protection.

When new parties join, the old association rules may become out-dated and need updates. The naive method of recomputing the association rules from scratch is expensive. In fact, we can reduce the cost by using the old results to incrementally update the rules. The maintenance problem of association rules in a centralized database is studied in [5,4]. They greatly reduce the number of candidate sets required to scan the database and hence reduce the total process time. Our idea is to apply a similar property used in [5,4], and aim to reduce the candidate set size and so the running time.

There is a tradeoff between privacy and efficiency in privacy-preserving problems. The solutions may be even impractical when complete privacy protection is required. In real world applications, controlled and limited information disclosure is usually acceptable. By lowering the restriction on privacy protection, we can achieve a much better performance. We have developed an efficient algorithm with acceptable privacy protection to maintain the association rules. Besides, some techniques we developed can reduce the cost in the recomputation algorithm in [10].

2 Related Work

The problem of association rule mining is to find interesting patterns among large set of data items [1]. The main focus of the problem is on mining large itemsets. An iterative approach is usually used. The k -th iteration finds all the large itemsets with size k . In [6], the problem is extended into a distributed environment. Different sites hold different individual databases. The problem is to find the global association rules. [6] points out that a globally large itemset must be locally large in some of the sites and gives an efficient algorithm to solve the problem.

[5] and [4] studied the maintenance problem of association rules and large itemsets when one needs to update the database. Old large itemsets can be used to save some effort in the new computation. [4] focuses on the maintenance when there are new transactions. [5] is a more general solution which also considers deletions of transactions. The computational cost in these maintenance algorithms is greatly reduced compared to a recomputation.

To solve the problem of association rule mining with privacy protection, some researchers take the data perturbation approaches [2]. On the other hand, [10] and [11] both proposed secure association rule mining algorithms with

cryptographic techniques. [10] studied the problem with horizontal partitioned databases, i.e., the databases have the same schema. [11] focuses on vertically partitioned databases. The parties share the same set of records with the same primary key but they have different schema. [11] can only handle the two-party case. The multiparty case is solved in [12] using secure set intersections.

We now study the problem of maintenance of association rule mining in horizontally partitioned databases using cryptographic technique. Note that none of the above work handles the maintenance problem in distributed environment with privacy protection. Although [10] can be also used in our problem by total recomputation but it wastes the effort that we have put in before. A more efficient algorithm which protects privacy as well is required.

3 Problem Definition

Let I be the set of items. Each transaction K is a subset of items, i.e., $K \subseteq I$. A transaction K contains an itemset X if and only if $X \subseteq K$. Given a support threshold $s\%$, an itemset X is said to be frequent in the database DB if and only if at least $|DB| \times s\%$ transactions contain X , where $|DB|$ is the number of transactions in DB . Given a confidence threshold $c\%$, we find association rules in the form of $X \Rightarrow Y$ where $X, X \cup Y$ are large itemsets and $c\%$ of the transactions that contain X also contain Y .

Suppose there are n sites, S_1, S_2, \dots, S_n . Each site S_i has a private transaction database DB_i , where $i = 1$ to n , all having the same schema. Each site holds a number of transactions, which is $|DB_i|$, for $i = 1$ to n . We have found the large itemsets (and the association rules) in $\bigcup_{k=1}^n DB_k$. There are r new sites, $S_{n+1}, S_{n+2}, \dots, S_{n+r}$ to join the n existing sites. Each of the new sites owns a private database DB_i , for $i = n + 1$ to $n + r$. The goal is to find the new set of association rules more efficiently than simple recomputation.

Definition 1. $X.count_i$ X S_i . X

$$\sum_{k=1}^{n+r} X.count_k \geq \sum_{k=1}^{n+r} |DB_k| \times s\%.$$

$$\sum_{k=n+1}^{n+r} X.count_k \geq \sum_{k=n+1}^{n+r} |DB_k| \times c\%.$$

$$\sum_{k=1}^n X.count_k \geq \sum_{k=1}^n |DB_k| \times s\%.$$

Privacy preserving is necessary in our data mining process. Assume all the parties are semi-honest, i.e. each party follows the protocol with the exception that it keeps a record of all its intermediate messages during the execution of the protocol. The formal definition of private multiparty computation in the semi-honest model can be found in [8]. A computation is secure if at the end of the computation, no party (site) knows anything except its own input and the results. Some limited information disclosure is allowed practice as tradeoff between privacy and efficiency.

The input of our problem is the private databases in the sites and the old results (e.g., old large itemsets) in the old sites. Note that the old results are only known to each old site but not the new sites. The support and confidence thresholds are known to all sites. The result of our solution is the new set of large

itemsets and association rules. We should not disclose any other information to any other parties in the mining process apart from these inputs and results (and the limited information disclosure).

4 Secure Protocol Utilities

There are several developed secure protocols which help us solve some of our sub-problems. More details of these protocols can be found in [7][13][9].

Secure Sum. Suppose there are n sites, $\{S_1, S_2, \dots, S_n\}$, where $n \geq 3$. Each site S_i holds a value v_i . Our goal is to securely find out the sum of these values $s = \sum_{i=1}^n v_i$, which has a known upper limit m , i.e., $s \leq m$. Assume S_1 is designated as the master site. First, S_1 generates a random number R which is in the range $[1, m]$. S_1 adds its value v_1 with R and sends the value $(v_1 + R) \bmod m$ to S_2 . Then, for the remaining sites S_j , $j = 2$ to n , S_j receives a value from S_{j-1} , which is equal to $R + \sum_{i=1}^{j-1} v_i$. S_j then adds its own value v_j to it and sends the new value $(R + \sum_{i=1}^j v_i) \bmod m$ to S_{j+1} . When the process finally goes to S_n , S_n will send the final sum $R + \sum_{i=1}^n v_i$ to S_1 . S_1 then subtracts the received value by R and gets the actual sum of all the values.

Secure Union. Each site S_i holds a set of items $I_i \in I$. The goal is to find the union of the set of items $\bigcup_{i=1}^n I_i$ for n sites without revealing the private items to any parties except the owners of the items. A commutative encryption is applied in the solution, i.e., for any permutation of order p, q , $E_{K_{p_1}}(\dots E_{K_{p_n}}(X)\dots) = E_{K_{q_1}}(\dots E_{K_{q_n}}(X)\dots)$. First, each site encrypts its own items. Next, the site sends the encrypted items to another site. When a site receives an encrypted item, it would then encrypt the item as well and send it to another site which has not encrypted the item yet. The process keeps going until all the sites have encrypted all items. Due to the property of commutative encryption, if the encrypted value is the same, it means the same item, so we can remove the duplicated items. The encrypted items will then be decrypted by the sites one by one, and we get the result we want.

Secure Comparison. Suppose there are two parties, Alice and Bob. Each of them holds a number, a and b respectively. The problem is to find out the larger number without revealing the numbers to each other. Assuming the number is bounded by n , Yao [13] suggested a generic protocol which takes a linear time complexity $O(n)$ to solve the problem. There is a more efficient protocol for solving this problem [9]. The protocol takes $O((\lg n)^2)$ time and can securely find the answer without a trusted third party using one-out-of-two oblivious transfer. The details of this protocol can be found in [9].

5 Incremental Maintenance of Association Rule Mining with Privacy Protection

We propose our solution SIMDAR (Secure Incremental Maintenance of Distributed Association Rules) to perform an incremental update to the found

association rules while protecting privacy. The group large itemsets (in old results) in the old sites, denoted L , is known to all old sites but not the new sites. Let L_k be the set with all itemsets in L with size k . No individual site knows the exact counts of these large itemsets (we will discuss it more in Sec 5.2). We take the Apriori algorithm as the framework and construct our algorithm using an iterative approach. The outline of SIMDAR is shown as follow:

1. Generating the candidate sets
2. Gathering information of candidates in the new data
3. Pruning itemsets and finding large itemset
4. Repeating steps 1-3 until no more large candidates can be found
5. Checking association rules

5.1 Candidate Set Generation

The aim of candidate set generation is to get a minimized list of itemsets C_k which may be large globally in the k -th iteration.

For the first iteration $k = 1$, we do not have enough information to conclude if an itemset must be small. So we simply include all the itemsets, $C_k = I$ where I is the entire set of items. If $k > 1$, a globally large itemset must be locally large in some new sites or it is group large in the old sites [5]. The Apriori property says that if some of the subsets with size $k - 1$ of an itemset X are small, X cannot be large (proved in [1]). So, we first generate local candidates in the new sites, $C_k^i = \text{Apriori_gen}(L'_{k-1} \cap LL_{k-1}^i)$ at S_i where L'_{k-1} is the new globally large itemsets with size $k - 1$ and LL_{k-1}^i is locally large itemset at site S_i for the $(k - 1)$ -th iteration. One of the old sites prepares the group large itemsets from the old large results, $C_k^{old} = \text{Apriori_gen}(L'_{k-1}) \cap L_k$. Then we perform a Secure Union to find the candidate sets $C_k = \bigcup_{k=1}^n C_k^i \cup C_k^{old}$.

5.2 Information Collection and Storage

We can determine if an itemset is large without knowing the counts of itemsets by combining Secure Comparison and Secure Sum [10]. Suppose there are n sites, S_1 to S_n , involved in the Secure Sum process of finding count of X . S_1 is the master site holding the generated random protecting key R_X . The last site S_n gets the sum with random key added, $\sum X.count_i + R_X$, in the last step. Then, if we want to check if $\sum X.count_i \geq c$ for some c , we can perform a Secure Comparison between $\sum X.count_i + R_X$ and $R_X + c$ at S_1 and S_n respectively. Hence we can know if X is large while $\sum X.count_i$ is protected. Instead of summing the support counts, excess count of each item is summed in [10].

Definition 2. $\begin{matrix} |DB_i| & & DB_i. \\ X & S_i & s\%, \\ X.excess_i, & X.count_i - s\% * |DB_i|. \end{matrix}$

Each site, instead of giving $X.count_i$, supplies $X.excess_i$ as the input to Secure Sum. We can check for large itemsets by checking $\sum X.excess_i \geq 0$ using Secure

Comparisons. However, since we have not calculated the exact value, problems arise if we want to reuse this value. We need to store the information in a secure and efficient way.

A simple approach in [10] is that each site stores its local counts of globally large itemsets on its own and we can perform a Secure Sum whenever we need the (excess) count of an itemset. It takes both more time and space compared to normal storage without concerning privacy. Actually, we can store the counts securely in a more efficient way. In our case, the excess counts are all generated by Secure Sum. The master site S_1 keeps the value of random number key R and the last site S_n stores the protected count $X.excess + R$. These information are some intermediate messages which the sites may store it on its own. Thus, storing the protected excesses and the keys does not introduce any further privacy problem. This requires less space and less access time. Sections 5.3 and 5.4 will discuss how we can use such protected values in the future securely.

5.3 Pruning Mechanism and Checking Large Itemsets

A globally large itemset must be locally large in some new sites or group large in the old sites. After we have got the candidate set C_k , each new site scans database to get the counts of candidates. We can first prune away itemsets which are locally small in all new sites and not large in the old sites. One site from old sites and all the new sites take part in a Secure Union process. The inputs to the Secure Union process are the locally large itemsets of the new sites and the originally large itemset L_k . After the Secure Union process, we have a possibly smaller candidate itemsets, C'_k .

The handling of the old large itemsets and the new potential large itemsets is different. This eventually requires us to partition the candidate sets into two groups of itemsets. Define $P'_k = C'_k \cap L_k$ and $Q'_k = C'_k - P'_k$. For P'_k , we just add the excess counts in the new sites to the stored excess counts. For Q'_k , we first sum the excess in the new sites. If an itemset is group large in new sites, we scan the databases in old sites. However, as new sites do not know L_k , new sites cannot distinguish the groups P'_k and Q'_k and we should not reveal this piece of information to the new sites. Our idea is to make the new sites have the same view in our algorithm for all itemsets. We perform a merged process consisting of four phases to find large itemsets and prune unnecessary candidates.

Phase 1: Pick up participants. All the new sites will join and we will pick two old sites to join. For itemset $X \in P'_k$, the two old sites are the sites holding the protected excess count and the protecting key. They can supply stored excess count of X . If $X \in Q'_k$, the two old sites are just randomly picked among all the old sites. These two sites are picked just to make the process looks like the same to the new sites.

Phase 2: Collect information. We perform a Secure Sum with all the selected participants in phase 1. One of the participants from old sites is assigned as the master site. The other participant from old sites cannot be the second site or the last site of Secure Sum. So, we will have a new site holding the protected sum and an old site holding the protecting key. The new sites use

their $X.excess_i$ as the inputs to Secure Sum. If $X \in P'_k$, the two old sites use their stored protected excess count and the protecting key as the inputs. For $X \in Q'_k$, the two old sites do not have the count for X in the old sites, and will add 0 to the sum which does not affect the sum. Let Sum_X be the sum from the Secure Sum. The last site gets $Sum_X + R_X$ where R_X is the protecting key hold in the master site. Note that, Sum_X means global excess count if $X \in P'_k$, or excess count in new sites otherwise.

Phase 3: Pruning. We check whether $Sum_X \geq 0$ by a Secure Comparison. All itemsets that cannot pass this condition are pruned. If $X \in P'_k$, we are checking whether X is globally large. If $X \in Q'_k$, we are checking if X is group large in new sites.

Phase 4: Final check. The itemsets which passed the pruning with the corresponding protected summed values are passed to old sites apart from the two old sites participated in previous phases. The remaining part will be done by the old sites. Let C''_k contain the candidate sets after the second pruning. For each itemset $X \in C''_k$, if $X \in P'_k$, X is large already. If $X \in Q'_k$, the itemset is broadcasted among the old sites requesting a scan for its count. Another Secure Sum is used to get the total excess count. Suppose S_1 is the master site holding the key R_X of last Secure Sum. An old site S_k receives the protected excess of X , $Sum_X + R_X$ from the new sites. Secure Sum continues, starts from S_k until the last site S_l . S_k also adds another random number R'_X to prevent the two old sites that have joined the previous Secure Sum process from discovering a partial excess of a group of sites. S_k sends R'_k to S_l so that S_l can find the protected actual excess count $\sum X.excess_i + R_X$. Finally, we can then check whether X is a globally large itemset by comparing the protected excess and the protecting key at S_l and S_1 .

Lemma 1.

$$C_k \qquad L'_k$$

- 1.
- 2.
- 3.
- 4.
- 5.

. According to definition of secure computation in [8], a computation is secure if the view of each party during the execution of the protocol can be effectively simulated given the results, the listed leaked information (which is acceptable), and the input of that party. So, we only need to show the existence of such a simulator for each party in our proof. Secure protocols in Section 4 are not discussed here. They are assumed to be secure in our proof.

Before the pruning, the sites find C'_k by a Secure Union. The old sites can find C'_k by using a set difference on C_k and the set of itemsets in point 1. The new sites can find C'_k by using a set difference on C_k and the set of itemsets in point 4. We will prove the security of pruning phase by phase.

1. The only communication in this phase is to tell all the sites which two old sites will join the later operations. Each old site, for an itemset $X \in L_k$, knows that the process will pick the sites which hold the protected excess information of X . Otherwise, the old site can simulate as we pick any two old sites randomly. This random picking simulation can also be applied in the view of each new site.

2. This phase consists of a Secure Sum only.

3. C''_k is generated by pruning some itemsets in C'_k . The old sites can generate C''_k by using a set difference on C'_k and the set of itemsets in point 2. One of the old sites also receives the summed information from the new sites. Suppose the arithmetic in Secure Sum is mod m . The site randomly chooses a real number in $[0, m)$. As the summed value is protected by a random number and these two numbers also fall in the range $[0, m)$ (after mod m), the view of the party and the output of the simulator are computationally indistinguishable. The probability of seeing a specific value in both is equal.

The new sites can construct a simulated C''_k , denoted C''_{k-sim} . the simulator first add all itemsets in L'_k into C''_{k-sim} . Next, it adds the itemsets in point 5 to C''_{k-sim} . This gives us a simulated C''_k for the new sites. Note that, the simulator in the new sites ends here.

4. After the new sites gives C''_k to the old sites and the corresponding protected sums, the old sites can divide C''_k into two groups. For an itemset $X \in L_k$, X is automatically add to the large itemsets.

For an itemset $X \notin L_k$, a Secure Sum is performed followed by a Secure Comparison. We can create a simulator for these two protocols in a similar way as proofs in these two protocols. For $X \in L'_k$, the simulation gives a positive result in Secure Comparison. For X in itemsets in point 3, the result in Secure Comparison in the simulation is negative. Note that the two old sites in phase 1 also join the Secure Sum. However, as S_k will add another random number to the sum, each party in Secure Sum gets a value which has two or more variables in $[0, m)$. So, the simulator (which randomly picks a number in $[0, m)$) gives an indistinguishable view of a party. □

5.4 Checking Association Rules

First, we sum up the database size in each site by a Secure Sum process. All the new sites join in this Secure Sum. As we may have already stored the total database size in the old sites, the two old sites storing the protected total database size can be representatives and give the protected total database size and the protected key as input to Secure Sum. The total database size is used to find out the confidence of an association rule from excess counts of itemsets. Let $T = \sum_{i=1}^{n+r} |DB_i|$. Let R_T be the generated random key in the Secure Sum process to protect the total database size. At the final stage, a site S_u holds the

value of $T + R_T$ and another site S_v holds R_T . S_u and S_v will not store the protected excess count and the protecting key of an itemset.

When we check an association rule $X \Rightarrow Y$, we need to check whether $\frac{Z.count}{X.count} \geq c\%$ where Z is $X \cup Y$. However, what we have got for X and Z are $X.count - s\% * T + R_X$, R_X , $Z.count - s\% * T + R_Z$, and R_Z . These values are available in two to four sites. Besides, we also get $T + R_T$ and R_T from S_u and S_v . We may rephrase $\frac{Z.count}{X.count} \geq c\%$ as follows:

$$\begin{aligned} \frac{Z.count}{X.count} \geq c\% &\iff Z.count - c\% * X.count \geq 0 \\ &\iff (Z.excess + R_Z) - c\% * (X.excess + R_X) - R_Z + (c\% * R_X) \\ &\quad + s\%(1 - c\%) * (T + R_T) + (c\% - 1) * s\% * R_T \geq 0 \end{aligned}$$

The six terms in the final inequality can be derived from the stored values we have. Thus, we can perform a Secure Sum process to add all these six terms together and check if the sum is greater than zero.

6 Experiments

We carried out a set of experiments to analyze (i) the efficiency of the algorithm, and (ii) the overhead introduced with privacy protection. We take CPU time as the measurement of cost and do not take idle time into account. We implemented two programs for comparisons. One program, which we call it SEC, is a simple privacy preserving mining algorithm without considering incremental maintenance. SEC (re)computes the new set of large itemsets and association rules securely. Efficiency of our algorithm is measured by comparing the CPU time used by SEC and our algorithm. We present the efficiency as the reduction ratio of SIMDAR over SEC. Another program we implemented, which we call it MAN, is a maintenance algorithm without privacy concerns. MAN uses simple messages for communication instead of secure protocols. Overhead of privacy protection of our algorithm is measured as the difference between CPU time consumed by MAN and our algorithm.

Definition 3. $t_{SEC} (. t_{SIM}, t_{MAN})$
 $(. ,)$.
 Eff_{PP} ffi $Eff_{PP} = \frac{t_{SEC} - t_{SIM}}{t_{SEC}}$.
 OH_{PP} $OH_{PP} = t_{SIM} - t_{MAN}$.

In the experiments, each site was simulated using a stand-alone computer (Dell Optiplex Gx240SD Pentium 4 1.7 GHz computers running Linux). We generated a large number of transactions using IBM synthetic data generator [3]. We supplied three parameters to the data generator: (i) number of transactions, (ii) number of items, and (iii) the length of maximal potentially large itemsets. We used the default values for other parameters. In order to introduce a larger difference between the large itemsets in the old sites and that in the new sites, we

set the length of maximal large itemsets of databases in the old sites to be 6 and that in the new sites to be 8. We set the number of items to 1000.

We performed three sets of experiments with the following varying factors:

Database sizes. The database size varies from 200K to 1M. We have 5 old and 5 new sites. The support threshold is set to 2%.

Support thresholds. The support threshold varies from 0.75% to 2%. We have 5 old and 5 new sites. The database size is set to 500K.

Ratios of the number of old sites to that of the new sites. We have in total fifteen sites. The number of old sites increases from 3 to 12 linearly while the number of new sites decreases from 12 to 3 respectively. The support threshold is 2%. The database size is 500K.

6.1 Database Sizes

Figure 1 shows the average CPU time in the new sites and in the old sites in this experiment. It shows that the CPU time is approximately linear to the database size for both new and old sites. All programs have a similar CPU time in new sites but SIMDAR and MAN have a lower CPU time in old sites. It shows that the incremental maintenance technique can efficiently reduce the CPU time for old parties. Eff_{PP} varies from 59.6% to 63.8% in the old sites. MAN has the lowest CPU time in all cases because both SEC and SIMDAR have implemented secure protocols like Secure Comparison which induce additional cost. However, as the major workload actually goes to the scanning of databases, the additional cost of secure protocols is relatively low. OH_{PP} takes 2.5% to 8.8% of the CPU time in the new sites and 2% to 13% of the CPU time in old sites.

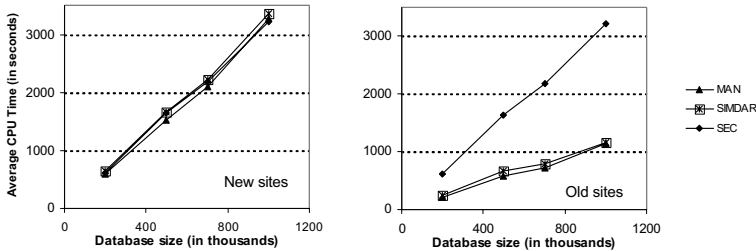


Fig. 1. Average CPU time with varying database size in each site, 5 old sites and 5 new sites, 2% support threshold

6.2 Support Threshold

Figure 2 shows the average CPU time in the new sites and in the old sites in this experiment. MAN and SIMDAR perform better than SEC in the old sites. As the support threshold decreases, the gap between SEC and SIMDAR increases significantly. Eff_{PP} increases from about 63.4% (3% support threshold) to about 75.4% (0.75% support threshold). When the number of large itemset increases, the number of candidate sets generated is exponentially increased. However, a

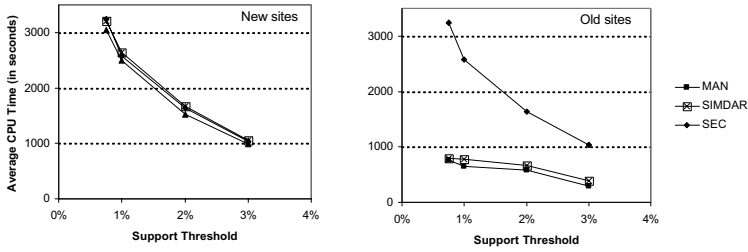


Fig. 2. Average CPU time with varying support threshold, 5 old sites and 5 new sites, 500K transactions in each site

large portion of candidate sets is pruned. OH_{PP} takes 5.4% to 8.9% in the new sites and 4.1% to 25.0% of the CPU time in the old sites.

6.3 Ratio of Old Sites to New Sites

Figure 3 shows the average CPU time in the new sites and in the old sites in this experiment. The average CPU time decreases when the number of old sites increases for all three programs in all sites. It is because the total number of large itemsets in the old sites is fewer than that in the new sites. Thus, when the majority is the old sites with fewer large itemsets, the total number of large itemsets and candidates decreases.

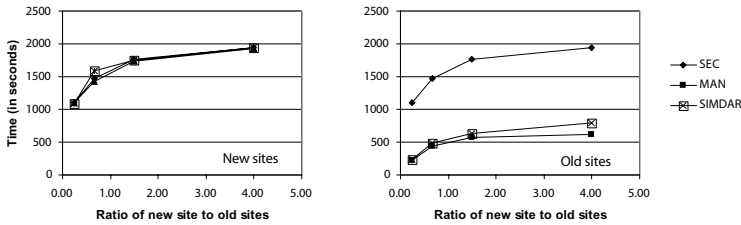


Fig. 3. Average CPU time in the old sites with varying ratio of number of old sites to new sites, 15 sites in total, 2% support threshold, 500K transactions in each site

Eff_{PP} increases when the proportion of old sites increases. Eff_{PP} at the ratio of 3 old sites to 12 new sites is 58.8%. When the ratio increases to 12 old sites to 3 new sites, Eff_{PP} significantly increases to 79.1%. It is because we have already known the large itemsets of the old sites which become the majority. When the number of old sites increases, it becomes more difficult for new sites to add new changes to the old results.

7 Conclusions

We studied an efficient algorithm to solve the maintenance problem of adding new sites. The developed method SIMDAR can successfully reduce the number

of candidate sets required to be scanned in the old sites. Experimental results showed that our algorithm SIMDAR can effectively reduce the workload of the old sites while the cost in the new sites is almost the same as in a recomputation. The entrance cost for a new party is not reduced much but the maintenance cost for an old party is much lower.

After working on the case of addition of new sites, we are now studying other cases: (i) removing sites, and (ii) updates of databases in one or more old sites. It is also challenging to consider the combination of all these cases, which is more likely to happen in practice.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: VLDB, Santiago, Chile (1994)
2. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: SIGMOD, Dallas, Texas (2000)
3. IBM Almaden Research Center. Synthetic data generation code for association and sequential patterns
4. Cheung, D.W., Han, J., Ng, V.T., Wong, C.Y.: Maintenance of discovered association rules in large databases: An incremental updating technique. In: ICDE, Washington, DC, USA (1996)
5. Cheung, D.W., Lee, S.D., Kao, B.: A general incremental technique for maintaining discovered association rules. In: Database Systems for advanced Applications, Melbourne, Australia (1997)
6. Cheung, D.W., Ng, V., Fu, A.W., Fu, Y.: Efficient mining of association rules in distributed databases. Special Issue in Data Mining, IEEE Transaction on Knowledge and Data Engineering 8(6) (December 1996)
7. Clifton, C., Kantarcioglu, M., Vaidya, J., Lin, X., Zhu, M.: Tools for privacy preserving distributed data mining. In: ACM SIGKDD Explorations Newsletter (2002)
8. Goldreich, O.: Foundations of Cryptography, May 2004, vol. 2. Weizmann Institute of Science, Israel (2004)
9. Ioannidis, I., Grama, A.: An efficient protocol for Yao's millionaires' problem. In: HICSS, Waikoloa Village, Hawaii (2003)
10. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. IEEE Trans. Knowledge Data Eng. 16(4) (July 2004)
11. Vaidya, J., Clifton, C.: Privacy preserving association rule mining in vertically partitioned data. In: KDD, Edmonton, Alberta, Canada (2002)
12. Vaidya, J., Clifton, C.: Secure set intersection cardinality with application to association rule mining. Journal of Computer Security 13(4) (November 2005)
13. Yao, A.C.: How to generate and exchange secrets. In: Proceedings of the 27th IEEE Symposium on Foundations of Computer Sciences (1986)

SEM: Mining Spatial Events from the Web*

Kaifeng Xu, Rui Li, Shenghua Bao, Dingyi Han, and Yong Yu

Department of Computer Science & Engineering
Shanghai Jiao Tong University
Shanghai, 200240, P.R. China

xukaifeng1986@sjtu.edu.cn, {rli, shhbao, handy, yyu}@apex.sjtu.edu.cn

Abstract. This paper is concerned with the problem of mining spatial events from the general Web. General search engine is inconvenient when searching vertical information (e.g., locations, experts) since it is designed for general purpose. For example, when finding *the battlefields of World War II*, listing the Web pages by relevance is not enough to tell users the spatial information clearly. A categorized result along with a map indicating these battlefields would be much easier to read. To present such a result, we propose a novel algorithm called Spatial Event Miner (SEM) to mine spatial event information from the general Web. Given a simple keyword query, SEM first collects and ranks a set of relevant locations from the Web. Then, to describe the events happened in the collected locations, SEM detects and sums up salient phrases as event topics from the context of these locations. For each specific location, the hottest event topics are also listed for quick understanding. Finally, a clear spatial distribution on the events of a given query is presented to the users. A prototype system based on SEM is also implemented. Preliminary experimental results on a set of 40 queries show that the proposed approach can capture the spatial event information effectively.

Keywords: Location and topic extraction, spatial events mining, evaluation.

1 Introduction

Search engine is a great tool for people to effectively access useful information from vast content on the Web. Given a query, it enables navigating a list of relevant Web pages, which provides shortcuts to the endless Web content. Although quite helpful, it is still inconvenient when searching vertical information, such as spatial information, since it is designed for general purpose. For example, when finding *the battlefields of World War II*, listing the Web pages by relevance is not enough to tell the spatial information clearly. The users have to read most of the results to get a general view about what happened in the battlefields and where are these locations. A categorized list along with a map indicating the locations would be much easier to read.

Literally, searching spatial information is a conventional task in today's cyber world. Quite a lot of people have greater needs to get the knowledge of the spatial

* This work is supported by National Natural Science Foundation of China (Grant Number: 60473122).

distribution on a certain topic. To name a few, they can be scientists who concern about where greenhouse effect and ozone depletion occur, or students working on historical researches to find out where some famous battles take places or company leaders who must keep one eye on the oversea markets where their products sell best. All of them can save much time if there exist any spatial information search services.

Pioneer researchers have already studied this kind of service on traditional data sources. For example, Smith detects events by hand with date and location information from historical documents to facilitate the review of past events [8, 9].

Unfortunately, there is no such service on the Web. Since “the Web is a sensor of the real world” [16], a mass of information, including news and thorough discussions usually appear in the Web when some hot events happened in the real world. It is actually a better and more useful source for spatial information search task.

Given a query, there are three tough problems when fetching spatial information from the general Web.

1. How to effectively retrieve the geographic information, ranging from continents to specific locations, from Web content without human labeling? I.e. How to make computer automatically recognize locations from relevant texts? Some ad-hoc locations are not known before the event happens. The location names are also evolving. Building a gazetteer (or location dictionary) is not enough.
2. How to approximately describe events with Web texts? Browsing all the Web pages to find out the events is time-consuming. Several keywords describing the events would be a simple and quick way for the users to know the relevant events.
3. How to draw a clear and understandable spatial distribution of the relevant events? For a lot of queries, like *the battlefields of World War II*, there are many relevant locations across the world. Instead of listing these locations, there should be a visual map to show them for clarity. The key point is to locate them on the map.

In this paper, we name the above problems as Location Identification Problem, Event Detection Problem and Spatial Presentation Problem, respectively. To solve the first two problems, we introduce a new algorithm, Spatial Event Miner (SEM), which is composed of two sub-algorithms, *Event Location Retrieval* (ELR) and *Event Topic Mining* (ETM). In ELR, a hybrid approach including gazetteer and pattern based method is employed by SEM to extract a set of associated locations of the input query, and rank them according to the correlation. For example, the successful extraction of the locations range from “America” to “World Trade Center” for the query “September 11 2001” demonstrates the achievements because World Trade Center is not relevant to the query before the day. In ETM, statistical features are applied by SEM to detect and sum up salient phrases as event topics from the contexts of these locations. The summary of “terrorist attacks” for the same query shows the achievement on Problem 2. Finally, to solve the last problem, we design an easy-to-use interface with retrieved locations classified by spatial scope. With the help of illustration, users can understand the spatial distribution of the events completely.

A prototype system based on SEM is implemented as shown in Figure 1. The four numbered components are: ①the event distribution classified by spatial scope. ②the hottest event topics relevant to the selected location. ③some event descriptions for the selected location. ④geographic information using an online map service.

Search
Location

PAKDD 2008 Search Location

PAKDD 2008 2 result(s) for PAKDD 2008 (1.609 seconds)

HOTTEST: Osaka, Japan

KEY WORDS: pacific asia conference, mining, anu data, knowledge discovery, submission deadline, conferences

▶ **Contient**

▼ **Country**
Japan

▼ **City**
Osaka

▶ **More**

1

Suzuki's Organizations.
- [PC Chair, PAKDD-2008 - The Twelfth Pacific-Asia Conference on Knowledge Discovery and Data Mining, **Osaka**, Japan, May 20-23, 2008

3

Florian Verhein's Conference Links.
PAKDD 2008: The 11th Pacific-Asia Conference on Knowledge Discovery and Data Mining, **Osaka**, Japan, 20-23 May, 2008. ICDE 2008: The 24th International.

BIRL - Bioinformatics Research Lab , Chiang Mai University - Home.
PAKDD 2008 [The Pacific-Asia Conference on Knowledge Discovery and Data Mining], **Osaka**, Japan, 20-23 May 2008. WCCI 2008 [IEEE World Congress on.

Manabu Sassano's Log (in Japanese) - June 2007
- [PAKDD 2008: 20 - 23 May 2008 **Osaka**, Japan, 06/08/2007 - Coling 2008: Submission deadline main conference: 30 March, 2008. ■ プロシーディング: かも ICML.

Map Satellite Hybrid

Osaka
port city on southern Honshu on Osaka Bay, a commercial and industrial center of Japan

4

Fig. 1. A sample snapshot of SEM on the query “PAKDD 2008”

To evaluate the effectiveness of the SEM algorithm, 40 queries in different fields are collected and tested. Thousands of spatial events are discovered. The results show that with the help of SEM, people can understand the spatial events more effectively.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 generally describes the proposed method. Section 4 discusses the algorithms of *Event Location Retrieval* and *Event Topic Mining*. The experimental results are presented in Section 5, followed by the discussion on conclusion and future work.

2 Related Work

2.1 Event Location Retrieval

Event location retrieval, so-called *geoparsing*, is widely applied in various fields including scientific research, vertical search engine, personal annotation mining, etc.

Most published algorithms in this category were based on various NLP heuristics. Li et al. presented in [5] a rigorous 5-step algorithm that was typical of many such publications. The authors reported a 93.8% precision on news and travel guide data. Improved algorithms were also proposed by Bilhaut et al. [2] and Smith et al. [10] on the data of French documents and digital library of historical texts respectively. The suggestion to apply such techniques to Web pages was first made by McCurley in [6]. His method, however, depends heavily on information such as postal tracts and phone directories that is much harder to come by in most parts of the world.

More seriously, all the methods mentioned above can only identify locations that are in the gazetteer list. It would be quite absurd if “World Trade Center” was not

found for query “September 11 2001” only because it was not in the gazetteer. In this paper, a hybrid approach involving gazetteer algorithm and pattern based extraction is introduced into SEM for not only improving the accuracy of the identification but also improving the recall, i.e., finding the locations not exist in the gazetteer.

2.2 Event Topic Mining

Event topic mining is to find out *something that happens at a particular time and place* (by Allan et al. [1]). Since 1996, it has been well studied in the community of Topic Detection and Tracking (TDT). Allan et al. focused on a strict on-line setting [1]. Yang et al. studied the problem of retrospective and on-line event detection [13], a two stage method was also proposed by them for automated detection of chronologically ordered documents [14]. However, as occurring in a certain location, most TDT systems do not directly take geographical location into account. More recently, a new approach of detecting event was put forward by Zhao et al. from the evolution of click-trough data [16]. Instead of their special event detection, we focus on detecting events from the general Web pages with respect to the users’ query.

The summary of events has also been studied for a long history. The work we are concerned is event summarization over locations. Smith studied the subject on mining spatio-temporal events from historical documents [8, 9], which summarizes events with date and location information. Tye et al. focused on the problem of extracting location and event semantics for tags [12] that are assigned to photos on *Flickr*¹, a popular photo sharing website. Similar subjects on mining spatio-temporal events were also studied from RSS feeds [3] and weblogs [7]. Differently, we use phrases automatically retrieved from Web pages as summaries.

3 Algorithm Overview

The purpose of SEM is to help users understand the spatial distribution and the detail information of the events relevant to a given query. The snippets returned from the search engine will be our data source since search engine can collect query related data from the Web easily and widely.

Given a query Q , SEM will run as the stream line in Table 1. In Stage 1, since few geographic names can be retrieved by directly querying search engine, we expend the query to collect more spatial information. The details of Stage 1 will be discussed in Section 4.1. In Stage 2, we extract the locations from collected snippets. Some features are introduced to rank these locations and decide m most credible locations where some events actually happened. The details of Stage 2 will be discussed in Section 4.2. In Stage 3 we first collect each location’s snippets and then take noun phrases in the snippets as candidate event topics. To extract those events that are most likely to be of interest to the user, we rank and get n most salient event topics. The details of Stage 3 will be discussed in Section 4.3. Finally, SEM outputs a list of classified locations, together with their associated event topics and descriptions.

¹ <http://www.flickr.com>

Table 1. Spatial Event Miner (SEM) algorithm

Algorithm SEM(Q)	
Input	Given the search phrase as query Q .
Stage 1	Snippet Acquisition
i)	Expand Q with prepositions, one preposition at a time.
ii)	Feed each expanded query to a search engine, store the returned snippets together.
Return	A snippet collection $S = \{s_1, s_2, \dots, s_l\}$.
Stage 2	Location Extraction
	For each s_i Do
i)	Extract original location strings from s_i .
ii)	Rank these location strings by some correlation features and take the top m to get a location collection.
Return	A location collection $P = \{p_1, p_2, \dots, p_m\}$, each p_i has one or more contexts in S .
Stage 3	Topic Extraction
	For each p_i Do
i)	Select all s_j from S which contain p_i as descriptions S_{p_i} .
ii)	Extract nouns and noun phrases as candidate event topics from S_{p_i} .
iii)	Rank all the candidate topics, extract top n based on a set of statistical features.
Return	An event topic collection $T = \{t_1, t_2, \dots, t_n\}$, each t_i is represented by a noun phrase.
Output	Display to the user with the well-organized events including locations, topics and descriptions.

4 Mining Spatial Event from the Web

4.1 Data Retrieving and Query Expansion

To prepare a data source of a given query from the Web, we use search engine. For example, we can directly query a search engine with “September 11 2001”, snippets such as “September 11, 2001 attacks - Wikipedia, the free encyclopedia” and “The September 11 Digital Archive” can be retrieved.

However these snippets contain few location names. To retrieve more location names for a given query, we define a set of linguistic patterns for retrieving the pages which may contain more location information. In our common sense, geographic phrase often follows a preposition. Therefore, 19 frequently-used location prepositions, which are divided into 3 groups as follows, are used by us to expand the query in total. Some example snippets are also listed for each group.

P1: such as *in*, *on* and *at* which express an accurate location.

Snippet: the history of the September 11, 2001 attacks in New York.

P2: such as *near*, *beside* and *around* which express an approximate location.

Snippet: Located near the Afghan border.

P3: such as *above* and *through* which express other location prepositions

Snippet: Department records of 911 calls from the World Trade Center.

We get top l snippets for each expanded query from search engine and put them together to form a snippet collection S .

4.2 Mining Locations for a Query

The purpose of this step is to extract the location names from the collected snippets.

To mine the useful locations from these Web page snippets which are unstructured, noisy and changeful, we employ a gazetteer to recognize the common locations, and a pattern-based approach to recognize the locations that are not listed in the gazetteers.

Gazetteer is used as a dictionary of common geographic vocabulary, which can help to extract the general locations names accurately. For example, “America” and “New York” can be easily extracted by gazetteer for query “September 11 2001”.

However, it is not enough for a gazetteer to detect the whole geographic space since many location names are seldom known to the public or being created and evolving every day. In order to detect these locations more freshly and specifically, we utilize 117 patterns divided into 2 groups as follows to give the unmatched noun phrases a second chance. “World Trade Center”, for example, can also be extracted for the same query mentioned above while gazetteer misses it.

Q1: man-made pattern such as *XX Building, XX Hospital, XX School* etc.

Q2: nature pattern such as *XX Sea, XX Island, XX Falls* etc.

Now we have collected a set of candidate locations such as “World Trade Center”, “Los Angeles” and “Asia”. Apparently, “Asia” is not strongly relevant to the query. It is extracted because some Asian media has reported the tragedy. Therefore, a ranking method should be employed. It should assign a higher rank to a more relevant location. I.e. “World Trade Center”, as where the tragedy happened, should be ranked first. “Los Angeles”, as the destination of the accident plane, should also be ranked high.

Location Relevance, denoted by $LR(p)$, is defined to calculate the correlation between the given query Q and the extracted location p . It can be expressed by:

$$LR(p) = \frac{|Q \cap p|}{|Q|} \left(1 - \frac{|Q \cap p|}{|p|}\right), \quad (1)$$

where $|p|$ means the hits number for querying p from the search engine and $|Q \cap p|$ means the number for querying Q and p . We place a high weight on the intersection size between Q and p and take the consideration of p 's size as compensation.

Location Confidence, named as $LC(p)$, is used to emphasize the location frequently appear after the preposition. For example, given the query “September 11 2001”, “Asia” and “World Trade Center” have similar total appearances. However, “World Trade Center” appears more frequently after the preposition, which indicates “World Trade Center” connects with the query more closely.

$$LC(p) = T \left(2^{\frac{TAP}{T}} - 1\right), \quad (2)$$

where T denotes the total times of p appearing in the snippet collection S while TAP means the times that p appears after the preposition. We focus more on the TAP for its confidence and an exponential function is used for this purpose.

We combine these two properties with multiplication for their both necessity.

$$Score(p) = LR(p) \cdot LC(p). \quad (3)$$

Finally we rank all the extracted locations by their score values and pick up the top m to form a location collection P .

4.3 Summarizing Topics for Query

The purpose of this step is to mine the event topics from the extracted locations. Due to the high complexity of the condition that varied events rarely share the common phrases, we need acquire a better understanding of nature language to find event topics. Yet, by our observations as follows, the event topics extraction can be solved by a salient phrase ranking method [15] based on existing data mining techniques.

1. The pages which contain both the given query and its locations often talk about the event happened in this location. For example, the pages containing "September 11 2001" and "World Trade Center" often talk about "attack" or "terrorism".
2. The phrase which is referred to the event topic also is a salient phrase in the data set. For example, "attack" may have more frequency than other terms.
3. The meaningful topics are more likely to be nouns or noun phrases. For example, "national commission" is a meaningful topic for "September 11 2001".

We parse the snippets with an NLP tool and use nouns and noun phrases as our candidates for extracting topics. To demonstrate meaningful event topics of a location, we extend the existing salient phrase ranking method on candidate locations by following the statistics features.

Topic Frequency / Inverted Snippet Frequency (TFISF) is defined in the same fashion as *TFIDF* and could be expressed by

$$TFISF(topic) = TF(topic) \cdot \log \frac{|S|}{SF(topic)} \quad (4)$$

where $TF(topic)$ is the total count of $topic$ in all the snippets S and $SF(topic)$ is the number of the snippets containing $topic$.

Intra-Cluster Similarity (ICS) [15] is the average cosine similarity between $topic$'s associated snippets and their centroids. It is defined as:

$$ICS(topic) = \frac{1}{SF(topic)} \sum_{s \in S \wedge topic \in s} \cos(s, O(topic)) \quad (5)$$

$$O(topic) = \frac{1}{SF(topic)} \sum_{s \in S \wedge topic \in s} s \quad (6)$$

Snippet s and $O(topic)$ are represented as their vector forms in *Vector Space Model*.

Cluster Entropy (CE) [15] is used to measure the distinctness of a $topic$. $P(t|topic)$ is the probability of term t occurring in the documents where $topic$ also occurs

$$CE(topic) = - \sum_{t \in C(topic)} P(t|topic) \log P(t|topic) \quad (7)$$

$$P(t|topic) = \frac{| \{s \mid s \in S \wedge topic \in s \wedge t \in s\} |}{SF(topic)} \quad (8)$$

Topic Independence (IND) is used to measure the independence of a $topic$ in [4]. We confirm $topic$'s independence when its left and right context is random. The following is the equation for IND_{RorL} which is the independence value for $topic$'s left or right context, where $0 \cdot \log 0$ is defined to be 0:

$$IND_{RorL}(topic) = - \sum_{t \in RorL(topic)} \frac{TF(t)}{TF(topic)} \log \frac{TF(t)}{TF(topic)} \quad (9)$$

$$IND(topic) = \frac{IND_R(topic) + IND_L(topic)}{2} \quad (10)$$

Spatial Salience (*SS*) is implied by the correlation between variable *T*, the presence or absence of *topic* and variable *P*, the places contained in snippets.

Suppose that there are no relationship between *T* and *P*. In the statistics [11] employed by *SS*, suppose that $\{p_1, p_2, \dots, p_m\}$ denote the set of locations we have extracted, a_i denotes the count of snippets containing both p_i and *topic*, b_i denotes the count of snippets containing p_i but not *topic*. The equation for the *SS* statistics is:

$$SS(topic) = 2 \cdot \sum_{i=1}^m a_i \cdot \ln \frac{a_i}{\frac{A(a_i+b_i)}{A+B}} \quad (11)$$

We use the linear combination of the above five properties as the salience score of topic *t*:

$$Score(t) = w_0 + w_1 PFISF(t) + w_2 ICS(t) + w_3 CE(t) + w_4 IND(t) + w_5 SS(t) \quad (12)$$

According to their salience scores, we sort all the candidate event topics for each place and select *n* most salient ones to form a topic collection *T*.

5 Experiments

5.1 Experimental Steps

To evaluate the effectiveness of SEM, a prototype system is implemented based on SEM. To step into the experiment, some preparations are done as follows.

1. The test query set consists of 40 queries are distributed in different fields, such as “iPhone”, “Formula 1”, “AIDS” and “Yao Ming”, which reflects varied aspects of the utility of our system. The system collects top 100 snippets for each expanded query from *Google*. The sum of snippets is 1900 per input query for 19 different pre-defined patterns. Each snippet is tagged by a NLP tool, *LingPipe*².
2. To run gazetteer based extraction mentioned in Section 4.2, a gazetteer collected from *Wikipedia*³ is used by our system which contains all of the world’s continents, countries and many of its cities (those having 5,000 inhabitants or more). It also contains states and provinces as well as many natural regions.
3. In Section 4.1, top half of the whole ranked locations are picked up. For each location, top 10 topics are chosen in Section 4.2. To calculate the weights of the linear combination mentioned in Section 4.3, 5 manual labeled queries listed in Table 2 are used which contain total 21 training topics per query and is divided into 10, 5 and 1 scores respectively. The result for w_0 , w_1 , w_2 , w_3 , w_4 , and w_5 are 2.720, 37.933, -3.597, 0.195, 0.205, and 0.002, respectively.

² <http://www.alias-i.com/lingpipe>

³ <http://www.wikipedia.org>

4. In the stage of topics ranking, the familiar situations that $Score(t_1) \geq Score(t_1 \cup t_2)$ may happen while $t_1 \cup t_2$ can describe the event more detailedly than t_1 . For example, we query “September 11 2001” and both “attacks” and “terrorist attacks” will be found for place “United States”. Although the former topic has a higher score, it should be replaced for the detailed description of the latter one.
5. *Google Maps API*⁴ is also prepared to build a friendly interface for easy understanding of spatial distribution.

More experiment data including the distribution of 40 queries, patterns for snippets retrieval and location identification and manual labeled queries for linear regression are available in online⁵.

Table 2. Manual labeled queries with example marked topics

mark ^{query}	October 1 1949	Cloning	Wright brothers	PAKDD 2007	Gulf War
10	mao zedong	therapeutic	flight	conference	oil
5	announcement	opposition	innovation	submissions	commander
1	flowers	wisdom	birthplace	references	joke

5.2 Results of Place Mining

In order to evaluate how effectively each kind of patterns defined in Section 4.2 discovers the pages containing places with comparison to traditional Web searching, we take “World War II” as an example to check out whether the returned pages contain more location names. The results are shown in the following Table 3.

Table 3. Statistics of pattern-based retrieval

	Original	Pattern-Based Retrieval			
		Total	P1(3)	P2(5)	P3(11)
#Snippets	100	1900	300	500	1100
#Locations	42	1563	302	504	757
#Loc-Snip Rate	42%	82.26%	100.01%	100.01%	68.82%

Through the experiment we can find that our approach increases the percentage of geographic information quantity in the returned snippets by 40%, especially with P1, the accurate-locations pattern and P2, the approximate locations-pattern.

In Table 4, the experiment shows the effectiveness of pattern-based extraction of specific place names described in Section 4.2, which share a 4.58% rate in the whole location extraction, Furthermore, in Table 5, it seems that these fresh locations which missed by gazetteer are usually more sensitive to the query. On the other hand, more man-made places are extracted with Q1 pattern than nature places with Q2 pattern.

⁴ <http://www.google.com/apis/maps>

⁵ http://apex.sjtu.edu.cn/apex_wiki/hay/sem

Table 4. Statistics of pattern-based extraction of a query

	Total	Q1	Q2
#Snippets	1900	1900	1900
#Locations	87	63	24
#Loc-Snip Rate	4.58%	3.32%	1.26%

Table 5. Comparison between pattern-based and no pattern-based extraction (Top 5 results)

Query	September 11 2001				
Pattern	World Trade Center	United States	New York	America	Washington
No Pattern	United States	New York	America	Washington	Iraq
Query	the Eight Power Allied Force				
Pattern	China	Beijing	Summer Palace	Taiwan	Shanghai
No Pattern	China	Beijing	Taiwan	Shanghai	Shandong

Table 6 demonstrates the results of location mining for 10 queries. For each query, at most 8 locations are displayed because of limited space. More results are also available online. We can see that our system is capable of discovering each input query’s event locations effectively and accurately.

Table 6. Event location discovery

September 11 2001	Berlin Wall	8848m	SARS	iphone
World Trade Center	Berlin	Everest	China	New York
United States	Berlin wall	Nepal	Hong Kong	China
New York	Germany	Mount Everest	Asia	Europe
America	Europe	Kathmandu	Canada	Apple Store
Washington	Soviet	India	Beijing	Canada
Iraq	West Berlin	China	Taiwan	America
Afghanistan	United States	Highest Mountain	United States	France
Pennsylvania	East Germany	Highest Peak	Singapore	Japan
Mona Lisa	World War II	Korean Hostage	PAKDD 2008	Atomic Bomb
Paris	Europe	Afghanistan	Japan	Hiroshima
Louvre	United States	Iraq	Osaka	Japan
France	Pacific	Kabul		Nagasaki
London	America	Korea		United States
Mona Lisa Restaurant	Germany	South Korea		New York
Italy	Pearl Harbor	Seoul		New Mexico
Chicago	Japan	China		Pacific
Amazon	France	Saemmul Church		Manhattan

5.3 Results of Topic Mining

Table 7 shows event topics summary for 4 input queries. Here presents top 3 locations and their top event topics respectively. Results show that the terms representing the event are meaningful.

Table 7. Event topic discovery

September 11 2001	
World Trade Center	pentagon september, twin towers, memorial Web pages
United States	terrorist attacks, national commission, full resolution image
New York	archive, september 11th, real life tragedy, special reports
Korean Hostage	
Afghanistan	killed korean hostage, held hostage, developments, taliban say
Iraq	taliban thugs, militants, executed, kim sun
Kabul	afghan government, german woman kidnapped, korean hostage talks
PAKDD 2008	
Japan	pakdd 2008, pacific asia conference, 2008 osaka
Osaka	mining, data, knowledge discovery, international conference, best
Atomic Bomb	
Hiroshima	atomic bomb dropped, 6th august, peaceful world, radiation research
Japan	cause, president truman, atomic bombs, decision, drop
Nagasaki	atomic bomb survivors, atomic bomb museum, atomic bomb attack

5.4 Case Studies

5.4.1 Time and Domain Restriction in the Topic

Table 8 shows that time and domain restriction for a larger-scale query can also improve the result. For example, historical venues will be extracted when directly querying “PAKDD”. However, if we restrict the time of “PAKDD” to “2008”, we will get the more accurate location: “Osaka” and “Japan”.

Table 8. Restriction result

PAKDD	Singapore	China	Hong Kong	Australia	Asia
PAKDD 2008	Japan	Osaka			
Tourist	London	America	India	Canada	Japan
Space Tourist	Space Station	Russia	Kazakhstan	Moscow	Europe

5.4.2 Time Evaluation in Event Search

As we know, the number of Web pages increases every minute especially when a topic becomes hot. Table 9 shows the location for query “Formula 1” at September 6 and 11 respectively. In the former result, “Spain” and “Paris” become hot because of the change of match system, while other locations are all the recent hosts. “Monza” and “Italy” pop out in the latter result due to the latest F1 match in Monza, Italy. It has come to an end with the winner of Alonso in September 9.

Table 9. Time evaluation result

Formula 1(9/6)	Istanbul	Europe	Spain	Paris	Turkey
Formula 1(9/11)	Monza	Istanbul	Paris	Europe	Italy

6 Conclusion and Future Work

In this paper, we studied the problem of event location retrieval and event topic summary from the Web. The main contributions are:

1. The proposal to study the problem of spatial event mining with the general Web.
2. The proposal of SEM, for mining location information and summarizing event topics from the general Web in which data is unstructured, noisy and changeful.
3. The implementation of SEM and a friendly interface with geographic support which offer an easy understanding for spatial distribution.

In our future work, we plan to cluster the extracted locations more hierarchically and disambiguously, based on which, more experiments will be done to evaluate SEM.

References

- [1] Allan, J., Papka, R., Lavrenko, V.: On-Line New Event Detection and Tracking. In: Proc. of SIGIR 1998, pp. 37–45 (1998)
- [2] Bilhaut, F., Charnois, T., Enjalbert, P., Mathet, Y.: Geographic Reference Analysis for Geographic Document Querying. In: Workshop on the Analysis of Geographic References, Edmonton, Alberta, Canada (2003)
- [3] Chen, Y.F., Fabbriozio, G.D., Gibbon, D., Jana, R., Jora, S., Renger, B., Wei, B.: GeoTracker. Geospatial and Temporal RSS Navigation. In: Proc. of WWW 2007, pp. 41–50 (2007)
- [4] Chien, L.F.: Pat-tree Based Adaptive Keyphrase Extraction for Intelligent Chinese information retrieval. In: SIGIR 1997, pp. 50–58 (1997)
- [5] Li, H., Srihari, R.K., Niu, C., Li, W.: Location Normalization for Information Extraction. In: Proc. of the 19th Conference on COLING 2002, Taipei, Taiwan (2002)
- [6] McCurley, K.S.: Geospatial Mapping and Navigation of the Web. In: Proc. of the 10th int. conference on World Wide Web, pp. 221–229. ACM Press, New York (2001)
- [7] Mei, Q., Liu, C., Su, H., Zhai, C.X.: A Probabilistic Approach to Spatiotemporal Theme Pattern Mining on Weblogs. In: Proc. of WWW 2006, pp. 533–542 (2006)
- [8] Smith, D.A.: Detecting and Browsing Events in Unstructured Text. In: Proc. of SIGIR 2002, pp. 73–80 (2002)
- [9] Smith, D.A.: Detecting Events with Date and Place Information in Unstructured Text. In: Proc. of JCDL 2002, pp. 191–196 (2002)
- [10] Smith, D.A., Crane, G.: Disambiguating Geographic Names in a Historical Digital Library. In: Constantopoulos, P., Sølvberg, I.T. (eds.) ECDL 2001. LNCS, vol. 2163, pp. 127–136. Springer, Heidelberg (2001)
- [11] Ted, D.: Accurate Methods for the Statistics of Surprise and Coincidence. *Computational Linguistics* 19(1), 61–74 (1993)
- [12] Tye, R., Nathaniel, G., Mor, N.: Towards Automatic Extraction of Event and Place Semantics from Flickr Tags. In: Proc. of SIGIR 2007, pp. 103–110 (2007)
- [13] Yang, Y., Pierce, T., Carbonell, J.: A Study of Retrospective and On-Line Event Detection. In: Proc. of SIGIR 1998 (1998)
- [14] Yang, Y., Pierce, T., Carbonell, J., Jin, C.: Topic-Conditioned Novelty Detection. In: Proc. of SIGKDD 2002, pp. 688–693 (2002)
- [15] Zeng, H.J., He, Q.C., Chen, Z., Ma, W.Y., Ma, J.: Learning to Cluster Web Search Results. In: Proc. of SIGIR 2004, pp. 210–217 (2004)
- [16] Zhao, Q., Liu, T.Y., Bhowmick, S.S., Ma, W.Y.: Event Detection from Evolution of Click-through Data. In: Proc. of SIGKDD 2006, pp. 484–493 (2006)

BOAI: Fast Alternating Decision Tree Induction Based on Bottom-Up Evaluation^{*}

Bishan Yang, Tengjiao Wang^{**}, Dongqing Yang, and Lei Chang

Key Laboratory of High Confidence Software Technologies (Peking University),
Ministry of Education, China
School of Electronics Engineering and Computer Science,
Peking University, Beijing, 100871, China
{bishan_yang, tjwang, dqyang, changlei}@pku.edu.cn

Abstract. Alternating Decision Tree (ADTree) is a successful classification model based on boosting and has a wide range of applications. The existing ADTree induction algorithms apply a “top-down” strategy to evaluate the best split at each boosting iteration, which is very time-consuming and thus is unsuitable for modeling on large data sets. This paper proposes a fast ADTree induction algorithm (BOAI) based on “bottom-up” evaluation, which offers high performance on massive data without sacrificing classification accuracy. BOAI uses a pre-sorting technique and dynamically evaluates splits by a bottom-up approach based on VW-group. With these techniques, huge redundancy in sorting and computation can be eliminated in the tree induction procedure. Experimental results on both real and synthetic data sets show that BOAI outperforms the best existing ADTree induction algorithm by a significant margin. In the real case study, BOAI also provides better performance than TreeNet and Random Forests, which are considered as efficient classification models.

Keywords: classification, decision tree, ADTree, BOAI.

1 Introduction

Boosting procedure has been proved to be very helpful to improve the accuracy of decision tree classifiers. AdaBoost, introduced by Freund and Schapire [1], is the most commonly used boosting procedure. It has been successfully used to combine with decision trees like C4.5 [2], and produces very good classifiers. However, the output classifiers are often large, complex and difficult to interpret. Freund and Mason solved this problem by proposing Alternating Decision Tree (ADTree) [3] and an induction algorithm based on AdaBoost. ADTrees can produce highly accurate classifiers while generating trees in small size which are

^{*} This work is supported by the National '863' High-Tech Program of China under grant No. 2007AA01Z191, and the NSFC Grants 60473051, 60642004.

^{**} Corresponding author.

easy to interpret. They can also provide a measure of classification which helps to rate prediction confidence. Based on these attractive features, ADTrees have a wide range of applications, such as customer churn prediction, fraud detection and disease trait modeling [4,5].

Whereas ADTree is a very popular model in classification, it faces a problem of training efficiency on huge volumes of data. The original induction algorithm proposed by Freund and Mason performs split evaluation with a top-down strategy at each boosting round. The algorithm is very expensive to apply to large knowledge discovery tasks. Several techniques have been developed to tackle the efficiency problem. However, there still be a large space to improve.

For very large data sets, several techniques have been developed, mainly based on traditional decision trees. SLIQ [6] and Sprint [7] use new data structures and processing methods to scale decision trees to large data sets. PUBLIC [8] integrates the MDL “pruning” phase into the tree “building” phase. RainForest [9] uses AVC-groups which are sufficient for split evaluation to speed up tree construction. BOAT [10] provides techniques to build trees based on a subset of data and results in faster tree construction. All these algorithms are based on traditional decision trees, which compute the split criteria only based on the information of the current node, and thus can not directly apply to ADTree.

With regards to the scalability of ADTree, several optimizing methods are introduced in [11]: Z_{pure} cutoff, merging and three heuristic mechanisms. The former two methods gain little efficiency until reaching 50 boosting iterations. Although the heuristic methods reduce the induction complexity obviously, they generate trees that are different from the original trees. In [12], ADTree is upgraded to first order logic and three efficiency improvements are proposed. The caching optimization, which stores the success (failure) of each rule for each relevant instance in a bit-matrix, was shown to be most effective. Nevertheless, the additional memory consumption grows fast in the number of boosting rounds.

To address the efficiency challenges, we introduce a novel ADTree induction algorithm called BOAI¹ that gains great efficiency in handling large data sets without sacrificing classification accuracy. BOAI uses a pre-sorting technique and a bottom-up evaluation approach based on VW-group to avoid much redundancy of sorting and computation in the tree building process. To validate the efficiency of BOAI on large data sets, we conduct comprehensive experiments on both synthetic and real data sets. We also apply BOAI to a real data mining application to evaluate its performance. The results are very encouraging as BOAI offers significant performance improvements.

The remainder of this paper is organized as follows. Section 2 describes ADTree and its Induction algorithm. Section 3 introduces the new techniques used in BOAI and then describes the algorithm and implementation issues. Section 4 presents the experimental results on both real and synthetic data. Finally, section 5 concludes the paper.

¹ The acronym BOAI stands for BOttom-up evaluation for ADTree Induction.

2 Preliminaries

2.1 Alternating Decision Tree

Unlike traditional decision trees, Alternating Decision Tree (ADTree) contains two kinds of nodes: decision nodes and prediction nodes. Each decision node involves a splitting test while each prediction node involves a real-valued number (Fig. 1 shows an example). A decision node splits sets of training instances into two parts with each part belonging to a prediction node. An instance defines a set of paths along the tree from the root to some of the leaves. The classification of an instance is the sign of the sum of the prediction values along the paths defined by this instance and the sum can be interpreted as a measure of confidence. For example, the classification of the instance $(age, income) = (35, 1300)$ is $sign(0.5 - 0.5 + 0.4 + 0.3) = sign(0.7) = +1$. The prediction nodes in the instance's defined paths are shadowed in the figure.

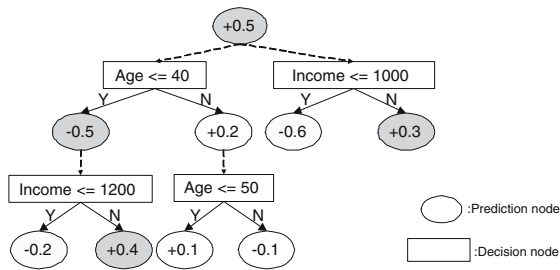


Fig. 1. An example of ADTree

2.2 ADTree Learning with AdaBoost

Freund and Mason presented the ADTree induction algorithm with the application of AdaBoost [3]. There are two sets maintained in the algorithm, a set of preconditions and a set of rules, denoted as \mathcal{P} and \mathcal{R} respectively. \mathcal{C} denotes the set of base conditions. The algorithm is given as Algorithm 1. The induction procedure can be divided into two phases at each boosting iteration: *partition* and *evaluation*. In the evaluation phase (line 2-5), the algorithm evaluates all the splits basically by a top-down strategy. It traverses the tree by a depth-first search. For each prediction node, it scans the instances at the node to compute the total weight of the instances that satisfy each possible condition. Before the computation, the instances need to be sorted on each numeric attribute to obtain the possible splits of the attribute. The best split is found by minimizing Z-value of the function that measures the weighted error of the rules (Equation 1). In the partition phase (line 6-8), a new rule is added to set \mathcal{R} and two prediction values are calculated. A decision node is created according to the rule and two prediction nodes are created associated with the prediction values. Applying the rule, the instances are split into two parts with each part propagated to one of the prediction nodes. After each boosting round, the weights of

the instances belonging to these two prediction nodes are updated, decreasing for correctly classified instances and increasing for incorrectly classified instances. As described above, the complexity of the algorithm mainly lies in the evaluation phase because of the huge sorting and computational cost. They will result in low-efficiency when training on massive data sets.

Algorithm 1. ADTree Learning with an application of AdaBoost

Input: $\mathcal{S} = \{(x_1, y_1), \dots, (x_m, y_m)\} \mid x_i \in R^d, y_i \in \{-1, +1\}$

Initialize. Set each instance’s weight $w_{i,0} = 1.0, 1 \leq i \leq m$. Set the rule set $\mathcal{R}_1 = \{True\}$. Calculate the prediction value for the root node as $a = \frac{1}{2} \ln \frac{W_+(c)}{W_-(c)}, c = True$. $W_+(c)$ (resp. $W_-(c)$) is the total weight of the positive (resp. negative) instances that satisfying condition c . Adjust the weights of the instances at the root node as $w_{i,1} = w_{i,0} e^{-ay_i}$.

- 1: **for** $t = 1$ to T **do**
- 2: **for all** c_1 such that $c_1 \in \mathcal{P}_t$ **do**
- 3: **for all** c_2 such that $c_2 \in \mathcal{C}$ **do**
- 4: Calculate

$$Z_t(c_1, c_2) = 2(\sqrt{W_+(c_1 \wedge c_2)W_-(c_1 \wedge c_2)} + \sqrt{W_+(c_1 \wedge \neg c_2)W_-(c_1 \wedge \neg c_2)}) + W_+(\neg c_1) \tag{1}$$

- 5: Select c_1, c_2 which minimize $Z(c_1, c_2)$ and set $\mathcal{R}_{t+1} = \mathcal{R}_t \cup \{r_t : \text{precondition } c_1, \text{ condition } c_2, \text{ two prediction values } a = \frac{1}{2} \ln \frac{W_+(c_1 \wedge c_2) + 1}{W_-(c_1 \wedge c_2) + 1}, b = \frac{1}{2} \ln \frac{W_+(c_1 \wedge \neg c_2) + 1}{W_-(c_1 \wedge \neg c_2) + 1}\}$
 - 6: $\mathcal{P}_{t+1} = \mathcal{P}_t \cup \{c_1 \wedge c_2, c_1 \wedge \neg c_2\}$
 - 7: Update weights: $w_{i,t+1} = w_{i,t} e^{-r_t(x_i)y_i}$, $r_t(x_i)$ is the prediction value that the rule r_t associates with the instance x_i .
-

3 BOAI - Bottom-Up Evaluation for ADTree Induction

In this section, we present BOAI, an efficient ADTree induction algorithm. Unlike the original top-down mechanism, BOAI performs split evaluation using a bottom-up approach. It gains significant efficiency of tree induction while maintaining the classification accuracy. In addition, it can easily combine with the optimizations in [11].

To bring down the large cost in the evaluation phase, BOAI uses a pre-sorting technique and applies a bottom-up evaluation based on VW-group for split evaluation. The pre-sorting technique aims at reducing the sorting cost to linear time. The bottom-up evaluation approach evaluates splits from the leaf nodes to the root node. On each prediction node, the evaluation is performed on a VW-group, which stores sufficient statistics for split evaluation. The VW-group can be built up in linear time by a bottom-up merging process. The combination of these techniques enables BOAI to induce ADTree efficiently on large data sets. Following are details about these techniques.

3.1 Pre-sorting Technique

BOAI uses a special sorting technique as a preprocessing step. It works as follows. At the beginning of the algorithm, the values of each numeric column in the input database are sorted separately. Suppose for attribute A , the sorting space of its distinct values is x_0, x_1, \dots, x_{m-1} . These values can be mapped into an integer value field $0, 1, \dots, m-1$, which reflects the offset address of each value in the sorted space. Then the original values in the data are replaced with their mapped values in the value field. As the replaced values preserve the original value distribution on the attribute, it will not affect the following evaluation on the attribute. The benefit of this method is that we can easily use the actual attribute values to index into a sorted array. The detailed analysis is given in Sect. [3.3](#).

3.2 Data Structure

Note that for a prediction node p , the possible splits of an attribute A can be evaluated separately from other attributes. Besides, the total weight of the instances that satisfy each condition on each prediction node is needed to compute for split evaluation. Let $F(p)$ denote the instances projected onto node p . Similar to the AVC-set structure in [\[9\]](#), the VW-set (The acronym VW stands for Attribute-Value, Class-Weight) of a predictor attribute A at node p is defined to preserve the weight distribution of each class for each distinct value of A in $F(p)$. Each element in a VW-set contains an attribute value field and a class-weight field (operations on the class-weight are performed on weights of two classes (positive and negative) respectively). The class-weight field can be viewed as caching $W_+(A = v)$ and $W_-(A = v)$ for each distinct attribute value v of A . Suppose in $F(p)$, v_0, \dots, v_{m-1} are the distinct values of A . If A is a categorical attribute, the split test is of form $A = v_i$, where $0 \leq i \leq m-1$. If A is a numeric attribute, the split test is of form $A \leq (v_i + v_{i+1})/2$, where $0 \leq i \leq m-2$, and v_0, \dots, v_{m-1} are in sort order. For each possible condition c on A , $W_+(c)$ and $W_-(c)$ can be easily calculated by scanning the VW-set of A at p . The VW-group of node p is defined to be the set of all VW-sets at node p , and p can be evaluated based on its VW-group, whose result is the same as that of being evaluated via scanning $F(p)$. The size of the VW-set of an attribute A at node p is determined by the number of distinct values of A in $F(p)$ and is not proportional to the size of $F(p)$.

3.3 Bottom-Up Evaluation

The great complexity in the split evaluation is due to the exhaustive exploring on all possible splits at each boosting round. Since the weights of instances change after each round, we can not simply ignore evaluating any possible split in the following round. A fundamental observation is that there are recurrences of instances at the prediction nodes. When evaluating the prediction nodes recursively from the root to the leaves, the instances in fact have a great deal

of computing and sorting overlap. To eliminate this crucial redundancy, we propose a bottom-up evaluation approach. The bottom-up approach evaluates splits from the leaf nodes to the root node based on the VW-group of each node. It uses the already computed VW-groups of the offspring nodes to construct the VW-groups of the ancestor nodes. The approach of VW-group construction is described as follows.

For a leaf prediction node p , it scans the instances at p to construct the VW-set of each attribute. For a categorical attribute A , a hash table is created to store the distinct values of A . As the attribute values in the VW-set of A are not required to be sorted, the VW-set can be constructed by collecting the distinct values of A from the hash table and computing the weight distributions on these values. For a numeric attribute A , the attribute values on A need to be sorted. With the pre-sorting technique, the sort takes linear time in most cases. Suppose there are N instances at node p and the mapped value field on A is range from 0 to $M - 1$, where M is the number of distinct values of A . It takes one pass over N instances mapping their weights into the value field of A . Then the attribute values together with their corresponding weights will be compressed into the VW-set of A . Fig. 2 shows the schematic for this construction process. The total time for getting sorted attribute values in the VW-set is $O(N + M)$. For most cases, M is smaller than N , in which case the running time is $O(N)$. If M is much larger than N , the algorithm switches to quick sort.

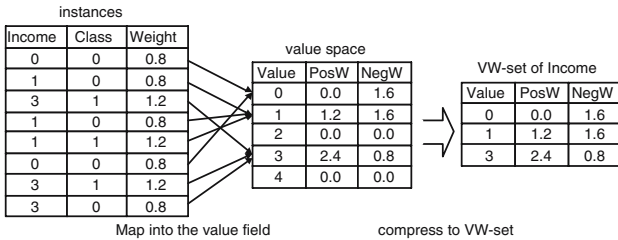


Fig. 2. Construct VW-set via scanning instances (numeric attribute): Example

For an internal prediction node p , the VW-group is constructed through a merging process, with the VW-set of each attribute generated at a time. Each generation only require time $O(m)$ where m is the total number of elements in the two merging VW-sets. Suppose Z is the VW-set of attribute A at node p and X, Y are the VW-sets of A at node p_1 and p_2 which are two prediction nodes under a decision node of p . If A is a categorical attribute, as the attribute values in X and Y are not sorted, Z can be generated by performing hash join on the attribute values in one pass over X and Y . If A is a numeric attribute, we can perform the merge procedure similar to merge sort to generate Z , and the attribute values in Z are kept in order after merging. Fig. 3 shows this process pictorially.

Since the VW-group of each prediction node can be constructed by the bottom-up approach, and each prediction node can be correctly evaluated based on its VW-group, the global minimum Z-value found by evaluating all the prediction

nodes is correct. Because the best split found at each boosting round is correct, the tree induced by the bottom-up evaluation approach is the same as that induced by the top-down evaluation approach.

The reduced cost by using the bottom-up evaluation is remarkable. In the top-down evaluation, instances are sorted on each numeric attribute on every prediction node, with each sort taking at least $O(n \log n)$ time (n is the number of the considered instances) in the average case. While in bottom-up evaluation, we focus on gaining orders of distinct attribute values whose cost is much inexpensive. Additionally, the spectacular redundancy in computing weight distributions is eliminated since the statistics are cached in VW-group to prevent being recomputed. Moreover, the bottom-up technique will not affect the accuracy of the original algorithm.

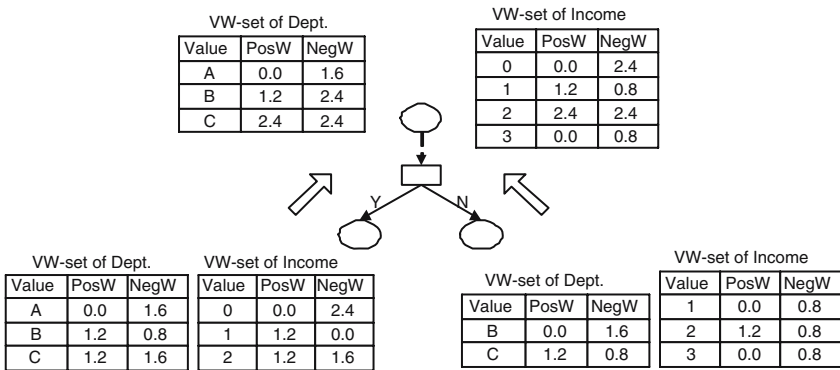


Fig. 3. Construct VW-set via merging: Example

3.4 Algorithm

In this section, we present BOAI algorithm. Note that the partition phase contributes a little to the complexity of tree induction. BOAI shares it with Algorithm 1. We just provide illustration about the evaluation phase here. Let p -VW-group denote the VW-group at prediction node p , and p -VW-set denote the VW-set contained in p -VW-group. The algorithm is given in Algorithm 2. The procedure is invoked at every boosting step, with the root node r as an input parameter. Since ADTree can have more than one decision node below a prediction node, the instances at the prediction node can be partitioned by different split criteria. We only consider partitions of one decision node for performing merging (we always choose the first decision node in BOAI). For other decision nodes, we view each of their prediction children as the root node of a subtree. The evaluating process will start from these root nodes individually. In this way, no redundant merging of VW-groups is performed. Note that when the combination is finished, the two VW-groups being merged can be deleted.

The optimizing techniques introduced in [11] can be easily integrated in BOAI and show better performance improvements. The Z_{pure} calculation can be sped

up by merging the sum of the weights of the positive (negative) instances through the merging process of the bottom-up approach. The heuristic mechanisms can be performed by only evaluating the tree portion included in the heuristic path.

Algorithm 2. EvaluateSplits(Prediction Node p)

```

1: if  $p$  is a leaf then
2:   generate p-VW-group via scanning  $F(p)$ 
3: else
4:   /* $p1$  and  $p2$  are two children of  $p$ 's first decision node*/
   p1-VW-group = EvaluateSplits( $p1$ );
5:   p2-VW-group = EvaluateSplits( $p2$ );
6:   p-VW-group  $\leftarrow$  Merge p1-VW-group and p2-VW-group
7: for each attribute  $A$  do
8:   traverse p-VW-set of attribute  $A$  /* the value field stores  $v_0, \dots, v_{m-1}$  */
9:   if  $A$  is a categorical attribute then
10:    for  $i = 0$  to  $m - 1$  do
11:      compute Z-value for test ( $A = v_i$ ) using class-weight associated with  $v_i$ 
12:   if  $A$  is a numeric attribute then
13:    for  $i = 0$  to  $m - 2$  do
14:      cumulate the sum of the class-weights associated with the former  $i$  values
      and  $v_i$  /* the values are in sorted order */
15:      compute Z-value for test ( $A \leq (v_i + v_{i+1})/2$ ) using the cumulated sum
16: for each node  $s$  such that  $s$  is the child of  $p$ 's other decision nodes do
17:   EvaluateSplits( $s$ );
18: return p-VW-group;

```

4 Experiments

In this section, we perform comprehensive experiments on both synthetic and real data sets to study the performance of BOAI. In the first experiment, we compare efficiency of BOAI and ADT on synthetic data sets up to 500,000 instances. In the next, we use the real data sets contained 290,000 records with 92 variables to evaluate the efficiency of BOAI. At last, we apply BOAI to churn prediction application, comparing to ADT, Random Forests [13] and TreeNet [14], which are considered as accurate and efficient classification models. (TreeNet won the Duke/Teradata Churn modeling competition in 2003 and won the KDD2000 data mining competition.)

BOAI and ADT are written in C++. The software of TreeNet and Random forests are downloaded from the web site (<http://www.salford-systems.com/churn.html>) of Salford Systems. All our experiments were performed on AMD 3200+ CPU running Windows XP with 768MB main memory.

4.1 Synthetic Databases

In order to study the efficiency of BOAI, we used the well-known synthetic data generation system developed by the IBM Quest data mining group [15], which

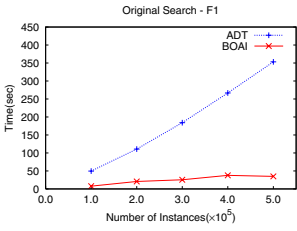


Fig. 4. Overall Time: F1

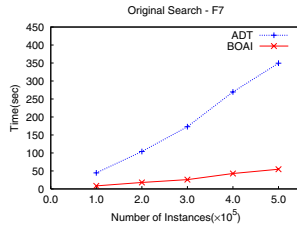


Fig. 5. Overall Time: F7

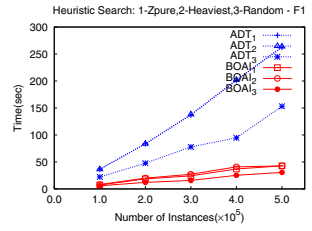


Fig. 6. Overall Time: F1

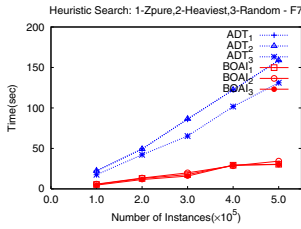


Fig. 7. Overall Time: F7

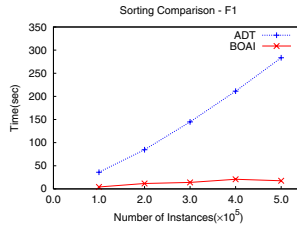


Fig. 8. Sorting Cost

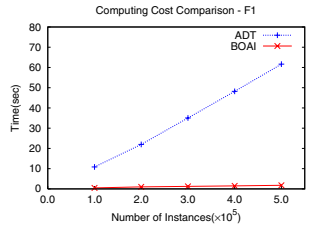


Fig. 9. Computing Cost

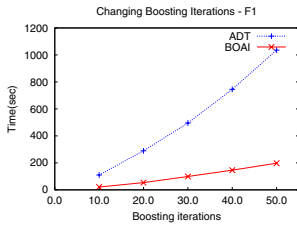


Fig. 10. Changing Iterations: F1

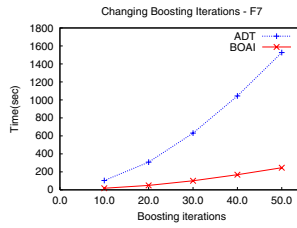


Fig. 11. Changing Iterations: F7

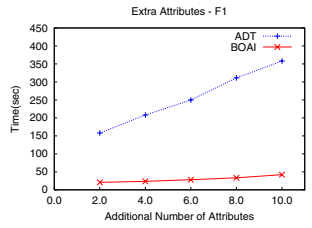


Fig. 12. Adding Attributes: F1

is often used to study the performance of decision tree construction [7,8,9,10]. Each record in this database consists of nine attributes. Among the attributes, six are numeric and the others are categorical. Ten classification functions are used to generate different data distributions. Function 1 involves two predictor attributes with respect to the class label. Function 7 is linear depending on four predictor attributes. We only show results of these two functions due to space limitation, the results are similar for other functions.

First, we examined the modeling time of BOAI and ADT as the number of the instances increases from 100,000 to 500,000. The number of boosting iterations is set to 10. We consider the Z_{pure} cut-off, merging and heuristic search techniques [11] in the comparison. In the following experiments, ADT and BOAI are default with Z_{pure} and merging options. Fig. 4 and Fig. 5 show the results of the two algorithms for function 1 and 7. BOAI is faster by a factor of six. Fig. 6 and Fig. 7 show the results of employing heuristic options (the produced models

are different from those of the original algorithm). BOAI also makes significant gains for each heuristic option. We further investigated the cost of sorting and computation in the split evaluation. Fig. 8 shows that the sorting cost in ADT rises eight times faster than BOAI in function 1. Fig. 9 shows that BOAI is about twenty-two times faster than ADT in comparison of Z-value computation cost in function 1. As the above two cost are dominant cost during tree induction, they can explain why BOAI outperforms ADT by a large margin.

We also examined the effect of boosting iterations on BOAI. We changed the number of boosting iterations from 10 to 50 while fixing the number of the instances at 200,000. Fig. 10 and Fig. 11 show the results for Function 1 and Function 7. The results are both encouraging as BOAI grows much smoother than ADT with the increasing number of boosting iterations.

Fig. 12 and Fig. 13 show the effect of adding extra attributes with random values to the instances in the input database. The number of the instances are kept at 200,000 and the number of boosting iterations is set at 10. The additional attributes need to be evaluated but they will never be chosen as the split attribute. Thus the extra attributes increase tree induction time while the final classifier keeps the same. BOAI exhibits much more steady performance with the increasing number of attributes.

4.2 Real Data sets

In order to study the performance of BOAI in real cases, we experimented with a real data set obtained from China Mobile Communication Company. The data refers to seven months of customer usage, from January 2005 through July 2005. The data set consists of 290,000 subscribers covering 92 variables, including customer demographic information, billing data, call detail data, service usage data and company interaction data. The churn indicator attribute is the class attribute. We first study the training time of BOAI on the real data sets. Then we apply BOAI to churn prediction, comparing its performance to ADT, TreeNet and Random Forests. To guarantee the prediction accuracy, we don't consider heuristic techniques in the comparison.

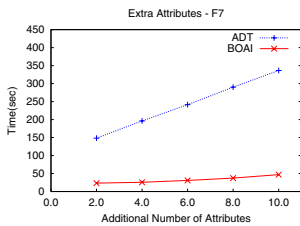


Fig. 13. Adding Attributes: F7

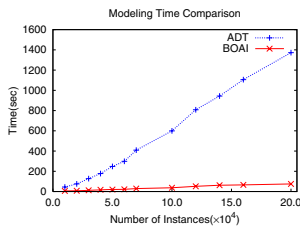


Fig. 14. Overall Time

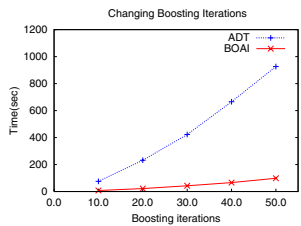


Fig. 15. Changing Iterations

We first compare the training time of BOAI and ADT. Fig. 14 shows the overall running time of the algorithms as the number of the input instances

increases from 20,083 to 219,644 and the number of boosting iterations sets at 10. BOAI is about fourteen times faster than ADT. Then we change the number of boosting iterations from 10 to 50 with 20,083 instances. Fig. 15 shows that BOAI offers more steady performance with the changing number of iterations. For memory usage, the largest size of the VW-group used in induction is only 10MB for 219,644 instances (with 92 attributes), which is also a small size to easily hold in memory.

In the next, we apply BOAI to churn prediction to study its performance. we sampled 20,083 examples from the original data set as a calibration set which has 2.1% churn rate, and 5,062 examples as a validation set which has 1.8% churn rate. Since the data is highly skewed, we take a re-balancing strategy to tackle the imbalanced problem. As a pre-processing step, we multiply the weight of each instance in the minority class by W_{maj}/W_{min} , where W_{maj} (resp. W_{min}) is the total weight of the majority (resp. minority) class instances. In this way, the total weights of the majority and minority instances are balanced. Unlike sampling [16], re-balancing weights has little information loss and does not introduce more computing power on average.

Table 1. Performance comparison on churn analysis

Models	F -measure	G -mean	W-accuracy	Modeling Time (sec)
ADT (w/o re-balancing)	56.04	65.65	44.53	75.56
Random Forests	19.21	84.04	84.71	960.00
TreeNet	72.81	79.61	64.40	30.00
BOAI	50.62	90.81	85.84	7.625

We compare the predicted accuracy of BOAI, ADT (without re-balancing), TreeNet and Random Forests, with measures of F -Measure, G -Mean and Weighted Accuracy [17], which are commonly used to evaluate performance on skewed class problem. The modeling time of these algorithms is also given. The results, shown in Table 1, indicate that BOAI outperforms ADT, TreeNet and RF when evaluated in terms of G -mean and Weighted-Accuracy. More importantly, BOAI uses the least modeling time.

5 Conclusion

In this paper, we have developed a novel approach for ADTree induction, called BOAI, to speed up ADTree construction on large training data sets. The key insight is to eliminate the great redundancy of sorting and computation in the tree induction by using a bottom-up evaluation approach based on VW-group. In experiments on both synthetic and real databases, BOAI offers significant performance improvements over the best existing algorithm while constructing exactly the same ADTree. We also study the performance of BOAI for churn prediction. With the re-balancing technique, BOAI offers good prediction accuracy while spends much less modeling time compared with ADT, TreeNet and

Random Forests, which are reported as efficient classification models. Therefore, BOAI is an attractive algorithm for modeling on large data sets. It has been successfully used for real-life churn prediction in telecommunication.

Acknowledgments. We gratefully thank Prof. Jian Pei in Simon Fraser University for his insightful suggestions.

References

1. Freund, Y., Schapire, R.E.: A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences* 55(1), 119–139 (1997)
2. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1993)
3. Freund, Y., Mason, L.: The alternating decision tree learning algorithm. In: 16th International Conference on Machine Learning, pp. 124–133 (1999)
4. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
5. Liu, K.Y., Lin, J., Zhou, X., Wong, S.: Boosting Alternating Decision Trees Modeling of Disease Trait Information. *BMC Genetics* 6(1) (2005)
6. Mehta, M., Agrawal, R., Rissanen, J.: SLIQ: A fast scalable classifier for data mining. In: 5th International Conference on Extending Database Technology, pp. 18–32 (1996)
7. Shafer, J., Agrawal, R., Mehta, M.: SPRINT: A scalable parallel classifier for data mining. In: 22nd International Conference on Very Large Databases, pp. 544–555 (1996)
8. Rastogi, R., Shim, K.: PUBLIC: A Decision Tree Classifier that Integrates Pruning and Building. In: 24th International Conference on Very Large Database, pp. 315–344 (1998)
9. Gehrke, J., Ramakrishnan, R., Ganti, V.: Rainforest: A framework for fast decision tree construction of large data sets. In: 24th International Conference on Very Large Database, pp. 127–162 (1998)
10. Gehrke, J., Ganti, V., Ramakrishnan, R., Loh, W.Y.: BOAT—optimistic decision tree construction. In: ACM SIGMOD International Conference on Management of Data, pp. 169–180 (1999)
11. Pfahringer, B., Holmes, G., Kirkby, R.: Optimizing the Induction of Alternating Decision Trees. In: 5th Pacific-Asia Conference on Knowledge Discovery and Data Mining, pp. 477–487 (2001)
12. Vanassche, A., Krzywania, D., Vaneyghen, J., Struyf, J., Blockeel, H.: First order alternating decision trees. In: 13th International Conference on Inductive Logic Programming, pp. 116–125 (2003)
13. Breiman, L.: Random forests. *Machine Learning Journal* 45, 5–32 (2001)
14. <http://www.salford-systems.com/products-treenet.html>
15. IBM Intelligent information systems, <http://www.almaden.ibm.com/software/quest/resources/>
16. Maloof, M.: Learning when data sets are imbalanced and when costs are unequal and unknown. In: ICML Workshop on Learning from Imbalanced Data Sets (2003)
17. Chen, C., Liaw, A., Breiman, L.: Using Random Forest to Learn Imbalanced Data. Technical Report 666, Statistics Department, University of California at Berkeley (2004)

Feature Selection by Nonparametric Bayes Error Minimization*

Shuang-Hong Yang^{1,2} and Bao-Gang Hu^{1,2}

¹ National Lab of Pattern Recognition(NLPR) & Sino-French IT Lab(LIAMA)
Institute of Automation, Chinese Academy of Sciences

² Graduate School, Chinese Academy of Sciences
P.O. Box 2728, Beijing, 100080 China
{shyang,hubg}@nlpr.ia.ac.cn

Abstract. This paper presents an algorithmic framework for feature selection, which selects a subset of features by minimizing the nonparametric Bayes error. A set of existing algorithms as well as new ones can be derived naturally from this framework. For example, we show that the Relief algorithm greedily attempts to minimize the Bayes error estimated by k -Nearest-Neighbor method. This new interpretation not only reveals the secret behind Relief but also offers various opportunities to improve it or to establish new alternatives. In particular, we develop a new feature weighting algorithm, named Parzen-Relief, which minimizes the Bayes error estimated by Parzen method. Additionally, to enhance its ability to handle imbalanced and multiclass data, we integrate the class distribution with the max-margin objective function, leading to a new algorithm, named MAP-Relief. Comparison on benchmark data sets confirms the effectiveness of the proposed algorithms.

1 Introduction

Feature selection is a process of selecting a small number of highly predictive features out of a large set of candidate attributes that might be strongly irrelevant or redundant. It plays a fundamental role in pattern recognition, data mining, and more generally machine learning tasks [6], e.g., facilitating data interpretation, reducing measurement and storage requirements, increasing predeceasing speeds, improving generalization performance, etc.

Most feature selection methods approach the task as a search problem, where each state in the search space is a possible feature subset. Suppose we are given a set of input vectors $\{\mathbf{x}_n\}_{n=1}^N$ along with corresponding targets $\{y_n\}_{n=1}^N$ drawn *i.i.d* from an unknown distribution $P(\mathbf{x},y)$, where $\mathbf{x}_n \in \mathbf{X} \subset \mathcal{R}^D$ is a training instance and $y_n \in \mathbf{Y}=\{0,1,\dots,C-1\}$ is its label, N, D, C denote the training set size, the input space dimensionality and the total number of categories respectively. The d -th feature of \mathbf{x} is denoted as $x^{(d)}$, $d=1,2,\dots,D$. The goal of feature selection is to select a subset of M ($M \ll D$) most predictive features, i.e., to

* This work is supported in part by NSFC (#60275025, #60121302).

find a preprocessing of data $\tau(\mathbf{x}) : \mathbf{x} \rightarrow (\mathbf{x} * \boldsymbol{\tau})$, where $\boldsymbol{\tau} = [\tau_1, \dots, \tau_D] \in \mathcal{S} = \{0, 1\}^D$, $\|\boldsymbol{\tau}\|_0 = M$, $(\mathbf{x} * \boldsymbol{\tau}) = [x^{(1)}\tau_1, \dots, x^{(D)}\tau_D]$ denotes the element-wise product. Let the feature selection criterion function be represented by $J(\cdot)$. Formally, the problem of feature selection can be formalized as:

$$\boldsymbol{\tau} = \arg \max_{\boldsymbol{\tau} \in \mathcal{S}, \|\boldsymbol{\tau}\|_0 = M} J(\boldsymbol{\tau}) \quad (1)$$

There are two basic elements in a typical feature selection method [4][8]: (.) the search strategy, a procedure to generate candidate $\boldsymbol{\tau}$; and (.) the evaluation criterion $J(\cdot)$, a measure to assess the goodness of $\boldsymbol{\tau}$. Existing search approaches are generally divided [8] into exhaustive (exhaustive, best first, branch and bound, beam, etc.), sequential (forward/backward sequential, greedy, etc.), or heuristic (simulated annealing, genetic algorithm, etc.). Feature weighting is a greedy algorithm. It assigns to each feature a real valued number to indicate its usefulness, making possible to efficiently select a subset of features simply by searching in a continuous space. For this reason, this paper will fix the search strategy at feature weighting and focus mainly on the evaluation criterion aspect. However, extensions to other search schemes are straightforward.

Most of the existing evaluation criteria are based on heuristic intuitions or domain knowledge, and therefore still lacks rigorous theoretical treatment. For example, the margin [9] algorithm is recently interpreted as a method to maximize the average margin [15][5]. However, the definition of margin is based on heuristics. The secret behind the margin is unclear.

In this paper, we present an algorithmic (evaluation criterion) framework for feature selection, which selects a subset of features by minimizing the nonparametric Bayes error. Many existing approaches as well as new ones can be naturally derived from this framework. In particular, we find that the Relief algorithm attempts to greedily minimize the nonparametric Bayes error that is estimated by k -nearest-neighbor (k NN) method. This new interpretation of Relief not only reveals the secret behind the margin concept, but also enables us to identify its weaknesses so as to establish new algorithms to mitigate the drawbacks. In this paper, an alternative algorithm, called Parzen-Relief, is proposed, which resembles the standard Relief algorithm but using the Parzen method to estimate the Bayes error. We will show that the empirical performance of Parzen-Relief usually outperforms Relief. In addition, we find that Relief makes an implicit assumption that the class distribution $P(c) = 1/2$. This undesirable assumption heavily limits its performance in handling imbalanced or multiclass data set. To address this drawbacks, we propose a MAP-Relief algorithm, which incorporate the class distribution into the margin maximization objective function. Both Parzen-Relief and MAP-Relief are of the same computational complexity as the standard Relief algorithm. However, both of them show significant performance improvement compared with Relief.

The organization of this paper is as follows. Section 2 presents the algorithmic framework. Section 3 offers a new interpretation of Relief and presents a new alternative, i.e., the Parzen-Relief algorithm. Section 4 establishes the MAP-Relief

algorithm to handle imbalanced and/or multiclass data. Section 5 presents the comparison results and Section 6 summarizes the whole paper.

2 An Algorithmic Framework for Feature Selection

2.1 Nonparametric Bayes Error Minimization

Theoretically, two different, but closely-related, optimal evaluation criteria can be identified. The first one [10] is based on information theory, which attempts to model the dependence between patterns and their labels in a reduced-dimensional space while retaining as much information as possible, i.e.,:

$$\begin{aligned} \min_{\tau} KL\{P(y|\mathbf{x})||P(y|\tau(\mathbf{x}))\} \\ s.t. : ||\tau||_0 = M, \end{aligned} \tag{2}$$

where $KL\{p(\mathbf{x})||q(\mathbf{x})\} = E_{\mathbf{x}}[p(\mathbf{x}) \log \frac{p(\mathbf{x})}{q(\mathbf{x})}]$ denotes the KL-divergence between two distribution $p(\mathbf{x})$ and $q(\mathbf{x})$. We refer this theoretical criterion as **ROC** to emphasize its aim to rule out irrelevant features. Various practical criteria are related to ROC. Examples include the entropy or mutual-information (MI) criterion and its variations [6], and so on.

We propose here another criterion. In contrast to ROC, this criterion is more straightforward and pragmatic. It considers classification directly and naturally reflects the Bayes error rate in the reduced space, i.e.:

$$\begin{aligned} \min_{\tau} \{ \inf_{\delta} E_{\mathbf{x}}[err(\delta|\tau(\mathbf{x}))] = E_{\mathbf{x}}[1 - \max_c P(c|\tau(\mathbf{x}))] \} \\ s.t. : ||\tau||_0 = M. \end{aligned} \tag{3}$$

where δ denotes a decision rule, $E_{\mathbf{x}}\{err(\delta|\tau(\mathbf{x}))\}$ is the generalization error of δ in the reduced space, $c \in \{0, 1, \dots, C - 1\}$. We called this optimal criterion **DOC** to highlight its goal to maximize the discriminating ability of features.

The ROC criterion has been proved powerful for its keeping as much information for modelling the posterior distribution, which is useful for many domains [1]. It is also closely related with the Bayes error for classification [7]. However, for many applications where \mathbf{x} have high dimensionality, modelling the posterior probability with limited samples is not only risky, but also wasteful of resources. In contrast, there are several compelling reasons for using DOC to assess the quality of features if we only wish to make classification decisions. One justification is from Vapnik [16] that "one should solve the problem (classification) directly and never solve a more general problem (modelling $P(y|\mathbf{x})$) as an intermediate step."

In practical cases, we cannot compute the Bayes error exactly because the precise distribution for generating the data is not available. Therefore, approximate methods for estimating the Bayes error is necessary. There are in general two distinct approaches.

- **Wrapper Methods** optimizes the generalization error of δ with respect to both τ and δ :

$$\tau, \delta = \arg \min_{\delta \in \mathcal{H}, \tau} \{E_{\mathbf{x}}[err(\delta|\tau(\mathbf{x}))]\} \tag{4}$$

where \mathcal{H} denotes a hypothesis space. Clearly, this is equivalent to estimate the Bayes risk by $\inf_{\delta \in \mathcal{H}} \{E_{\mathbf{x}}[err(\delta|\tau(\mathbf{x}))]\}$.

- **Filter Methods**¹ estimates the Bayes risk directly without using a specific form of classifier. For example, the estimation of Bayes risk can be obtained by estimating the probability distribution involved in Eq.(3).

Given a set of training data, the expectation over \mathbf{x} can be approximated by the empirical average, i.e.:

$$E_{\mathbf{x}}[1 - \max_c P(c|\tau(\mathbf{x}))] \approx \frac{1}{N} \sum_{n=1}^N \left(1 - \max_c P(c|\tau(\mathbf{x}_n))\right). \tag{5}$$

To estimate $P(c|\tau(\mathbf{x}_n))$, there are two types of methods, namely, the parametric and nonparametric estimation methods. When the parametric methods [1] are concerned, one typically assumes a generative model for (\mathbf{x}, y) and estimates $P(\mathbf{x}, y)$ through, e.g., maximum-likelihood or Bayesian estimation methods.

We use non-parametric estimators to estimate $P(c|\tau(\mathbf{x}_n))$ and approximately minimize the Bayes error by solving the following problem:

$$\tau = \arg \max_{\|\tau\|_0=M} \frac{1}{N} \sum_{n=1}^N \left(P(y_n|\tau(\mathbf{x}_n)) - \sum_{c \neq y_n} P(c|\tau(\mathbf{x}_n)) \right). \tag{6}$$

An obvious advantage to use nonparametric estimators is that the results will be robust to any probability distribution since the estimators do not rely on specific distribution assumptions. However, to directly estimate $P(c|\tau(\mathbf{x}_n))$ is practically difficult because \mathbf{x}_n is usually in a high dimensional continuous space. Since $P(c|\mathbf{x}_n) \propto P(c)P(\mathbf{x}_n|c)$, we can estimate $P(c)$ by the class ratio and use nonparametric method to estimate $P(\mathbf{x}_n|c)$.

2.2 Related Works

Saon et al [14] also proposed to reduce the input dimensionality by minimizing the Bayes error. However, their methods were established for the purpose of linear feature transformation, not for feature subset selection. In addition, they used indirect approaches to approximately minimize Bayes error, i.e., by maximizing the *margin* or minimizing the *Bayes error*. To make their approach tractable, Gaussian assumption has to be made about the class-conditional densities, which strongly limits the performance of their methods, because multimodality, or non-Gaussian distribution is frequently

¹ Note that some filter methods are based on ROC framework, e.g., MI-based feature ranking.

observed in practical applications. These drawbacks are also shared by similar methods such as the Bhattacharyya distance approach [3,19]. In the context of vision recognition, Vasconcelos [17] proposed a feature selection approach based on the infomax principal. Although their criterion were shown closely related to Bayes error, it is suboptimal in the minimum Bayes error sense, too. Carneiro et al [2] proposed a joint feature extraction and selection algorithm by minimizing the Bayes error. However, they used Gaussian mixture models to estimate the class-conditional distribution. A technical difficulty is that the number of mixture components, which has a dominant importance in their method, is very difficult to be determined in practice. Weston et al [18] proposed a learning algorithm that achieves variable selection by minimizing the zero-norm regularization to enforce sparseness of a kernel machine. To make the computation tractable, convex loss functions, e.g., the hinge loss function used in support vector machines (SVM), have to be employed as the optimization objective. Although the Bayes error can be naturally reflected by the zero-one loss function, these convex surrogate loss functions offer poor approximation to the zero-one loss. Therefore, their approach does not directly minimize Bayes error, either.

3 Relief and Nonparametric Bayes Error Minimization

Among the existing feature weighting methods, Relief [9,15] is considered one of the most successful one due to its effectiveness, simplicity and particularly the ability to tackle dependent features [13]. Recently, Gilad et al [5] established a new variation of Relief based on the concept of margin. This idea was further explored by Sun [15] to provide a new interpretation of Relief as a max-margin convex optimization problem. This new perspective simplifies the computation significantly. However, the secret behind the success of Relief is still unclear. We will show that Relief approximately minimizes the nonparametric Bayes error estimated by k NN method via a greedy feature weighting search scheme.

According to [15], Relief is equivalent to a convex optimization problem:

$$\begin{aligned} \max \quad & \sum_{n=1}^N \mathbf{w}^T \mathbf{m}_n \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{7}$$

where $\mathbf{w}=(w_1, w_2, \dots, w_D)^T$, $\mathbf{m}_n = |\mathbf{x}_n - M(\mathbf{x}_n)| - |\mathbf{x}_n - H(\mathbf{x}_n)|$ is called the margin for the pattern \mathbf{x}_n , $H(\mathbf{x}_n)$ and $M(\mathbf{x}_n)$ denote the nearest-hit (the nearest neighbor from the same class) and nearest-miss(the nearest neighbor form different class) of \mathbf{x}_n respectively. By using the Lagrangian technique, a simple close-form solution to Eq.(7) can be derived, i.e.:

$$\mathbf{w} = (\overline{\mathbf{m}})^+ / \|(\overline{\mathbf{m}})^+\| \tag{8}$$

where $\overline{\mathbf{m}} = \frac{1}{N} \sum_{n=1}^n \mathbf{m}_n$ is the average margin, $(\cdot)^+$ denotes the positive part.

We now show how Relief minimizes the nonparametric Bayes error via greedy feature weighting. Since Relief is originally established for binary classification tasks, we first consider binary labels, i.e., $y_n \in \{0,1\}$, and will extend the results

to multi-class scenarios in Section 4. Suppose the class distribution $(c=1)=0.5$, $(c=0)=0.5$, estimating the class-conditional probability by 1-NN estimator, we have:

$$\begin{aligned} & P(c = y_n | \tau(\mathbf{x}_n)) - P(c \neq y_n | \tau(\mathbf{x}_n)) \\ & \propto P(\tau(\mathbf{x}_n) | c = y_n) - P(\tau(\mathbf{x}_n) | c \neq y_n) \\ & = \frac{1/N}{V^{(1)}} - \frac{1/N}{V^{(2)}} \\ & \propto \frac{1}{\|\tau(\mathbf{x}_n) - H(\tau(\mathbf{x}_n))\|_2^M} - \frac{1}{\|\tau(\mathbf{x}_n) - M(\tau(\mathbf{x}_n))\|_2^M} \end{aligned} \tag{9}$$

where $V^{(1)}$ and $V^{(2)}$ denote the volumes of the hyper-spheres from \mathbf{x}_n to $H(\tau(\mathbf{x}_n))$ and to $M(\tau(\mathbf{x}_n))$ respectively. However, to obtain the optimal $\tau(\cdot)$, minimizing Eq.(9) is an NP-hard combinatorial optimization problem. Therefore, heuristic search is necessary. Considering a greedy search scheme, which searches each dimension independently with a feature weighting scheme:

$$\begin{aligned} \mathbf{w} &= \arg \max_{\mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|=1} \sum_{d=1}^D w_d \sum_{n=1}^N P(c = y_n | x_n^{(d)}) - P(c \neq y_n | x_n^{(d)}) \\ &= \arg \max_{\mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|=1} \sum_{d=1}^D w_d \sum_{n=1}^N \frac{1}{|x_n^{(d)} - H(x_n^{(d)})|} - \frac{1}{|x_n^{(d)} - M(x_n^{(d)})|} \\ &\approx \arg \max_{\mathbf{w} \geq \mathbf{0}, \|\mathbf{w}\|=1} \sum_{d=1}^D w_d \sum_{n=1}^N |x_n^{(d)} - M_n^{(d)}| - |x_n^{(d)} - H_n^{(d)}| \end{aligned} \tag{10}$$

That is:

$$\begin{aligned} & \max \sum_{d=1}^D w_d \sum_{n=1}^N |x_n^{(d)} - M_n^{(d)}| - |x_n^{(d)} - H_n^{(d)}| \\ & s.t. : \|\mathbf{w}\| = 1, \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{11}$$

where $M_n^{(d)}$ and $H_n^{(d)}$ denote the nearest-miss and nearest-hit of \mathbf{x}_n in the d -th dimensional subspace, the last line of Eq.(10) follows from the consideration to avoid numerical overflows. Clearly, Eq.(11) will be identical to Eq.(7) when we approximate $M_n^{(d)}$ and $H_n^{(d)}$ with $M^{(d)}(\mathbf{x}_n)$ and $H^{(d)}(\mathbf{x}_n)$, which means, instead of using d -dimensional nearest-miss's $M_n^{(d)}$, we use a single d -dimensional nearest-miss $M(\mathbf{x}_n)$ and approximate the d -th 1-dimensional nearest-miss $M_n^{(d)}$ with the d -th element $M^{(d)}(\mathbf{x}_n)$ of $M(\mathbf{x}_n)$. We now establish two alternatives by exploring the new interpretation.

Parametric estimation methods are usually considered preferable when a proper generative model (based on prior knowledge) is available. Particularly, if we consider a Naïve Bayes generative model: $p(\mathbf{x}|y) = \prod_{d=1}^D p(x^{(d)}|y)$, where $p(x^{(d)}|y = c) = \mathcal{N}(x^{(d)}|\mu_{c,d}, \sigma_{d,c})$, we will get the following algorithm:

$$\begin{aligned} & \max \sum_{d=1}^D w_d \sum_{n=1}^N \left(\frac{(x_n^{(d)} - \mu_{d,1-y})^2}{\sigma_{d,1-y}^2} - \frac{(x_n^{(d)} - \mu_{d,y})^2}{\sigma_{d,y}^2} \right) \\ & s.t. : \|\mathbf{w}\| = 1, \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{12}$$

We refer this algorithm as ‘**Fisher-Score-based Feature Weighting**’ (**FFW**), since it has close similarity with the feature ranking criterion \dots : $FS_d = \left| \frac{\mu_{d,1} - \mu_{d,0}}{\sigma_{d,1}^2 + \sigma_{d,0}^2} \right|$. In fact, if assuming $\sigma_{d,1}^2 = \sigma_{d,0}^2$, the order of feature weights obtained from Eq. (12) is identical to the ranking results of Fisher score.

Parzen widow estimator is also applicable. Here we consider a specific Parzen widow function, i.e., the truncated potential function:

$$g(\mathbf{x}, \mathbf{x}_n) = \begin{cases} 0, & \text{if } \|\mathbf{x} - \mathbf{x}_n\| > \varsigma \\ \frac{1}{\|\mathbf{x} - \mathbf{x}_n\|^2}, & \text{else} \end{cases} \tag{13}$$

We have:

$$P(\tau(\mathbf{x}_n)|c = y_n) - P(\tau(\mathbf{x}_n)|c \neq y_n) \propto \sum_{\mathbf{x}_i \in \Omega_n^{(o)}} \frac{1/\varsigma_1^M}{\|\mathbf{x}_i - \mathbf{x}_n\|^2} - \sum_{\mathbf{x}_j \in \Omega_n^{(e)}} \frac{1/\varsigma_1^M}{\|\mathbf{x}_j - \mathbf{x}_n\|^2} \tag{14}$$

where $\Omega_n^{(o)} = \{x : \|\mathbf{x} - \mathbf{x}_n\| \leq \varsigma_1, y = y_n\}$ and $\Omega_n^{(e)} = \{x : \|\mathbf{x} - \mathbf{x}_n\| \leq \varsigma_2, y \neq y_n\}$ denote the Homogenous and Heterogeneous Neighbor Set of \mathbf{x}_n .

Following a series of similar simplification, we have a new feature weighting method, which we called ‘**Parzen-Relief**’ (**P-Relief**):

$$\begin{aligned} \max \quad & \sum_{n=1}^N \mathbf{w}^T \mathbf{m}_n^p \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{15}$$

where the margin $\mathbf{m}_n^p = \sum_{\mathbf{x}_i \in \Omega_n^{(o)}} \frac{|x_n^{(d)} - x_i^{(d)}|}{\varsigma_1(\mathbf{x}_n)} - \sum_{\mathbf{x}_j \in \Omega_n^{(e)}} \frac{|x_n^{(d)} - x_j^{(d)}|}{\varsigma_2(\mathbf{x}_n)}$.

4 Handling Imbalanced Multi-class Data Set

An undesirable assumption made by the Relief algorithm is that $P(c) = 1/2$ for all the classes $c = 0, 1, \dots, C$. However, this is less likely the case in practice. To address this problem, we modify the definition of the margin to include the prior probability of each class, i.e.:

$$\begin{aligned} \max \quad & \sum_{n=1}^N \mathbf{w}^T \mathbf{m}_n^\pi \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, \mathbf{w} \geq \mathbf{0} \end{aligned} \tag{16}$$

where $\mathbf{m}_n^\pi = P(y_n)|\mathbf{x}_n - M(\mathbf{x}_n)| - (1 - P(y_n))|\mathbf{x}_n - H(\mathbf{x}_n)|$, $\boldsymbol{\pi} = [P(0), \dots, P(C-1)]$ is the class distribution. We term this algorithm as **MAP-Relief** (**M-Relief**). From our new interpretation, the rationale of this formulation is clear. In Section 5, we will show that while the performance of other algorithms in the Relief family degrades significantly when the data set is strongly imbalanced, M-Relief is less sensitive to such sampling bias.

Another advantage of M-Relief algorithm is its ability of handling multi-class data. The original Relief algorithm only works for binary classification problems.

[11] extends it to multi-class tasks by a heuristic updating rule, which is equivalent to solve Relief with the margin vector:

Table 1. Characteristics of Twelve UCI Data Sets

Data Set	Train Size	Test Size	#Feature	#Class
Breast	400	283	9	2
German	700	300	20	2
Ionosphere	235	116	34	2
Waveform	400	4600	21	2
Pima	400	368	8	2
Heart	170	100	13	2
Sonar	165	43	60	2
Splice	1000	2175	60	2
LRS	380	151	93	48
Glass	120	94	9	6
Ecoli	200	136	7	8
Segmentation	210	2100	18	7

$$\mathbf{m}_n^F = \sum_{c \neq y_n} P(c) |\mathbf{x}_n - M_c(\mathbf{x}_n)| - |\mathbf{x}_n - H(\mathbf{x}_n)|,$$

where $M_c(\mathbf{x}_n)$ is the nearest miss of \mathbf{x}_n in class c , $c \in \{0, \dots, C-1\}$. Therefore, one needs to search for k -nearest-hit and $k \times (C-1)$ -nearest-miss for each sample to solve ReliefF. However, from our new interpretation of Relief, this is clearly unnecessary, because in general the following relationship always holds:

$$\sum_{c \neq y_n} P(c) P(\mathbf{x}_n | c) = P(\mathbf{x}_n, c \neq y_n) = (1 - P(y_n)) P(\mathbf{x}_n | c \neq y_n)$$

The Iterative-Relief (I-Relief, [15]) algorithm deals with multi-class data using a margin vector defined somewhat similar with our definition \mathbf{m}_n^π , but with an implicit assumption $P(y_n)=0.5$. Obviously, this assumption is inappropriate for problems involving ($C > 3$) categories. This could become more severe when C goes larger such that the 'one-versus-rest' splits (i.e.: $\{\mathbf{x}_i: y_i = y_n\}$ and $\{\mathbf{x}_j: y_j \neq y_n\}$) of the data set become more and more imbalanced.

It is interesting to see that M-Relief possesses advantages of both ReliefF and I-Relief, and at the same time mitigates their drawbacks: (i) Similar with ReliefF, M-Relief incorporates the class distribution to tackling imbalanceness; (ii) Similar with I-Relief, M-Relief needs only one, instead of $k \times (C-1)$, nearest-miss for each pattern. Both advantages, i.e., computational efficiency and ability to handling imbalanceness, would become especially preferable when problems with very large C are faced.

5 Experiments

We conduct extensive experiments to evaluate the effectiveness of the proposed methods. Twelve benchmark machine learning data sets from UCI collection are selected because of their diversity in the numbers of features, instances and classes, as summarized in Table.1. To facilitate the comparison, fifty irrelevant

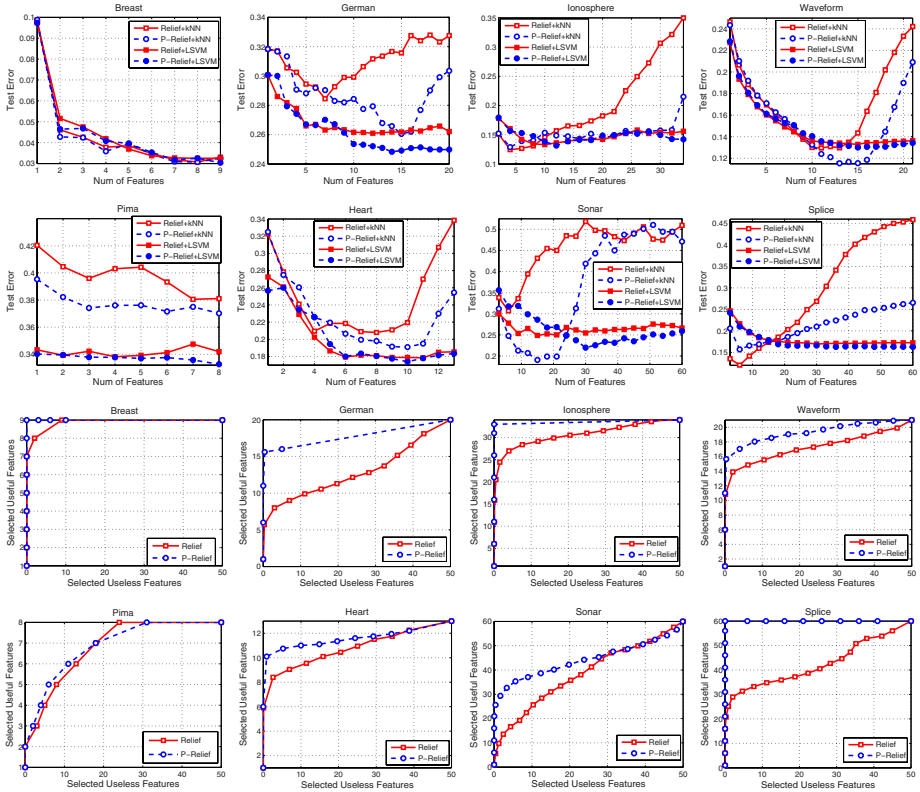


Fig. 1. Comparison of P-Relief and Relief

features (known as ‘probes’) are added to each pattern, each of which is an independently Gaussian distributed random variable, i.e., $\mathcal{N}(0,20)$. Two distinct metrics are used to evaluate the effectiveness of the feature selection algorithms. One is the classification accuracy, which is estimated by the k NN classifier (in some cases also by the Lagrangian Support Vector Machine (LSVM, [12]), an efficient implementation of SVMs). The other metric is the Receiver Operating Characteristic (ROC) curve [15], which can indicate the effectiveness of different feature selection algorithms in identifying relevant features and at the same time ruling out useless ones. To eliminate statistical deviations, all the experiments are averaged over 20 random runs. The hyper-parameters, i.e., the number of nearest neighbors k in k NN and the regularization parameter C in LSVM are determined by five-fold cross validation on the training data set.

We first apply Relief and P-Relief to the eight binary classification data sets. For this comparison, both k NN and LSVM are tested. The hyper-parameter in P-Relief is determined based on a simple rule: $\zeta_1(\mathbf{x}_n) = 1.2 \times \|\mathbf{x}_n - H_n\|$, $\zeta_2(\mathbf{x}_n) = 1.2 \times \|\mathbf{x}_n - M_n\|$, so as to keep the running time of both methods comparable. The top two lines of Fig. 1 shows the average testing error of each selector-classifier

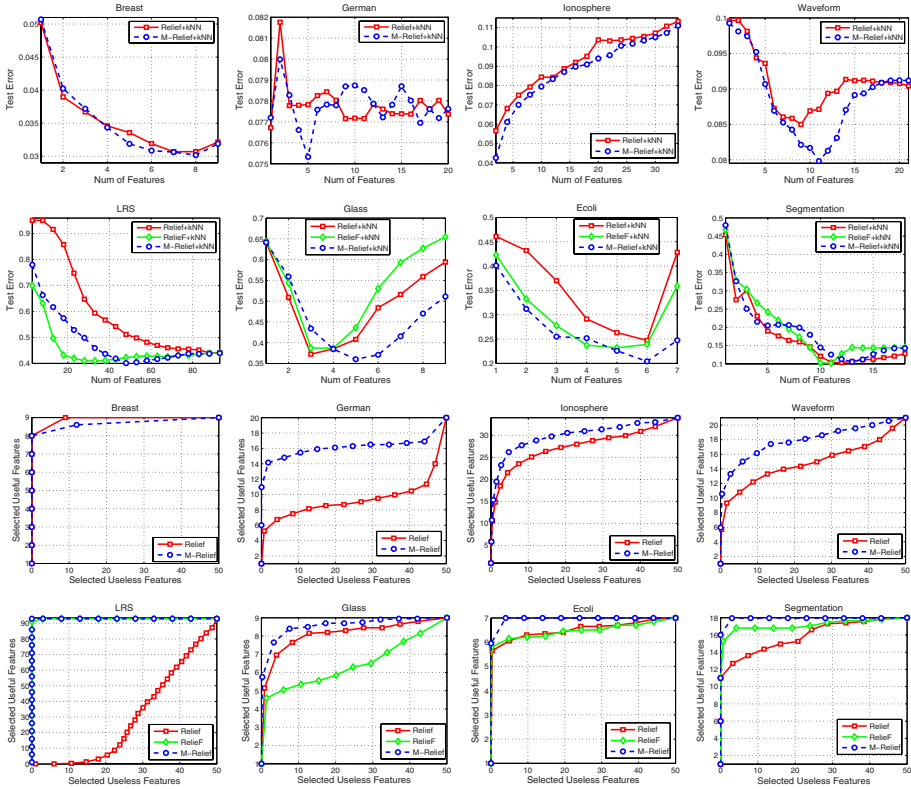


Fig. 2. Comparison of Relief, ReliefF and M-Relief

combination, as a function of the number of top-ranked features. The ROC curves are plotted in the bottom two lines of Fig. 1. As a reference, the best average classification error and standard deviation of each algorithm are also plotted as a bar chart in Fig. 3. We can see that, although P-Relief shares the same computational complexity as Relief, it usually achieves better performance than Relief. In particular, P-Relief outperforms Relief significantly in five (out of eight) data sets and performs comparably on the other three when the testing error metric is concerned. In the meanwhile, for all the eight data sets, P-Relief has a larger area under ROC curve than Relief does.

We now compare Relief, ReliefF and the proposed M-Relief in handling imbalanced/multiclass data. For this purpose, four binary data sets, which are relatively more imbalanced, and four multiclass data sets are used. To facilitate the comparison, a further bias-sampling procedure is applied to the four binary classification data sets to make them more imbalanced, i.e., 80% of the patterns randomly sampled from the minority class are discarded. Since Relief is originally established for binary classification, to enable it to apply to multiclass tasks, we use the margin definition used in I-Relief. To make a fair comparison,

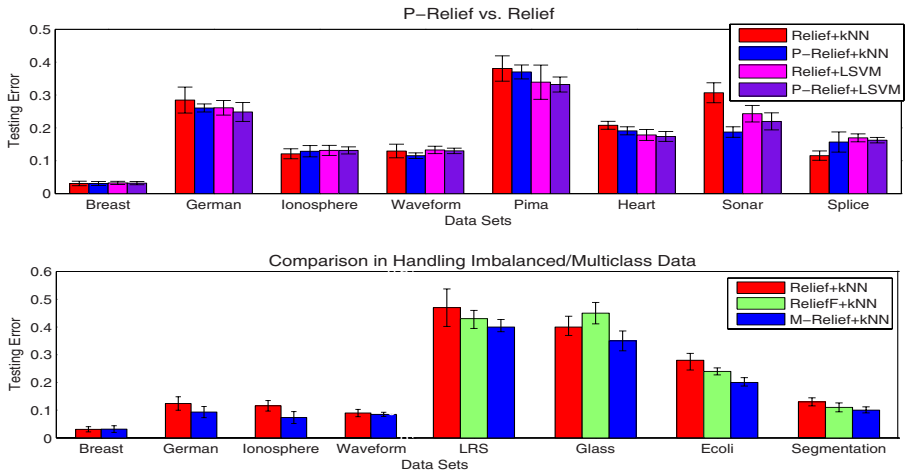


Fig. 3. Best Average Testing Errors and Standard Deviations

one nearest hit and -1 nearest misses (one for each class) are used in ReliefF. This configuration ensures that the differences are mainly resulted by the strategies used to handling imbalance.

ReliefF is relatively more time-consuming than the other two algorithms since it needs searching nearest miss for each class. However, the differences are not very significant because the number of classes are not very large. For convenience, only the k NN classifier is used to estimate the classification error. The top two lines of Fig. 2 show the testing error of each approach, as a function of the number of top-ranked features. The ROC curves are plotted in the bottom two lines. And the bar plot indicating the best average testing errors and standard deviations are also shown in Fig. 3. Note that for binary classification, ReliefF and Relief are identical to each other. From these results, we arrive at the following observations: () the performance of Relief is degraded significantly when the data is highly imbalanced; (.) M-Relief, with a simple trick that does not introduce much extra computation, improves the performance significantly. It performs the best in six (out of eight) data sets with respect to the classification error metric, and in seven data sets with respect to the ROC curve metric.

6 Conclusion

A natural optimal criterion for feature selection would be the Bayes error minimization in the reduced space, because the generalization error of any classifier is lower bounded by Bayes error, hence, the Bayes error only depends on features rather than classifiers. However, this criterion is difficult in practice where only training data is given. This paper has presented an algorithmic framework for feature selection based on nonparametric Bayes error minimization. When feature weighting are used as the search strategy, this framework reveals that

the Relief algorithm greedily attempts to minimize Bayes error estimated by k NN estimator. As an alternative, we have presented a new algorithm named Parzen-Relief. In addition, to enhance its ability to deal with imbalanced and/or multiclass data, we have proposed a MAP-Relief algorithm.

References

1. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, Heidelberg (2006)
2. Carneiro, G., Vasconcelos, N.: Minimum Bayes Error Features for Visual Recognition by Sequential Feature Selection and Extraction. In: Proc. of CRV (2005)
3. Choi, E.: Feature Extraction Based on the Bhattacharyya Distance. Pattern Recognition 36, 1703–1709 (2003)
4. Dash, M., Liu, H.: Feature Selection for Classification. Intelligent Data Analysis 1, 1131–1156 (1997)
5. Gilad-Bachrach, R., Navot, A., Tishby, N.: Margin Based Feature Selection - Theory and Algorithms. In: Proc. of 21th ICML (2004)
6. Guyon, I., Elisseev, A.: An Introduction to Variable and Feature Selection. JMLR 3, 1157–1182 (2003)
7. Hild II, K.E., Erdogmus, D., Torkkola, K., Principe, C.: Feature Extraction Using Information-Theoretic Learning. IEEE Trans. PAMI 28(9), 1385–1392 (2006)
8. Jain, A., Zongker, D.: Feature Selection: Evaluation, Application, and Small Sample Performance. IEEE Trans. PAMI 19(2), 153–158 (1997)
9. Kira, K., Rendell, L.A.: A Practical Approach to Feature Selection. In: Proc. of 9th ICML, pp. 249–256 (1992)
10. Koller, D., Sahami, M.: Toward Optimal Feature Selection. In: Proc. of 13th ICML (1996)
11. Kononenko, I.: Estimating Attributes: Analysis and Extensions of RELIEF. In: Bergadano, F., De Raedt, L. (eds.) ECML 1994. LNCS, vol. 784, pp. 171–182. Springer, Heidelberg (1994)
12. Mangasarian, O.L., Musicant, D.R.: Lagrangian Support Vector Machines. JMLR 1, 161–177 (2001)
13. Robnik-Šikonja, M., Kononenko, I.: Theoretical and Empirical Analysis of ReliefF and RRelief. Machine Learning 53(1-2), 23–69 (2003)
14. Saon, G., Padmanabhan, M.: Minimum Bayes Error Feature Selection for Continuous Speech Recognition. In: Proc. of NIPS (2002)
15. Sun, Y.J.: Iterative Relief for Feature Weighting: Algorithms, Theories, and Applications. IEEE Trans. PAMI 29(6), 1035–1051 (2007)
16. Vapnik, V.: Statistical Learning Theory. John Wiley & Sons, Chichester (1998)
17. Vasconcelos, N.: Feature Selection by Maximum Marginal Diversity: Optimality and Implications for Visual Recognition. In: Proc. of IEEE CVPR (2003)
18. Weston, J., Elisseeff, A., Schölkopf, B., Tipping, M.: Use of Zero-Norm with Linear Models and Kernel Methods. JMLR 3, 1439–1461 (2003)
19. Xuan, G., Zhu, X., Chai, P., Shi, Y., Fu, D.: Feature Selection Based on the Bhattacharyya Distance. In: Proc. of ICPR (2006)

A Framework for Modeling Positive Class Expansion with Single Snapshot

Yang Yu and Zhi-Hua Zhou

National Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, China
{yuy, zhouzh}@lamda.nju.edu.cn

Abstract. In many real-world data mining tasks, the coverage of the target concept may change as the time changes. For example, the coverage of “learned knowledge” of a student today may be different from his/her “learned knowledge” tomorrow, since the “learned knowledge” of the student is in expanding everyday. In order to learn a model capable of making accurate predictions, the evolution of the concept must be considered, and thus, a series of data sets collected at different time is needed. However, in many cases there is only a single data set instead of a series of data sets. In other words, only a *single snapshot* of the data along the time axis is available. In this paper, we show that for *positive class expansion*, i.e., the coverage of the target concept is in expanding as illustrated in the above “learned knowledge” example, we can learn an accurate model from the single snapshot data with the help of domain knowledge given by user. The effectiveness of the proposed framework is validated in experiments.

1 Introduction

In conventional machine learning and data mining research, it is assumed, explicitly or implicitly, that the test set is drawn from the same distribution of the training set and the target concept, i.e. *a posteriori* probability of class membership $p(y|\mathbf{x})$, is unchanged [4,14]. However, in many real-world applications, we may encounter problems with varying target concept. The following are two examples.

Example 1: A manufacturer has released a new product to replace its old product. After an expensive mass-advertising for one week, the manufacturer got to know that a few customers have turned to the new product. In order to reduce the cost of advertisement, the manufacturer wants to identify target customers who have high chance to turn to the new product based on an analysis on customers who have already turned to the new product. Note that here, the number of customers turning to the new product is keeping on increasing.

Example 2: A Disease Control and Prevention Center receives several fatal cases of an unclear infectious disease. A model is urgently required to predict who are potential victims, based on periodical physical examination database, in order to perform effective quarantining actions. Note that here, the victims of the disease is keeping on increasing.

In these examples, a common difficulty is that the training data is collected at an earlier time point but the predictions are to be made later, meanwhile the coverage of the target concept at different time points may be quite different. For example, in the above Example 1, suppose there were only 10 customers who had already turned to the new product (i.e., 10 *positive examples*) at the time when the data were collected; but when the model is used to make prediction, 20 more customers have already turned to the new product. Thus, if we only consider the original 10 positive examples, maybe the model we build for predicting who will turn to the new product could not meet our demand.

A possible solution to the above problem is to wait for a long period to collect a series of training sets at a series of time points, such that the pattern of the evolution can be considered. However, this may be infeasible in most cases since waiting for a long period will cause, for example, great loss of benefits in the above Example 1 and loss of human lives in the above Example 2. Moreover, data used in data mining tasks are usually *observational* [7]. That is, data is usually given by other people and the data miner could not collect more data. So, the varying target concept problem has to be addressed when there is only a *single snapshot* data set instead of a series of data sets taken at different time points.

In this paper, we propose a framework to deal with the *positive class expansion* problem, which has been illustrated in the above examples, with a single snapshot data set. Our framework has two elements. The first element is the utilization of the observation that *the instances that are currently positive become positive ahead of the instances that are currently negative*, which leads to an AUC optimization problem. The second element is the incorporation of domain knowledge expressed by user preferences of pairs of instances. These two elements are unified as an optimization problem, which is solved by the SGBDota (Stochastic Gradient Boosting with Double Target) approach. The SGBDota approach achieves success in experiments, which validate the usefulness of our framework for dealing with positive class expansion problem with single snapshot.

The rest of the paper is organized as follows. Section 2 formalizes the problem. Section 3 reviews some related work. Section 4 proposes our framework. Section 5 reports on experiments. Finally, Section 6 concludes.

2 The Problem

Given a training set of i.i.d. instances $D = \{\mathbf{x}_i\}_{i=1}^n$ drawn from a distribution \mathcal{D} . Each instance is associated with a random variable y called class, which is determined by $p(y|\mathbf{x})$. In conventional classification problem, a learning algorithm outputs a function $\hat{f}(\cdot; D, p(y|\mathbf{x}))$ that minimizes the error of a loss function L :

$$err_{\hat{f}(\cdot; D, p(y|\mathbf{x}))} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} L(\hat{f}(\mathbf{x}; D, p(y|\mathbf{x})), p(y|\mathbf{x})).$$

In the scenario of varying target concept, the training set D is collected at a time point, where we call the training set as a *snapshot* and the time point as *training time*. After trained on the snapshot, a classifier is used to make predictions at a later time point,

which we call *testing time*. From the training time to the testing time, $p(y|\mathbf{x})$ changes, while \mathcal{D} keeps steady. The evaluation of \hat{f} is therefore changed

$$\text{err}_{\hat{f}(\cdot; D, p(y|\mathbf{x}))} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}} \mathbf{L}(\hat{f}(\mathbf{x}; D, p_{tr}(y|\mathbf{x})), p_{te}(y|\mathbf{x}))$$

where p_{tr} and p_{te} indicate the probability functions at training and testing time, respectively. The formalization of positive class expansion consists of two constraints:

- 1) $f^*(\mathbf{x}) \in \{-1, +1\}$
- 2) $\forall \mathbf{x} \sim \mathcal{D} : p_{te}(y = +1|\mathbf{x}) \geq p_{tr}(y = +1|\mathbf{x})$,

which means there are only two classes, positive and negative, and the positive class is in expanding.

3 Related Works

The positive class expansion problem appears to have some relationship with *PU-Learning* [12][17], *concept drift* [9][10], and *covariate shift* [8][1]. But in fact it is very different from these tasks.

In PU-Learning, i.e., learning with positive and unlabeled data, it is required to discriminate positive instances from negative instances, while only positive instances are available in training data. A large part of works addressing PU-Learning follow two steps, e.g. [12][17]. First, strong negative instances are discovered from the unlabeled data. Then a predictive model is built from the positive and identified negative instances.

In concept drift, an online learning environment is considered, where instances are coming sequentially batch by batch, and the target concept may change in the coming batch. A desired approach for concept drift problem is the one that correctly detects and fast adapts to the drift, e.g. [9][10].

In covariate shift problem (or *sample selection bias* [13]), training and test instances are drawn from different distributions, while the *a posteriori* probability is unchanged. Using the notations in the previous section, it minimizes the error:

$$\text{err}_{\hat{f}(\cdot; D, p(y|\mathbf{x}))} = \mathbb{E}_{\mathbf{x} \sim \mathcal{D}_{te}} \mathbf{L}(\hat{f}(\mathbf{x}; D \sim \mathcal{D}_{tr}, p(y|\mathbf{x})), p(y|\mathbf{x})),$$

where \mathcal{D}_{tr} and \mathcal{D}_{te} are the distributions at training and testing time, respectively. Approaches (e.g. [8][1]) addressing this problem try to correct the bias in the training instances, such that minimizing error on the training instances corresponds to minimizing error on the test instances.

The positive class expansion problem is apparently different from the above problems. In PU-Learning problem, most works make an assumption that the positive instances in the training set are representative of the positive class concept. But in our problem, the positive class is in expanding thus are not representative. We have noticed a recent work of PU-Learning considers different training and test distributions [11]. However, that work gears heavily to text mining by using specialized mechanisms, i.e., it tries to synthesize samples by using additional keywords. In concept drift, it expects a series of data sets with information for drift detection, but in our problem, there is

no such information at all since only a single snapshot is available. In covariate shift, it assumes the *a posteriori* probability is unchanged from the training time to the testing time. On the contrary, the *a posteriori* changes in our problem.

The approach, we proposed to solve the optimization problem in our framework, is derived from Gradient Boosting [5,6]. Gradient boosting is a greedy optimization approach that avoids solving complex equations by iteratively fitting residuals of the objective function. In order to minimize an arbitrary loss function L ,

$$f^* = \arg \min_f \mathbb{E}_{\mathbf{x}} L(f(\mathbf{x}), y_{\mathbf{x}}),$$

the approach builds an additive model $F(\mathbf{x}) = \sum_{t=0}^T \beta_t h(\mathbf{x}; \theta_t)$ as the solution to the minimization problem, where θ_t is the parameter of h , β_t and θ_t are decided by

$$(\beta_t, \theta_t) = \arg \min_{(\beta, \theta)} L(F_{t-1} + \beta h(\cdot; \theta)),$$

where $F_t = F_{t-1} + \beta_t h(\cdot; \theta_t)$ and $F = F_T$. To avoid dealing with the complex loss function L , it first solves θ_t by fitting pseudo-residuals least-squarley according to

$$\theta_t = \arg \min_{\theta} \sum_{\mathbf{x} \in D} \left(h(\mathbf{x}; \theta) + \frac{\partial L(f(\mathbf{x}))}{\partial f(\mathbf{x})} \Big|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})} \right)^2,$$

and then it solves β_t according to

$$\beta_t = \arg \min_{\beta} L(F_{t-1} + \beta h(\cdot; \theta_t)).$$

4 The Proposed Framework

Since the positive class is in expanding, it is reasonable to assume that *the instances that are currently positive become positive ahead of the instances that are currently negative*. This assumption requires that all the positive instances should be ranked above the negative instances, which is exactly expressed by the AUC (*area under ROC curve*) [3] criterion. This assumption may imply the total information that we can obtain from the data set per se. Thus, we use *1 minus the AUC value* as the loss function to evaluate how the information provided by the training data is utilized:

$$L_{auc}(f) = 1 - \frac{1}{|D^+| \cdot |D^-|} \sum_{\mathbf{x}^+ \in D^+} \sum_{\mathbf{x}^- \in D^-} I(f(\mathbf{x}^+) - f(\mathbf{x}^-)). \quad (1)$$

where D^+ and D^- are subsets of D that contains all of the positive and negative instances, respectively, and $I(a)$ gets 1 if $a \geq 0$ and 0 otherwise.

Since the training data is not sufficient to build a good model in our problem, we need to incorporate domain knowledge from the user. It is not hard for the user to indicate pairwise preferences on some instances. For example, pairs of instances can be randomly drawn from the training set, and then the user is asked to judge which instance would become positive earlier in his/her opinion. Another possibility is to apply *a priori* rules to decide which instance would become positive earlier, such as *people of*

Mongoloid race may be easier to get SARS than people of Caucasoid race. Let $k(\cdot, \cdot)$ denotes the user’s pairwise preferences, such that

$$k(\mathbf{x}_a, \mathbf{x}_b) = \begin{cases} +1, & \mathbf{x}_a \text{ is preferred} \\ -1, & \mathbf{x}_b \text{ is preferred} \\ 0, & \text{equal or undecided} \end{cases} .$$

where “ \mathbf{x}_a is preferred” means that the user thought that \mathbf{x}_a would become positive earlier than \mathbf{x}_b . We then fit these preferences by the loss function:

$$L_{pref}(f) = 1 - \frac{1}{|D|^2} \sum_{\mathbf{x}_a \in D} \sum_{\mathbf{x}_b \in D} I((f(\mathbf{x}_a) - f(\mathbf{x}_b)) \cdot k(\mathbf{x}_a, \mathbf{x}_b)) \quad (2)$$

which imposes that the sign of $f(\mathbf{x}_a) - f(\mathbf{x}_b)$ should equal to the sign of $k(\mathbf{x}_a, \mathbf{x}_b)$.

Our final objective function combines Eq 1 and 2 by a prior weight λ :

$$\hat{f} = \arg \min_f L_\lambda(f) = \arg \min_f L_{auc}(f) + \lambda L_{pref}(f) \quad (3)$$

As mentioned before, we realize our framework based on Gradient Boosting [5,6]. In the original Gradient Boosting algorithm, each instance \mathbf{x} is associated to a target label $y_{\mathbf{x}}$. However, in Eq 3 each instance \mathbf{x} is associated to two targets, one is $y_{\mathbf{x}}$ while the other is determined by $k(\mathbf{x}, \cdot)$. Therefore, we build an additive model F with two base learners in each iteration, h_1 and h_2 , as:

$$F(\mathbf{x}) = \sum_{t=0}^T (\beta_{t,1} h_1(\mathbf{x}; \theta_{t,1}) + \beta_{t,2} h_2(\mathbf{x}; \theta_{t,2})) ,$$

where h_1 and h_2 fit the residuals of L_{auc} and L_{pref} , respectively.

In the first step, we solve h_1 and h_2 . In Eq 1 and Eq 2 $I(\cdot)$ is non-differentiable. We use sigmoid function $(1 + e^{-a})^{-1}$ to replace the identification function $I(a)$. So the loss functions are rewritten as:

$$L_{auc}(f) = 1 - \frac{1}{|D^+| \cdot |D^-|} \sum_{\mathbf{x}^+ \in D^+} \sum_{\mathbf{x}^- \in D^-} \left(1 + e^{-(f(\mathbf{x}^+) - f(\mathbf{x}^-))}\right)^{-1} \quad (4)$$

$$L_{pref}(f) = 1 - \frac{1}{|D|^2} \sum_{\mathbf{x}_a \in D} \sum_{\mathbf{x}_b \in D} \left(1 + e^{-(f(\mathbf{x}_a) - f(\mathbf{x}_b)) \cdot k(\mathbf{x}_a, \mathbf{x}_b)}\right)^{-1} \quad (5)$$

The residual of L_{auc} for each positive instance $x \in D^+$ is

$$\tilde{y}_{\mathbf{x}} = - \left. \frac{\partial L_{auc}(f)}{\partial f(\mathbf{x}^+)} \right|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})} \propto \sum_{\mathbf{x}^- \in D^-} \frac{e^{-(F_{t-1}(\mathbf{x}) - F_{t-1}(\mathbf{x}^-))}}{\left(1 + e^{-(F_{t-1}(\mathbf{x}) - F_{t-1}(\mathbf{x}^-))}\right)^2} \quad (6)$$

and for each negative instance $x \in D^-$ is

$$\tilde{y}_{\mathbf{x}} = - \left. \frac{\partial L_{auc}(f)}{\partial f(\mathbf{x}^-)} \right|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})} \propto - \sum_{\mathbf{x}^+ \in D^+} \frac{e^{-(F_{t-1}(\mathbf{x}^+) - F_{t-1}(\mathbf{x}))}}{\left(1 + e^{-(F_{t-1}(\mathbf{x}^+) - F_{t-1}(\mathbf{x}))}\right)^2} \quad (7)$$

The residuals of L_{pref} is:

$$\tilde{k}_{\mathbf{x}} = - \left. \frac{\partial L_{pref}(f)}{\partial f(\mathbf{x})} \right|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})} \propto \sum_{\mathbf{x}_a \in D} \frac{-k(\mathbf{x}, \mathbf{x}_a) e^{-(F_{t-1}(\mathbf{x}) - F_{t-1}(\mathbf{x}')) \cdot k(\mathbf{x}, \mathbf{x}_a)}}{(1 + e^{-(F_{t-1}(\mathbf{x}) - F_{t-1}(\mathbf{x}')) \cdot k(\mathbf{x}, \mathbf{x}_a)})^2} \tag{8}$$

Then, fitting to the residuals least-squarely, we have

$$\theta_{t,1} = \arg \min_{\theta} \sum_{\mathbf{x} \in D} (\tilde{y}_{\mathbf{x}} - h(\mathbf{x}; \theta))^2, \quad \theta_{t,2} = \arg \min_{\theta} \sum_{\mathbf{x} \in D} (\tilde{k}_{\mathbf{x}} - h(\mathbf{x}; \theta))^2$$

Next, defining $\beta \doteq (\beta_1, \beta_2)$, we solve $\beta_{t,1}$ and $\beta_{t,2}$ that minimize

$$(\beta_{t,1}, \beta_{t,2}) = \arg \min_{\beta} L_{\lambda}(F_{t-1} + \beta_1 h_1(\cdot; \theta_{t,1}) + \beta_2 h_2(\cdot; \theta_{t,2}))$$

To do the minimization, we solve

$$G(\beta) \doteq \frac{\partial L_{\lambda}(F_{t-1} + \beta_1 h_1(\cdot; \theta_{t,1}) + \beta_2 h_2(\cdot; \theta_{t,2}))}{\partial \beta} = 0$$

by the Newton-Raphson iteration with one step:

$$\left\{ \begin{array}{l} \frac{\partial G(\beta)_1}{\partial \beta_1} \delta_1 + \frac{\partial G(\beta)_1}{\partial \beta_2} \delta_2 + G(\beta)_1 = 0 \\ \frac{\partial G(\beta)_2}{\partial \beta_1} \delta_1 + \frac{\partial G(\beta)_2}{\partial \beta_2} \delta_2 + G(\beta)_2 = 0 \end{array} \right|_{\beta = \beta^0 = (1, \lambda)} \tag{9}$$

where δ_1 and δ_2 are *corrections* of β_1 and β_2 , respectively. Also note that $G(\mathbf{w})$ is a vector, $G(\mathbf{w})_1$ and $G(\mathbf{w})_2$ are its first and second elements, respectively, and $(1, \lambda)$ is set as the initial guess of β . After δ_1 and δ_2 have been solved, we have

$$\beta_1 = 1 - \delta_1, \quad \beta_2 = \lambda - \delta_2. \tag{10}$$

Before forming the approach, There are two important issues to be dealt with. One is that, when the indication function $\mathbf{I}(\cdot)$ is replaced by the sigmoid function, there have different behaviors between Eq. 1 and Eq. 4. Consider two instances \mathbf{x}_a and \mathbf{x}_b with $y_{\mathbf{x}_a} = +1$ and $y_{\mathbf{x}_b} = -1$. By Eq. 1, f receives no punishment as long as $f(\mathbf{x}_a) > f(\mathbf{x}_b)$. However, by Eq. 4, f always receives punishment even if $f(\mathbf{x}_a) > f(\mathbf{x}_b)$. When \mathbf{x}_a and \mathbf{x}_b are equal or very close¹ according to the user’s preference, the objective function will still pay much attention on ranking \mathbf{x}_a before \mathbf{x}_b , which makes the built model over complex. We handle this issue by removing a part of negative instances that are closest to the positive instances according to the user’s preference, when fitting the model for AUC residuals.

The other issue is that some instances have either y values or preferences, but not both. We fit the model of AUC on instances where y values are available, fit the model for user preference on instances where preferences are available, and calculate the combination weights of the two models on the intersection of the instances.

¹ According to the user’s preferences, we determine how close \mathbf{x}_a is to \mathbf{x}_b by counting how many instances that are either more preferred or less preferred than both of \mathbf{x}_a and \mathbf{x}_b .

Table 1. The SGBDota Approach

 ALGORITHM (D, T, h, λ, p, ν)

D : training data; T : number of iterations; h : base learner; λ : balance parameter
 p : proportion of negative instances to be removed; ν : shrinkage

1. $D_{auc} \leftarrow \{\mathbf{x} \in D \mid y_{\mathbf{x}} \text{ is available}\}$
 2. $D_{pref} \leftarrow \{\mathbf{x} \in D \mid \text{preference is available}\}$
 3. $D_{auc} \leftarrow D_{auc} - \{\mathbf{x} \in D_{auc}^- \mid \mathbf{x} \text{ is in the most preferred } p \text{ percent of } D_{auc}^-\}$
 4. $D_b \leftarrow D_{auc} \cap D_{pref}$
 5. $F_0(\mathbf{x}) \leftarrow 0$
 6. **for** $t \leftarrow 1$ **to** T
 // calculate residuals
 7. $\forall \mathbf{x} \in D_{auc} : \tilde{y}_{\mathbf{x}} = - \left. \frac{\partial L_{auc}(f)}{\partial f(\mathbf{x})} \right|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})}$ by Eq 6 and 7
 8. $\forall \mathbf{x} \in D_{pref} : \tilde{k}_{\mathbf{x}} = - \left. \frac{\partial L_{pref}(f)}{\partial f(\mathbf{x})} \right|_{f(\mathbf{x})=F_{t-1}(\mathbf{x})}$ by Eq 8*// fit models*
 9. $\theta_{t,1} \leftarrow \arg \min_{\theta} \sum_{\mathbf{x} \in \text{Sample}(D_{auc})} (h_1(\mathbf{x}; \theta) - \tilde{y}_{\mathbf{x}})^2$
 10. $\theta_{t,2} \leftarrow \arg \min_{\theta} \sum_{\mathbf{x} \in \text{Sample}(D_{pref})} (h_2(\mathbf{x}; \theta) - \tilde{k}_{\mathbf{x}})^2$*// calculate combination weights*
 11. $(\beta_{t,1}, \beta_{t,2}) \leftarrow \arg \min_{(\beta_1, \beta_2)} L_{\lambda}(F_{t-1} + \beta_1 h_1(\cdot; \theta_{t,1}) + \beta_2 h_2(\cdot; \theta_{t,2}))$
 by Eq 9 and 10 on D_b
 // update learner
 12. $F_t(\cdot) \leftarrow F_{t-1}(\cdot) + \nu(\beta_{t,1} h_1(\cdot; \theta_{t,1}) + \beta_{t,2} h_2(\cdot; \theta_{t,2}))$
 13. **end for**
 14. **return** $F_T(\cdot)$
-

Table 1 presents the SGBDota (Stochastic Gradient Boosting with Double Target) algorithm. Note that in line 3, D_b might be small because it contains only the instances where y values and preferences are both available. But since D_b is only used to determine the ‘step size’ of the greedy search in line 11, it is unlikely to cause a great effect, especially when this effect will be further reduced by shrinkage parameter in line 12. In lines 9 and 10, the base learners are trained on a sample of the training data, but not on the training data directly, which is the essential part of Stochastic Gradient Boosting [6]. In line 12, the shrinkage is used to prevent from overfitting.

SGBDota have several parameters. λ can be set according to the user’s confidence of the domain knowledge, ν could be set as 0.01 [6]. We eliminate p by a simple strategy: we run SGBDota with different p , and choose the value with which the preference is best fitted, i.e., the minimum L_{pref} that SGBDota reaches. Here L_{auc} is not involved in choosing of p because it contains no information about the expansion.

5 Experiments

5.1 Experimental Setting

In order to visualize the behavior of the proposed approach, we generate a 2-dimensional synthetic data set, by sampling 1,000 instances from four Gaussian models. In order to

evaluate the performance of the proposed approach on real-world data sets, we derive four data sets from the UCI Machine Learning Repository [2].

There are three classes in *postoperative*, i.e., *I*: patient sent to Intensive Care Unit, *A*: patient sent to general hospital floor, and *S*: patient prepared to go home. We use only *I* as the positive class at training time, and *I* plus *A* as the positive class at testing time.

segment contains seven classes of outdoor images, i.e., *brickface*, *sky*, *cement*, *window*, *path*, *foliage*, and *grass*. We set *grass* as the positive class at training time, and *grass+foliage+path* as the positive class at testing time.

veteran is the veteran's administration lung cancer trial data. We set instances with survival time less than 12 hours as positive at training time, and instances with survival time within 48 hours as positive at testing time.

pcb is the data recorded from the Mayo Clinic trial in primary biliary cirrhosis, of the liver conducted between 1974 and 1984. We set instances with living time within 365 days and 1460 days as positive at training and testing time, respectively.

On data sets *postoperative*, *veteran*, *segment* and *pcb*, we randomly split 2/3 of the instances to be training set, and the rest 1/3 instances are used for test. Note that the class labeling is different for training set and test set. Any learner is evaluated on the test set using AUC criterion. The split is repeated 20 times to obtain an average performance.

We try three assumptions of "domain knowledge" for experiment. The first one assumes that the positive class expands from dense positive area to spares positive area, the second one assumes the positive class expands from dense positive area to spares positive and spares negative area. For efficiency, the density is estimated by 200 times of random space splits and counting instances in the local region. The third one assumes the positive class expands along with the neighborhoods using linear neighborhoods propagation [15] (positive label is +1 and negative label is zero). We name the SGBDota with the three assumptions as SGBDota-1, SGBDota-2, and SGBDota-3. Nevertheless, it is expected to incorporate real domain knowledge in real world applications.

The default parameter setting of SGBDota is: $T = 50$, $\nu = 0.01$ approximately be the log-median of the suggested range $[0.005, 0.05]$ from [6], h is set to be a regression tree [16] for its efficiency, $\lambda = 1$, and p is searched from $\{1, 0.95, 0.9, 0.6, 0.3, 0\}$ on training set by the search strategy mentioned before, where $p = 1$ means only the negative instances with the lowest preference are kept undeleted for minimizing L_{auc} .

Approaches compared to SGBDota include Random Forests, PU-SVM [17], SG-BAUC and Random. Random Forests includes 100 trees. PU-SVM uses RBF kernel. SG-BAUC is AUC optimization by stochastic gradient boosting with shrinkage 0.01 and 50 iterations. Random is the random guess approach, which serves as a lower bound. All the other default parameters are taken from WEKA [16].

5.2 Comparison with Other Methods

First, we visualize the behavior of our approach by the synthetic data set. The data set is plotted in Fig 1(a), the instance space is $[0, 1] \times [0, 1]$. Four approaches, Random Forests, PU-SVM, SG-BAUC, and SGBDota-1, are trained on the data set. Then, the ranking decision of each approach is probed by testing on every instance $x = (x_1, x_2) \in \{0, 0.01, \dots, 1\} \times \{0, 0.01, \dots, 1\}$, from which bitmaps are constructed and displayed in Fig 1(b), (c), (d), and (e), using histogram equalization.

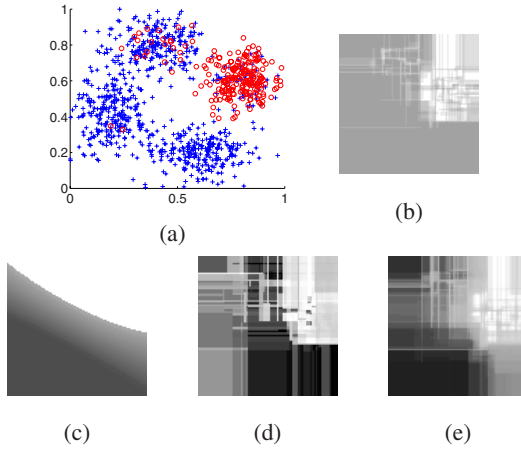


Fig. 1. (a) The synthetic data set, where red cycles indicate positive instances; (b)-(e) show the ranking decisions of Random Forests (b), PU-SVM (c), SGBAUC (d), and SGBDota-1 (e), respectively, where the more white the more positive.

Table 2. AUC values of SGBDota, Random Forests (RF), PU-SVM, SGBAUC, and Random. Each cell presents a *mean ± standard derivation*

Data set	SGBDota-1	SGBDota-2	SGBDota-3	RF	PU-SVM	SGBAUC	Random
posto	.470±.131	.483±.111	.459±.132	.448±.076	.457±.107	.457±.084	.456±.148
segment	.821±.031	.822±.029	.744±.025	.750±.014	.753±.020	.744±.012	.506±.018
veteran	.658±.118	.650±.115	.544±.094	.637±.102	.627±.146	.658±.093	.522±.069
pbc	.721±.034	.726±.032	.638±.054	.710±.043	.709±.033	.665±.041	.503±.043

Table 3. Win/tie/loss counts of SGBDota against Random Forests (RF), PU-SVM, SGBAUC, and Random, by pairwise *t*-tests at 95% significance level

	RF	PU-SVM	SGBAUC	Random
SGBDota-1	1/3/0	1/3/0	2/2/0	3/1/0
SGBDota-2	2/2/0	2/2/0	2/2/0	3/1/0
SGBDota-3	0/2/2	0/2/2	0/2/2	0/2/2

In Fig 1(a), since the expansion is from regions of high positive density to low positive density, we expect that there are two expanding paths, one is from the right cluster to the top cluster and then to the left cluster, the other is from the right cluster to the bottom cluster. From Fig 1(b), it is observed that Random Forests does not find the expanding path from the right cluster to the bottom cluster. From Fig 1(c), PU-SVM does not find the expanding path from the right cluster to the top cluster. From Fig 1(d),

SGBAUC ignores the path from the left cluster to the bottom cluster. Finally, SGBDota concerns all the expanding paths, as shown in Fig 1(e).

Results of comparisons between SGBDota and the other methods are presented in Table 2. Since each approach is tested 20 times on each data set, for two approaches in comparison, we employ a pairwise two-tail *t*-test with significance level at 0.05 to test if they have significant differences. A win/loss is counted SGBDota is significantly better/worse than the comparing approach on a data set, otherwise a tie is counted. Table 3 lists the counts. It can be found that SGBDota-1 and SGBDota-2 never lose, and are better than all the other approaches on *segment*. SGBDota-2 is moreover better than all the other approaches on *pbc*. This indicates that, with effective domain knowledge, our approach can exceed the stat-of-the-arts learning approaches.

5.3 Influence of Parameters

We study the influence of the parameter *p*. The performances of SGBDota with $p = \{1, 0.95, 0.9, 0.6, 0.3, 0\}$, are depicted in Fig 2, where the performance of Random Forests, PU-SVM, SGBAUC, and Random are also plotted.

On *segment* and *pbc*, there is a large flat area where SGBDota-1 and SGBDota-2 have the best performance. On *postoperative* and *veteran*, SGBDota-1 and SGBDota-2 have less chance to be the best, which may be because the domain knowledge we used here is not strong. We also note that *postoperative* is a hard data set, where Random Forests is worse than Random, and SGBAUC can only achieve equal performance with Random, but SGBDota-1 and SGBDota-2 have a significant chance to exceed Random.

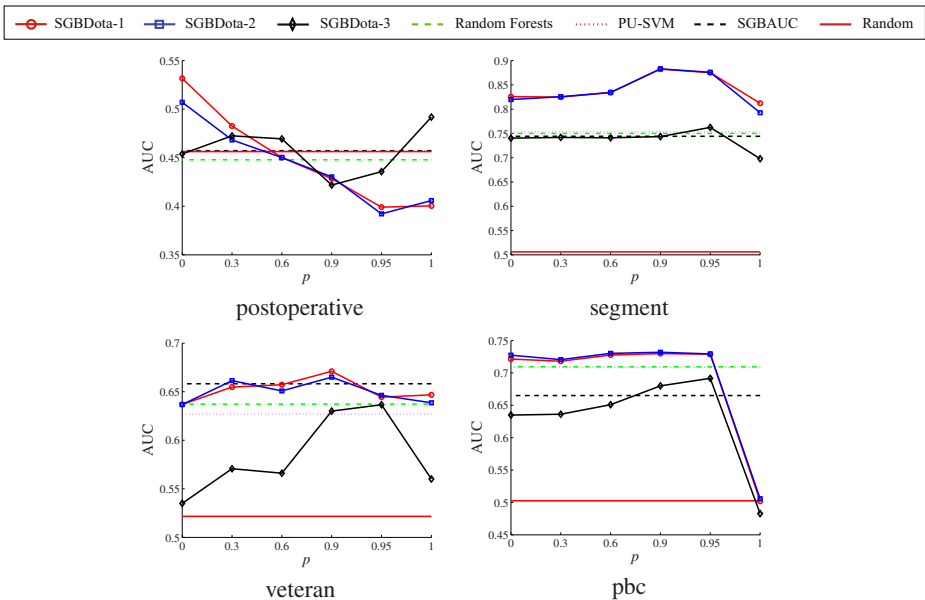


Fig. 2. The influence of the parameter *p* on the performance of SGBDota

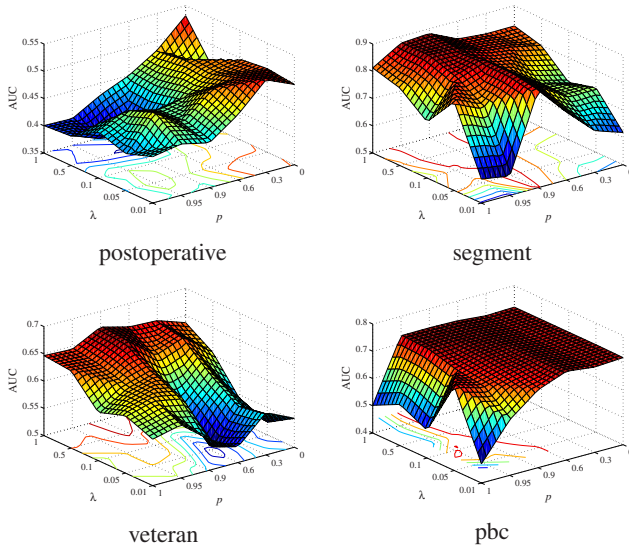


Fig. 3. The influence of the parameters λ and p on the performance of SGBDota

Then, we study how the parameter λ affects the performance of SGBDota. Here we use SGBDota-1 as a representative of the SGBDota approach. We test it by varying p in $\{1, 0.95, 0.9, 0.6, 0.3, 0\}$ and λ in $\{0.01, 0.05, 0.1, 0.5, 1\}$. Fig. 3 plots the test results of SGBDota on the four UCI data sets.

From Fig. 3, it can be observed that $\lambda = 1$ is better than smaller value, which is reasonable because if the used domain knowledge is useful, it should be heavily weighted, otherwise the knowledge is probably fake. While it is hard to choose a good fixed value of p , because when the domain knowledge is strong, p needs to be large to reduce the side-effect of optimizing L_{auc} , otherwise p should be small. Therefore, in practical use, it is convenient to let $\lambda = 1$, and set p according to user's confidence about the domain knowledge or leaving p be determined by the search strategy.

6 Conclusion

In this paper, we propose a framework to deal with positive class expansion problem with single snapshot. Our framework includes two elements, i.e., the utilization of the observation that *the instances that are currently positive become positive ahead of the instances that are currently negative*, and the incorporation of domain knowledge expressed as user preferences of pairs of instances. We formulate the problem as an optimization problem, and propose the SGBDota (Stochastic Gradient Boosting with Double Target) approach which achieves success in experiments.

In our future work we will try to apply our approach to some real-world tasks which suffer from the positive class expansion problem. We will also try to extend the proposed framework to other varying target concept problems, such as the positive class shrinking problem.

Acknowledgement

This research was supported by the National Science Foundation of China (60635030, 60721002), the National High Technology Research and Development Program of China (2007-AA01Z169), and the Foundation for the Author of National Excellent Doctoral Dissertation of China (200343).

References

1. Bickel, S., Brückner, M., Scheffer, T.: Discriminative learning for differing training and test distributions. In: Proceedings of the 24th International Conference on Machine Learning, Corvallis, OR, pp. 81–88 (2007)
2. Blake, C.L., Keogh, E., Merz, C.J.: UCI repository of machine learning databases. Department of Information and Computer Science, University of California, Irvine, CA (1998), <http://www.ics.uci.edu/~mllearn/MLRepository.html>
3. Bradley, A.P.: The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition* 30(7), 1145–1159 (1997)
4. Duda, R.O., Hart, P.E., Stork, D.G.: *Pattern Classification*, 2nd edn. John Wiley and Sons, New York (2001)
5. Friedman, J.H.: Greedy function approximation: A gradient boosting machine. *Annals of Statistics* 29(5), 1189–1232 (2001)
6. Friedman, J.H.: Stochastic gradient boosting. *Computational Statistics & Data Analysis* 38(4), 367–378 (2002)
7. Hand, D., Mannila, H., Smyth, P.: *Principles of Data Mining*. MIT Press, Cambridge (2001)
8. Huang, J., Smola, A.J., Gretton, A., Borgwardt, K.M., Schölkopf, B.: Correcting sample selection bias by unlabeled data. In: Schölkopf, B., Platt, J., Hoffman, T. (eds.) *Advances in Neural Information Processing Systems 19*, pp. 601–608. MIT Press, Cambridge (2007)
9. Klinkenberg, R., Joachims, T.: Detecting concept drift with support vector machines. In: Proceedings of the 17th International Conference on Machine Learning, Stanford, CA, pp. 487–494 (2000)
10. Kolter, J.Z., Maloof, M.A.: Dynamic weighted majority: A new ensemble method for tracking concept drift. In: Proceedings of the 3rd IEEE International Conference on Data Mining, Los Alamitos, CA, pp. 123–130 (2003)
11. Li, X., Liu, B.: Learning from positive and unlabeled examples with different data distributions. In: Proceedings of the 16th European Conference on Machine Learning, Porto, Portugal, pp. 218–229 (2005)
12. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially supervised classification of text documents. In: Proceedings of the 19th International Conference on Machine Learning, Sydney, Australia, pp. 387–394 (2002)
13. Shimodaira, H.: Improving predictive inference under covariate shift by weighting the log-likelihood function. *Journal of Statistical Planning and Inference* 90(2), 227–244 (2000)
14. Vapnik, V.: *Statistical Learning Theory*. John Wiley and Sons, New York (1998)
15. Wang, F., Zhang, C.: Label propagation through linear neighborhoods. In: Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, pp. 985–992 (2006)
16. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)
17. Yu, H., Han, J., Chang, K.C.C.: PEBL: Positive example based learning for web page classification using svm. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Alberta, Canada, pp. 239–248 (2002)

A Decomposition Algorithm for Learning Bayesian Network Structures from Data

Yifeng Zeng and Jorge Cordero Hernandez

Dept. of Computer Science, Aalborg University, DK-9220 Aalborg, Denmark
{yfzeng, jorgecordero}@cs.aau.dk

Abstract. It is a challenging task of learning a large Bayesian network from a small data set. Most conventional structural learning approaches run into the computational as well as the statistical problems. We propose a decomposition algorithm for the structure construction without having to learn the complete network. The new learning algorithm firstly finds local components from the data, and then recover the complete network by joining the learned components. We show the empirical performance of the decomposition algorithm in several benchmark networks.

Keywords: Bayesian Networks, Graphical Models, Structure Learning.

1 Introduction

Bayesian networks (BN) [1,2] are widely used to represent probabilistic relationship among random variables. They have been successfully applied in many domains such as medical diagnosis, gene data analysis, and hardware troubleshooting. Over the last decade, much progress has been made regarding structural learning in Bayesian networks including both the score-based and the constraint-based learning methods [3,4,5,6]. The score-based method tries to optimize a scoring function by means of a search strategy. Since finding the optimal Bayesian networks was shown to be an NP-complete problem [7], one has to resort to some heuristic search strategy. The other is constraint-based and infers structures through conditional independency tests. The constraint-based is generally faster than the score-based method and gives a trustworthy result provided there are sufficient data. We focus on the constraint-based learning methods in this paper. Currently, with the mass-throughput data in biomedical informatics, data analysis demands more powerful learning algorithms that could handle data sets having thousands of variables but with limited sample sizes. Most conventional learning methods run into the computational and statistical problems in such a domain. They either can't complete the learning process, or produce a poor structure even when the learning is done. Hence, in this paper, we propose a decomposition algorithm for learning a large Bayesian network from a small amount of data.

The decomposition learning algorithm adopts the divide-and-conquer strategy and contains several procedures to complete the learning task. We discover a set of clusters from a dependency graph built directly from the data. Each

cluster is expected to represent a local domain structure. We establish connection between clusters and learn each cluster separately. The learned clusters are joined together to compose the complete targeted network.

The novel algorithm reduces the computational complexity since it learns clusters instead of the complete network directly. In addition, the algorithm learns clusters in a separated way so that the structural error (due to conditional independency tests) that occurs in the learning of clusters does not influence the global structure learning. Hence, the algorithm avoids the cascading effect of incorrect statistical test results in the structural learning. We experiment the proposed algorithm on several benchmark Bayesian networks and compare with other typical constraint-based learning algorithms. The empirical results show that the decomposition algorithm achieves good performance regarding both structure learning accuracy and run times.

This paper is organized as follows. In Section 2, we discuss some related works on structural learning. In Section 3, we present the decomposition learning algorithm by illustrating embeded procedures. Then, we show comparison results in Section 4. Finally, in Section 5, we conclude the paper with some hint on future work.

2 Related Work

The divide-and-conquer strategy has served as the technique of many learning algorithms that aim at recovering a large Bayesian network structure from data [8,9,10]. The foundation of this strategy is to identify some appropriate components in a large model. For example, in the approach of learning module networks [9], a module is defined as a set of variables that have similar behavior. All variables in a module share both the same parents and the same conditional probability distribution. It seems that the module formulation is quite strict. However, the formulation is well consistent with some domain concepts such as genes in a cell, stocks in a stock sector, and so on. The sparse candidate algorithm [8] recovers Bayesian networks by specifying the maximum number of parents of variables in the learning process, which significantly reduces the learning complexity. Both the module learning and the sparse candidate approaches orient score-based learning.

The most relevant work is the block learning algorithm [10] that already shows the ability of learning a large Bayesian network from limited data. Similarly, the block learning algorithm recovers structures through procedures of identifying blocks and combining the learned blocks. However, the block identification procedure is incomplete since the block is composed of nodes that have at most two-length distance from block centers. The searching largely depends on the topology of a dependency graph and probably leads to disconnected blocks in some domains. In addition, as shown in the previous work, a large amount of run times are spent in learning overlapping structures. However, the extra procedure does not have much benefit to the final (heuristic) structural combination. Our new algorithm improves the block learning algorithm by designing more robust and appropriate procedures.

3 Decomposition Learning Algorithms

A constraint-based approach learns a network structure by using some statistical hypothesis tests to detect dependency or (conditional) independency among variables or attributes in a data set. The results of several tests are combined by the constraint-based approach in order to construct a Bayesian network structure. The test results might be incorrect especially when insufficient data are provided. Since various test results might depend on each other in some unknown manner, the error of the induced structure is not under control and spread in a global way. In addition, a large number of variables lead to an increasing order of conditional independency tests, which makes the learning intractable. To circumvent these shortcomings, we propose the decomposition algorithm that enhances the learning ability of conventional learning techniques by specifying a modular framework.

We briefly describe the decomposition learning algorithm in Fig. 1. The algorithm receives the data set D of size l and the parameter ε used to control the cluster expansion (line 4). We firstly construct the dependency graph M directly from the data through the procedure of building a dependency graph (BDG)(line 2). The first procedure also produces two sets of edge weights, W^M and W^G , for the dependency graph M and the complete graph G respectively. Then, we partition M into several disjoint clusters using the procedure of star discovery (SD)(line 3). The procedure is highly motivated by current research work on complex network [11,12] and is rather reliable to generate consistent clusters. The disjoint clusters reveal some local components that shall be connected in the domain. Hence, we expand the clusters into a set of overlapping clusters by discovering a high correlation between inter-cluster memberships. The procedure of cluster expansion (CE) uses the parameter ϵ to control the proportion of overlapping variables in the truly correlated clusters (line 4). We proceed to learn a Bayesian knot for each overlapping cluster separately by structuring the relation of cluster variables. The procedure of learning Bayesian knots (LBK) may utilize any of available structural learning algorithms (line 5). Finally, we use structural rules to combine the learned Bayesian knots and recover the complete Bayesian network structure B in which a set of nodes V^B are connected with directed edges E^B .

3.1 Build a Dependency Graph

Bayesian network structures exhibit the dependency among variables in a data set. A strong dependency always gathers the variables into one local component. In other words, the tightly linked variables are potential nodes that will be enclosed in the same cluster. We expect to build a representative dependency graph from which some sound clusters could be discovered. The graph must be able to characterize a strong dependency of domain variables through their connectivity. We select the maximum spanning tree [13] as the dependency graph M since the tree is the smallest connected graph that approximately

¹ Both the attribute x_i in data set and the node or vertex v_i in graphs represent variables in the domain. They are interchangeable and not further distinguished in this paper.

Decomposition Learning Algorithm (DL)

Input: $D = \{x_{1,l}, \dots, x_{n,l}\}$ and parameter ϵ

Output: A Bayesian network structure $B = (V^B, E^B)$

- 1: Load the data D
- 2: Build the dependency graph $M: (M, W^M, W^G) = BDG(D)$
- 3: Partition M into a set of disjoint clusters $C: C = SD(M, W^M)$
- 4: Expand C into a set of overlapping clusters $OC: OC = CE(C, W^G, \epsilon)$
- 5: Learn a set of Bayesian Knots $BK: BK = LBK(OC, D)$
- 6: Compose the final Bayesian network structure $B: B = CBK(BK)$

Fig. 1. The decomposition learning framework includes multiple procedures that will be illustrated subsequently

Build a Dependency Graph (BDG)

Input: Data $D = \{x_{1,l}, \dots, x_{n,l}\}$

Output: $M = (V^M, E^M), W^M, W^G$

- 1: Compute the complete Graph $G = \{V^G, E^G\}$ with weights
 $W^G = \{w_{i,j} = MI(v_i, v_j) | i, j = 1, \dots, n \text{ and } i \neq j\}$
- 2: $k = 0, E^M \leftarrow \emptyset, W^M \leftarrow \emptyset$ \triangleright Initialization
- 3: Sort E^G decreasingly according to the weight $w_{i,j} \in W^G$
- 4: **For** $e_{i,j} \in E^G \wedge k < |V^M|$ **do**
- 5: **If** $(\{e_{i,j}\} \cup E^M)$ do not create a cycle in M **Then**
- 6: $E^M \overset{\cup}{\leftarrow} \{e_{i,j}\}, W^M \overset{\cup}{\leftarrow} \{w_{i,j}\}, k = k + 1$

Fig. 2. The BDG procedure builds the maximum spanning tree as the dependency graph M from the data D

joint distribution of domain variables. The dependency graph $M = (V^M, E^M)$ is a tree in which each edge, $e_{i,j} \in E^M$, connects a pair of nodes v_i and v_j ($v_i, v_j \in V^M$) and the edge has the weight $w_{i,j}$ ($w_{i,j} \in W^M$) measured by the mutual information $MI(v_i, v_j)$. We show the procedure of building a dependency graph (BDG) in Fig. 2

We compute $MI(v_i, v_j)$ for all pairs of variables (for l -size samples) and construct the complete graph G (line 1). We use the hash table that shows efficient computation. The complexity of this task is in the order of $O(n^2)$. We slightly modify the Kruskal’s algorithm to build the tree M (the original Kruskal’s algorithm [14] finds the minimum spanning tree by sorting weights decreasingly instead of increasingly) (lines 3-6). We use an union-finder data structure and a sorted list for adding arcs into M . The complexity is in the order of $O(n \log n)$.

3.2 Discover Local Components

The output M is a minimal description of dependency among the variables. We opt for this dependency graph because it represents the most significant interactions in a topology that could be clustered (recall that variable clustering

in complex graphs is an NP-hard problem). Many clustering methods [15,16] have appeared and shown competitive results in some domains. However, most of them aim for different optimization problems. Moreover, they can't generate consistent clusters due to random selection of initial cluster modes. We are interested in offering a robust algorithm that clusters a set of truly dependent variables by examining a graph topology together with edge weights.

We aim to find a set of clusters C (each cluster C_i contains a set of vertices V^{C_i} in which one vertex v_j is called as cluster center node o_j) that maximize the function in Eq. 1. In other words, we want to maximize the sum of dependency weights (between cluster variables v_i and cluster nodes o_j) over multiple clusters.

$$C = \operatorname{argmax}_C \sum_{C_i \in C} \sum_{v_i \in (V^{C_i} - \{o_j\})} w_{i,j} \tag{1}$$

where o_j is the center node v_j in the cluster C_i and $w_{i,j} \in W^G$.

We use some sound graph operations to maximize Eq. 1 and show the Star Discovery (SD) procedure in Fig. 3. The idea is motivated by current research results on complex networks and evolves from the spanning star in the scale-free networks [17]. The research characterizes domain patterns in terms of connectivity of nodes, densities of clusters of nodes, and so on. It indicates that nodes of strong relations are always close and reside in a neighboring position. It suggests some hidden, but natural, domain patterns could be discovered by investigating the constructed graph topology.

We start by building a set of stars $S = \{S_1, \dots, S_n | S_i = (V^{S_i}, E^{S_i})\}$ (lines 2-7). Each star S_i is not a single node, but a connected sub-graph in the dependency

Star Discovery Procedure (SD)

Input: $M = (V^M, E^M), W^M$

Output: $C = \{C_1, C_2, \dots, C_k\}$

1: For each $v_i \in V^M$

2: $o_i \leftarrow v_i$

3: $Adj(v_i) \stackrel{\cup}{\leftarrow} \{v_j\}$ iff $e_{i,j} \in E^M$

4: $Leaf(v_j) \stackrel{\cup}{\leftarrow} \{v_h\}$ iff $e_{h,j} \in E^M \wedge v_j \in Adj(v_i) \wedge e_{h,*} \notin (E^M - \{e_{h,j}\})$

5: $V^{S_i} \stackrel{\cup}{\leftarrow} \{o_i\} \cup Adj(v_i) \cup Leaf(v_j)$

6: $E^{S_i} \stackrel{\cup}{\leftarrow} \{e_{i,j}\} \cup \{e_{h,j}\}$

7: $W^{S_i} = \sum (w_{i,j} + w_{h,j})$ \triangleright Star weights

8: While $V^S \neq \emptyset$

9: $C_k \leftarrow V^{S_i}$ iff $S_i = \operatorname{argmax}_{S_i \in S} (W^{S_i} \in W^S)$

10: $S \leftarrow (S - S_i - S_j), W^S \leftarrow (W^S - W^{S_i} - W^{S_j})$
iff $v_j \in S_i \wedge v_j = (o_j \in S_j)$

11: $C \stackrel{\cup}{\leftarrow} C_k, V^S \leftarrow (V^S - C)$

Fig. 3. The SD procedure finds a set of disjoint clusters C from M through the building of star graph S and avoids random initialization of clusters

graph. We initialize each node v_i as the star center node o_i (line 2). The center node o_i , together with its adjacent nodes $Adj(v_i)$ and leaf nodes $Leaf(v_j)$ next to the adjacent nodes v_j ($v_j \in Adj(v_i)$), composes the initial n stars (lines 3-6). In addition, we compute the star weight W^{S_i} that is the sum of weights for all edges in S_i (line 7). Then, we find a set of clusters C from the set of stars S and each cluster C_k contains only vertices V^{C_i} without edges (lines 8-11) [\[2\]](#). The star S_i that has the largest star weight W^{S_i} of all remained stars is chosen as the cluster (line 9). When the star S_i becomes a cluster it will be removed from the set S together with the stars S_j that have the center node o_j residing in the selected star S_i (line 10). Afterwards, we select the star of the second largest weight as a new cluster. Hence, we get a set of k clusters in an iterative way without having to specify the cluster number k in the initialization. The SD complexity is dominated by the building of stars and takes $O(n^3)$ operations searching for all adjacent and leaf nodes.

The SD procedure maximizes Eq. [\[1\]](#) through finding clusters that contain nodes close to cluster centers in the dependency graph. We notice the SD procedure avoids random initialization of clusters since it builds clusters by selecting the star that has the largest weight among all remained stars. Consequently, we do not need to specify the cluster number k and get consistent clusters upon one data set. This is significantly different from other clustering methods that need to assume a number of initial clusters at random.

3.3 Cluster Expansion

A cluster contains a set of most correlated variables that may compose a local component in the domain. Since the SD procedure may result in disjoint clusters we may lose some local correlations that link variables in separated clusters. In addition, we need to recover the complete network structure by joining local cluster structures. The interdependency of clusters will provide foundation in the combination phase. Hence, we proceed to expand disjoint clusters into overlapping clusters by discovering cluster interdependency.

We present the Cluster Expansion (CE) procedure in Fig. [\[4\]](#). The basic idea is to expand clusters by including outlier variables that have most strong dependency with cluster memberships. The procedure uses two phases, cluster expansion (lines 1-6) and region expansion (lines 7-14), to generate a set of overlapping clusters. In the first phase, we use the parameter ϵ to control the number of overlapping variables for possibly expanded clusters (line 3). For each cluster C_i , we identify $\lceil |C_i| * \epsilon \rceil$ (the ceil function $\lceil \cdot \rceil$) numbers of outlier variables v_j that have the most strong dependency with cluster variables v_i by measuring their weights (line 4), and include these outlier variables into the targeted cluster (line 6). The complexity of this phase is governed by the searching of relevant variables in k disjoint clusters and is in the order of $O(ksn)$ where s is the maximal cardinality of any given cluster C_i .

² Since a cluster contains only vertices we sometimes use C_i as V^{C_i} depending on the context.

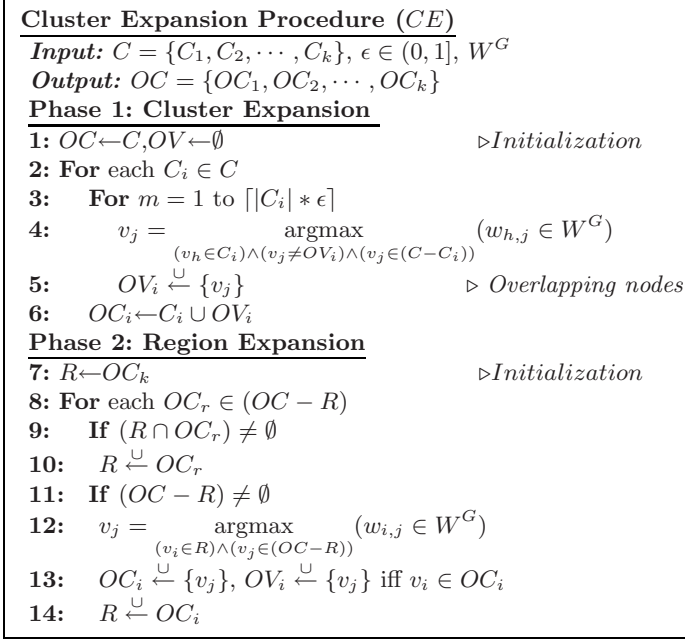


Fig. 4. The CE procedure expands disjoint clusters by absorbing most relevant outlier variables into targeted clusters

In the first phase, clusters are expanded through absorbing a limited number of outlier variables that have strong dependency with cluster memberships. Consequently, some isolated regions that contain a set of connected clusters may appear. For example, through the CE procedure, four disjoint clusters (C_1 , C_2 , C_3 , and C_4) may result in two isolated regions, $(OC_1 \cup OC_2)$ and $(OC_3 \cup OC_4)$. The cluster C_1 locates $\lceil |C_1| * \epsilon \rceil$ most relevant variables all of which reside in the cluster C_2 , and the cluster C_2 finds all the most relevant variables in the cluster C_1 ; so do the clusters C_3 and C_4 . We need to remedy the cluster expansion phase to ensure the cluster reachability (direct or undirect) if it is necessary.

In the same vein as cluster expansion phase, the second phase expands isolated regions by including (region) outlier variables that have the most dependency with region variables. We compose the region R by connecting the clusters (from the first phase) that have already shared some overlapping variables (lines 9-10). Then, we detect possible isolated regions (line 11). If such regions exist we need to connect them by adding the most relevant outlier variables v_j into the targeted cluster (lines 12-13). We also get the byproduct of a set of overlapping nodes OV_i (line 13). The complexity of the second phase is dominated by the searching of relevant nodes in possibly isolated regions and is in the order of $O(nm)$ where m is the maximum number of variables within one region.

Learning Bayesian Knots(LBK)
Input: Data D , Clusters OC
Output: $BK = \{BK_1, BK_2, \dots, BK_k\}$
1: Load the data D
2: For each $OC_i \in OC$
3: Construct BK_i using any structural learning algorithm
4: $BK \stackrel{\cup}{\leftarrow} BK_i$

Fig. 5. The LBK procedure learns Bayesian knots (much smaller than the complete network) using any of available structural learning algorithms and recovers local domain structures

3.4 Recover Bayesian Network Structures

The CE procedure expands the disjoint clusters so that each cluster is connected to at least one of other clusters. We proceed to learn a set of Bayesian Knots (BK) by structuring relations of variables in clusters. We describe the learning Bayesian knots (LBK) procedure in Fig. 5. The procedure receives the input of the data set D and a set of overlapping clusters OC (line 1). We apply any of available structural learning algorithms to construct a Bayesian knot (BK) that is a directed acyclic graph (line 3). Each BK_i contains a set of nodes V^{BK_i} connected by directed edges E^{BK_i} and could be viewed as a local structure in the domain. The procedure complexity relies on the selected learning algorithm (line 3). For example, if the PC algorithm is used the complexity is in the order of $O(kr^q)$ for learning k clusters where q is the maximum number of parents for a node and r is the largest cluster size. In general, a cluster contains a small subset of domain variables ($r \ll n$). The complexity is relatively low comparing with the order of $O(n^q)$ for learning the complete network directly.

The final procedure is to complete the learning task by joining the learned Bayesian knots that share common variables. We show the procedure of combining Bayesian knots in Fig. 6. The procedure takes some rules to address conflicting structural problems and to avoid global directed cycles in the network. The conflict occurs when the direction of arcs connecting overlapping nodes differs in linked knots.

We start the Bayesian network B with a complete undirected graph (line 1). Then, we remove edges from the complete graph that do not exist in any of the learned Bayesian knots (line 2). All the remained edges must be directed in at least one of the Bayesian knots. We direct those edges that have already been oriented in at most one Bayesian knot (line 4). Subsequently, we use three rules to orient the rest undirected edges since the edges are directed differently in overlapping Bayesian knots (lines 5-8). The first rule is to avoid new v-structures (line 6). In most constraint-based learning methods, directions of edges participating in v-structures are uncovered using independency tests, rather than through structural rules afterwards. The second rule avoids directed cycles by forcing the arc direction (line 7). Finally, if both rules can't be applied we orient edges randomly following directions in one Bayesian knot (line 8). The combination

Combine Bayesian Knots(CBK)
Input: $BK = \{BK_1, BK_2, \dots, BK_k\}$
Output: $B = (V^B, E^B)$

- 1:** Start the global skeleton of a complete network $B = \{V^B, E^B\}$
- 2:** $E^B \leftarrow (E^B - \{e_{ij}\})$ iff $e_{ij} \notin E^{BK_i}$
- 3:** **For** each $BK_i \in BK$
- 4:** Orient $e_{ij} \in E^B$ iff $e_{ij} \in E^{BK_i} \wedge e_{ij} \notin (E^{BK} - E^{BK_i})$
- 5:** **For** all undirected edges $e_{ij} \in E^B$
- 6:** If $v_i \rightarrow v_j$, v_j and v_h are adjacent, and v_i, v_h are not adjacent, then orient $v_j - v_h$ as $v_j \rightarrow v_h$
- 7:** If there is a directed path from v_i to v_j , and v_i, v_j are adjacent, then orient $v_i - v_j$ as $v_i \rightarrow v_j$
- 8:** Otherwise, orient $v_i - v_j$ at random

Fig. 6. The *CBK* procedure joins Bayesian knots into the complete network through structural rules without furthering (in)dependency tests

procedure aims for the arc orientation using structural rules instead of expensive independency tests.

4 Experimental Results

We demonstrate the empirical performance of the decomposition learning algorithm on several benchmarks : ALARM (37 nodes), Hailfinder (56 nodes), HeparII (70 nodes), Pathfinder (109 nodes), and Andes (223 nodes). We also compare the performance with two typical constraint-based learning methods. One is the basic learning method of the *PC* algorithm [3] and the other is three phase dependency analysis (*TPDA*) [18] algorithm that is the winner of 2001 KDD cup. In addition, we compare with the block learning algorithm. We generate several data sets (ranging from small to large sample sizes) and compute the Euclidean distance (of the precision and recall from the perfect score 1) [19] between the learned structures and the benchmarks. The Euclidean distance is defined in Eq. 2.

$$distance = \sqrt{(1 - sensitivity)^2 + (1 - specificity)^2} \tag{2}$$

where the precision of the algorithm is the ratio of correctly identified edges (undirected arcs) over the total number of edges in the real network while the recall is the ratio of edges correctly identified as not belonging in the graph over the true number of edges not present in the real network.

In most cases, we show that the decomposition learning algorithm outperforms other learning algorithms and achieves lower distance values. In particular, the new algorithm keeps a good quality structure even when the data set is reduced. Furthermore, we obtain computational savings from using the decomposition algorithm as indicated by the low run times.

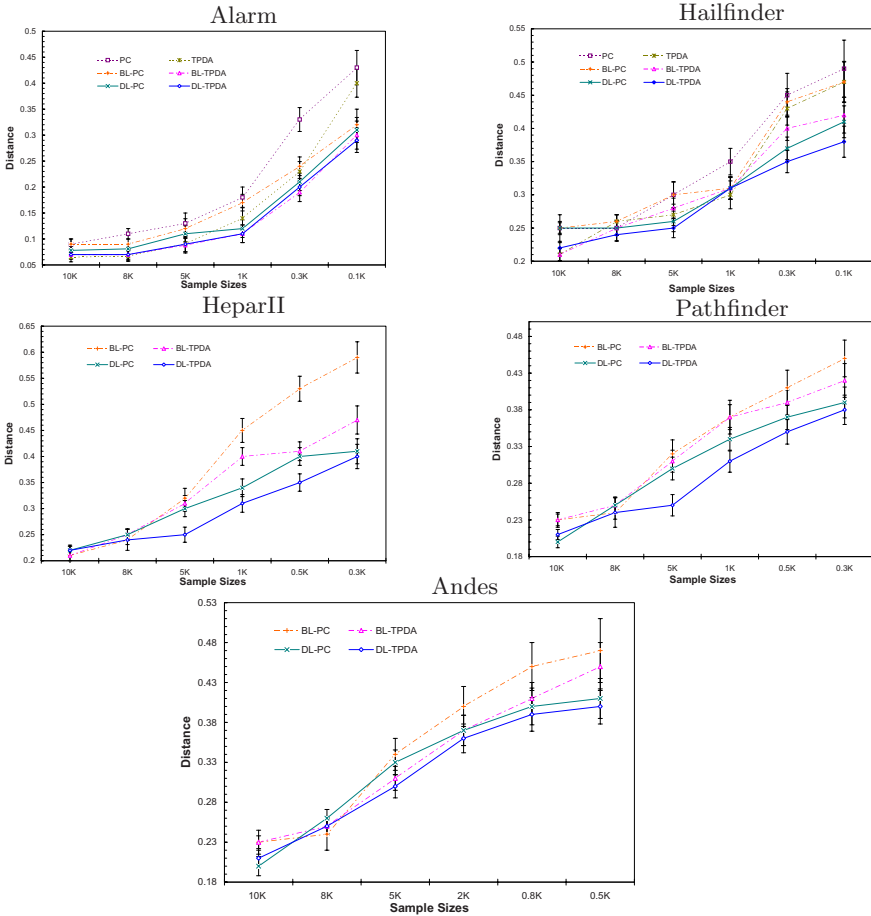


Fig. 7. Performance profile of the decomposition learning algorithm comparing with other learning algorithms. The dotted lines with different colors denote the *PC* and *TPDA* learning algorithms, the dashed lines, *BL – PC* and *BL – TPDA*, denote the block learning algorithms configured by the *PC* and *TPDA* learning engines respectively, and the solid lines, *DL – PC* and *DL – TPDA*, denote the decomposition learning algorithms equipped with the *PC* and *TPDA* learning techniques (in the *LBK* procedure) respectively.

We show the performance of the decomposition learning algorithm in Fig. [7](#)³. Each data point is the average of 10 runs for different data sets of same size [4. Both the decomposition and the block learning algorithms that are equipped with learning engines have better performance than the *PC* and *TPDA* learning](#)

³ We specify $\epsilon \in [0.40, 0.60]$ concerning the tradeoff between the cluster size and the overlapping set.

⁴ We only count successful runs of the *BL* algorithm when it produces connected local structures.

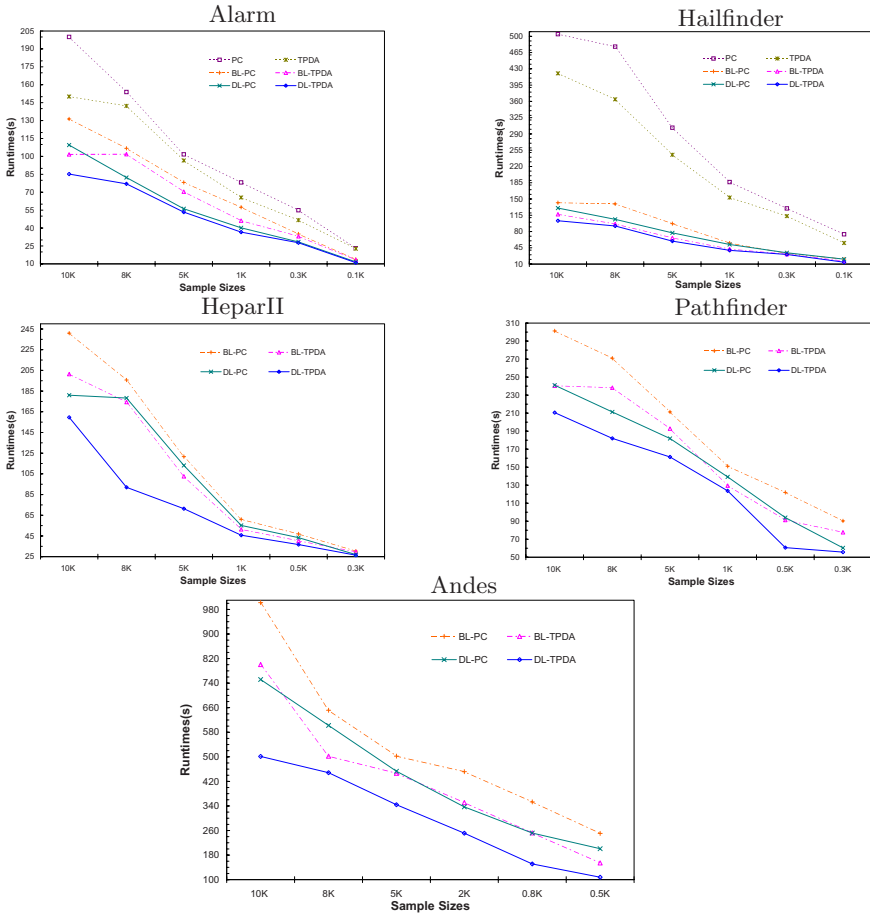


Fig. 8. Runtimes comparison (3GHz, 2GB RAM). The decomposition algorithm is scalable in learning large domains.

algorithms regarding the distance measure. This remains true for a range of data sets. For small domains, such as Alarm and Hailfinder networks, both the decomposition learning algorithm and the block learning algorithm exhibit similar performance of low distance. However, the decomposition algorithm has significantly better results on the rest three large networks, especially for small data sets. We report only the performance of the *BL* and *DL* learning algorithms on the three larger networks since both the *PC* and *TPDA* algorithms fail.

We also observe from Fig. 7 that the decomposition learning algorithm retains a good quality of learned structures when the sample size is noticeably reduced. In addition, the decomposition algorithms have a lower variance than the block learning algorithms. This is due to the *DL* method has a reliable clustering method *SD* comparing with the incomplete block identification in *BL*.

Finally, the run times in Fig. 8 are indicative of the computational savings incurred by using the decomposition learning algorithm. The decomposition algorithm achieves more savings than the block learning algorithm since the latter needs an expensive procedure of learning overlapping structures. Using the decomposition algorithm we were able to learn the three large domains of HeparII, Pathfinder, and Andes, while both the *PC* and *TPDA* algorithms run out of memory. We expect similar results of good performance without intensive computation in real applications.

5 Discussion

The decomposition learning algorithm is able to learn a large Bayesian network structure and shows good performance even when insufficient data are provided. It significantly improves the block learning algorithm on the aspects of robust clustering methods and well-defined combination rules. The modular design provides a way to exploit state-of-the-art of both Bayesian network learning and attribute clustering techniques. In addition, the decomposition learning algorithm offers useful intermediate clusters or Bayesian knots that represent local domain structures and may attract interest into further study. Several issues relevant to the decomposition learning algorithm deserves further study. We are currently investigating an adaptive cluster expansion.

References

1. Pearl, J.: Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann Publishers Inc., San Francisco (1988)
2. Jensen, F.V.: An introduction to Bayesian networks. Springer, New York (1996)
3. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Springer, New York (1993)
4. Heckerman, D.: A tutorial on learning in bayesian networks (1995)
5. Neapolitan, R.E.: Learning Bayesian Networks. Prentice-Hall, Englewood Cliffs (2003)
6. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. *Mach. Learn.* 65(1), 31–78 (2006)
7. Chickering, D.M.: Learning bayesian networks is np-complete. In: Proceedings of AI and Statistics 1995 (1995)
8. Friedman, N., Nachman, I., Pe'er, D.: Learning of bayesian network structure from massive datasets: the sparse candidate algorithm. In: UAI, pp. 206–215 (1999)
9. Segal, E., Pe'er, D., Regev, A.: Learning module networks. In: UAI, pp. 525–534 (2003)
10. Zeng, Y., Poh, K.L.: Block learning bayesian network structure from data. In: Proceedings of the Fourth International Conference on Hybrid Intelligent Systems (HIS 2004), pp. 14–19 (2004)
11. Newman, M.E.J.: The structure and function of complex networks. *SIAM Review* 45(2), 167–256 (1957)
12. Albert, R.Z., Barabasi, A.L.: Statistical mechanics of complex networks. *Modern Physics* (74), 47–97 (2002)

13. Chow, C., Liu, C.: Approximating discrete probability distributions with dependence trees. *IEEE Transaction on Information Theory* (12), 462–467 (1968)
14. Sedgewick, R.: *Algorithms in Java, Part 5 Graph Algorithms*. Addison-Wesley, Reading (2004)
15. Asano, T., Bhattacharya, B., Keil, M., Yao, F.: Clustering algorithms based on minimum and maximum spanning trees. In: *Proceedings of the fourth annual symposium on Computational Geometry*, pp. 252–257 (1988)
16. Grygorash, O., Zhou, Y., Jorgensen, Z.: Minimum spanning tree based clustering algorithms. In: *IEEE International Conference on Tools with AI* (2006)
17. Gallian, J.: Dynamic survey of graph labeling. *Elec. J. Combin.* 14(6) (2007)
18. Cheng, J., Greiner, R., Kelly, J., Bell, D., Liu, W.: Learning bayesian networks from data: An information-theory based approach. *Artificial Intelligence* 137(1), 43–90 (2002)
19. Tsamardinos, I., Aliferis, C., Statnikov, A.: Time and sample efficient discovery of markov blankets and direct causal relations. In: *KDD*, pp. 673–678 (2003)

Learning Classification Rules for Multiple Target Attributes

Bernard Ženko and Sašo Džeroski

Department of Knowledge Technologies, Jožef Stefan Institute
Jamova cesta 39, SI-1000 Ljubljana, Slovenia
Bernard.Zenko@ijs.si, Saso.Dzeroski@ijs.si

Abstract. Among predictive models, ‘if-then’ rule sets are one of the most expressive and human readable model representations. Most of the existing approaches for rule learning focus on predicting a single target attribute/class. In practice, however, we encounter many problems where the task is to predict not one, but several related target attributes. We employ the predictive clustering approach to learn rules for simultaneous prediction of multiple target attributes. We propose a new rule learning algorithm, which (unlike existing rule learning approaches) generalizes to multiple target prediction. We empirically evaluate the new method and show that rule sets for multiple target prediction yield comparable accuracy to the respective collection of single target rule sets. The size of the multiple target rule set, however, is much smaller than the total size of the collection of single target rule sets.

1 Introduction

Traditionally, inductive machine learning focuses on problems where the task is to predict a value of a single target attribute. However, there exist many real life problems where the task is to predict not one, but several related target attributes. Of course, this problem can easily be solved by constructing separate models for each target attribute. If our only goal is to achieve high predictive accuracy, the resulting collection of (single target) models should be sufficient, provided that we have selected a suitable method for single target prediction. On the other hand, if, besides the predictive accuracy, the interpretability of induced models is also important, the collection of single target models is far less understandable than a single model that jointly predicts all target attributes. Therefore, the research on extending machine learning methods that produce interpretable models (such as decision trees and rules) towards multiple target prediction is justified.

One of the possible approaches to multiple target prediction is predictive clustering, which was originally applied to decision trees. The goal of this paper is to adopt the predictive clustering approach in order to design a method for learning rules for multiple target prediction; we call it *predictive clustering rules*.¹ We focus on classification tasks only, though the method can also be extended to regression problems.

¹ An initial solution to the problem of learning predictive clustering rules has been presented in [17]. The algorithm presented here includes an updated search heuristic, a new error weighted covering algorithm, and extended experimental evaluation.

The rest of the paper is organized as follows. Section 2 summarizes predictive clustering. The algorithm for learning predictive clustering rules is presented in Section 3. Section 4 describes the evaluation methodology, and Section 5 presents the experimental results. Last section concludes and gives some directions for further work.

2 Predictive Clustering

The predictive clustering approach [12] builds on ideas from two machine learning areas, predictive modeling and clustering [9]. Predictive modeling is concerned with the construction of models that can be used to predict some object's target property from the description of this object (attribute-value representation is most commonly used for describing objects and their properties). Clustering, on the other hand, is concerned with grouping of objects into classes of similar objects, called clusters; there is no target property to be predicted, and usually no symbolic description of discovered clusters, (though a symbolic descriptions can be added to already constructed clusters as in *conceptual clustering* [13]). Both areas are usually regarded as completely different tasks. However, predictive modeling methods that partition the example space, such as decision trees and rules are also very similar to clustering [10]. They partition the set of examples into subsets in which examples have similar values of the target variable, while clustering produces subsets in which examples have similar values of all descriptive variables. Predictive clustering builds on this similarity. As is common in 'ordinary' clustering, predictive clustering constructs clusters of examples that are similar to each other, but in general taking both the descriptive and the target variables into account. In addition, a predictive model is associated with each cluster which describes the cluster, and, based on the values of the descriptive variables, predicts the values of the target variables. Methods for predictive clustering enable us to construct models for predicting multiple target variables which are normally simpler and more comprehensible than the corresponding collection of models, each predicting a single variable.² So far, this approach has been limited to the tree learning methods. The method described in the next section extends predictive clustering towards methods for learning rules.

3 Predictive Clustering Rules

Predictive clustering rules (PCRs) include ideas from rule learning and clustering. The learning algorithm itself is a generalization of existing rule learning approaches. The rule evaluation function which serves as a search heuristic, however, employs techniques commonly used in clustering. We start with a description of the top level of the algorithm, while specific aspects of the algorithm, such as learning single rules and modification of the learning set between subsequent iterations of the algorithm, are discussed in separate sections.

² Related to multiple target prediction is *Multi-Task learning* [3], where a single model (neural network) is trained for multiple target attributes (learning tasks) with a presumption that a set of hidden submodels will emerge that are used for modeling of all learning tasks.

Table 1. The algorithm for learning predictive clustering rules. **a)** ‘LearnRuleSet’, **b)** ‘FindCandidateRule’, and **c)** ‘ModifyLearningSet’ procedures.

<p>a) LearnRuleSet()</p> <p>Input: learning set E $R = \emptyset$ {rule set} $E_c = E$ {current learning set}</p> <p>repeat</p> <p style="padding-left: 20px;">$r_i = \text{FindCandidateRule}(E_c)$ $R = R \cup \{r_i\}$ $E_c = \text{ModifyLearningSet}(E_c, r_i)$</p> <p>until $((r_i = \emptyset) \text{ or } (\ E_c\ = 0))$ $R = R \cup \text{DefaultRule}(E)$</p> <p>return R</p>	<p>b) FindCandidateRule()</p> <p>Input: current learning set E_c $c_{last} = \text{“true”}$ $C = C_{best} = \{c_{last}\}$</p> <p>while $(C \neq \emptyset)$ do</p> <p style="padding-left: 20px;">$C_{new} = \emptyset$</p> <p style="padding-left: 20px;">for all $(c \in C)$ do</p> <p style="padding-left: 40px;">for all $(t \in T_p \wedge t \notin c)$ do</p> <p style="padding-left: 60px;">$c_{new} = c \wedge t$</p> <p style="padding-left: 60px;">if $(h(c_{new}) > h(c_{last}))$ then</p> <p style="padding-left: 80px;">$C_{new} = C_{new} \cup \{c_{new}\}$ $C_{best} = C_{best} \cup \{c_{new}\}$</p> <p style="padding-left: 60px;">if $(C_{new} > b_w)$ then</p> <p style="padding-left: 80px;">$C_{new} = C_{new} \setminus \arg \min_{c' \in C_{new}} h(c')$</p> <p style="padding-left: 60px;">if $(C_{best} > b_w)$ then</p> <p style="padding-left: 80px;">$C_{best} = C_{best} \setminus \arg \min_{c' \in C_{best}} h(c')$</p> <p style="padding-left: 60px;">$c_{last} = \arg \min_{c' \in C_{best}} h(c')$</p> <p style="padding-left: 20px;">$C = C_{new}$</p> <p style="padding-left: 20px;">$c_{best} = \arg \max_{c' \in C_{best}} h(c')$</p> <p>return (C_{best}, p_{best})</p>
---	--

c) ModifyLearningSet()

Input: current learning set E_c , newly added rule r_i

case $(M_{mod} = \text{“Std-Covering”})$ **do**

for all $(e_i \in E_c)$ **do**

if $(r_i \text{ covers } e_i)$ **then**

$w_{ei} = 0$

return E_c

case $(M_{mod} = \text{“Err-Weight-Covering”})$ **do**

for all $(e_i \in E_c)$ **do**

if $(r_i \text{ covers } e_i)$ **then**

$w_{ei} = w_{ei} \cdot g(e_i, r_i)$

if $(w_{ei} < \epsilon)$ **then**

$w_{ei} = 0$

return E_c

3.1 Learning Algorithm

Most of existing approaches to rule learning are based on the *covering algorithm* [12]. Its main problem, however, is that it was originally designed for two-class (binary) classification problem domains. In addition, the rule sets produced by the original covering algorithm are by nature ordered, unless rules for only one class value are constructed. Our algorithm is based on the CN2 method [54], which uses a version of the covering algorithm that can learn ordered or unordered rules, and is also applicable to (single target) multi-class problems.

The algorithm for learning predictive clustering rules is presented in Table 1. Top level procedure ‘LearnRuleSet’ (Table 1.a) starts with an empty rule set R and a set of learning examples E . In each iteration we learn a candidate rule r_i and add it to the rule set. Next, we modify the current learning set E_c and, unless some stopping criterion is met, repeat the loop. There are two stopping criteria; we stop adding rules if the ‘FindCandidateRule’ procedure could not find any non-empty rule, and when the $\|E_c\|$ becomes zero ($\|E_c\|$ is the number of examples with non-zero weights). Before the

learning procedure is finished, we add the default rule. The default rule is a rule with an empty condition and is used for examples that are not covered by any other rule. Its prediction part is a cluster prototype of the complete learning set E .

The interpretation of PCRs is the same as that of CN2 rules: ordered rules are scanned and the first one that covers the example is used; predictions of all unordered rules that cover the example are combined into the final prediction via weighted voting, where the weights are equal to the number of covered examples on the training data.

3.2 Learning Single Rule

The ‘FindCandidateRule’ procedure is given in Table 1b, and is a general-to-specific beam search algorithm, which is very similar to the one implemented in the CN2. The input to the procedure is the learning set of examples E_c . The width of the beam b_w determines the number of partial rules maintained during the search. A set of b_w best rules (or actually conditions) found so far as evaluated by the heuristic function h is denoted as C_{best} . We start with the most general condition (“true”) that is satisfied by all examples in the learning set E_c . Now we begin specialization of all conditions in the current set of conditions C by conjunctively adding an extra test. Here we consider all possible tests (T_p) that are not already in the condition that we are specializing. Here we only consider conditions that cover at least a predefined minimal number of examples μ . Every specialization is evaluated using the heuristic function h . If any specialization is better than the worst condition in the set C_{best} , we add it to this set and to set C_{new} . We remove the worst conditions if the sizes of these sets increase over their predefined maximum sizes. When all specializations of the current set of conditions C are examined, the set C becomes set C_{new} , and the search is continued until no better specializations can be found. At the end, the best condition from the set C_{best} is coupled with the prototype of target attributes of examples that it covers (p_{best}), and returned as a candidate rule.

Search Heuristic. The crucial part of the algorithm is the search heuristic h . The heuristic is used for the evaluation of rules under construction and basically leads the search procedure towards rules of the desired quality. Therefore, the heuristic should reflect the qualities we expect from each individual rule in the rule set. Typically, we want the rules to be accurate and, at the same time, general, i.e., we want the rules to cover as many examples as possible. More generalization means that the rule covers more examples and in the end, it also means that the final rule set will have fewer rules and will be more comprehensible. Unfortunately, more generalization most often also means larger error in the model, and a compromise between the two must be found.

Normally, the accuracy measures are tailored to single target prediction, while for predictive clustering rules we need a measure that also works for multiple target prediction. We define such a measure, we call it *dispersion*, as follows. Let E' be the set of N examples that are covered by a specific rule, and each example \mathbf{e}_i is represented as a vector of K attribute values x_{ji} , where x_{ji} stands for the value of the attribute a_j of the example \mathbf{e}_i . The dispersion of a set of examples E' is an average of the dispersions along each attribute

$$\text{disp}(E') = \frac{1}{K} \sum_{j=1}^K \text{disp}(E', a_j). \quad (1)$$

Here we take into account only the target attributes, although in principle, we could include also the non-target attributes [17].

The definition of dispersion along a single nominal attribute is the average distance of a single example from a set to the prototype of this set. Let the attribute a_j have L possible values with labels l_1 to l_L . The prototype of a set of examples E' of an attribute a_j is a vector of relative frequencies f_k of possible values within the set

$$\mathbf{p}_{E'} = \mathbf{p}(E'; a_j) = [f_1, f_2, \dots, f_L]; \quad f_k = \frac{n_k}{N}, \quad (2)$$

where n_k stands for the number of examples in the set E' whose value of attribute a_j equals l_k . Accordingly, (the prototype of) a single example \mathbf{e}_i with the value of attribute a_j equal to l_k is

$$\mathbf{p}_{\mathbf{e}_i} = [f'_1, f'_2, \dots, f'_L]; \quad f'_k = \begin{cases} 1, & \text{if } x_{ji} = l_k, \\ 0, & \text{otherwise.} \end{cases} \quad (3)$$

The distance between the two prototypes can be measured using any of the distance measures defined on vectors; we have decided to use the *Manhattan distance*. Now the distance between an example \mathbf{e}_i with the value of attribute a_j equal to l_k (i.e., the prototype $\mathbf{p}_{\mathbf{e}_i}$) and prototype of the entire set E' is

$$d(\mathbf{p}_{\mathbf{e}_i}, \mathbf{p}_{E'}) = |1 - f_k| + \sum_{\substack{m=1 \\ m \neq k}}^L |f_m| = 2(1 - f_k); \quad (4)$$

where we have taken into account that $f_k < 1$ and $\sum f_m = 1$. Finally, the dispersion of the set of examples E' along the nominal attribute a_j is the normalized average distance

$$\text{disp}(E', a_j) = \frac{1}{2N} \frac{L}{L-1} \sum_{i=1}^N d(\mathbf{p}_{\mathbf{e}_i}, \mathbf{p}_{E'}). \quad (5)$$

The normalization factor normalizes the value of dispersion to the $[0, 1]$ interval which is necessary, if we want the dispersions between different attributes to be comparable.

The rule's generality is typically measured by its *coverage*, which is defined as the proportion of covered examples, i.e., the number of examples covered by a rule divided by the number of all examples. This definition assumes that all examples are equally important, i.e., they all have equal weight. As we will see later, sometimes it is useful to introduce example weights that are not uniform. Each example \mathbf{e}_i then has an associated weight w_{ei} . The relative coverage of rule r in this case is simply the sum of weights of the examples covered by r divided by the sum of weights of all examples

$$\text{cov}(r; E, \mathbf{w}) = \frac{\sum_{\mathbf{e}_i \in E'} w_i}{\sum_{\mathbf{e}_i \in E} w_{ei}}. \quad (6)$$

Now, we have to combine the two measures into a single heuristic function. Analogously to the WRAcc heuristic [11], we do this as follows. Let c be the condition of rule r that we are evaluating, and E be the set of all learning examples. E_r is the subset of E with examples that satisfy condition c (i.e., are covered by rule r). \mathbf{w}_e is the example

weight vector. By means of example weights we can give preference to selected examples, which should more likely lead to the construction of rules covering these examples (more on this later). The heuristic function is

$$h^*(c) = [d_{def} - \text{disp}(E_r)] \cdot \text{cov}(r; E, \mathbf{w}_e)^\alpha. \quad (7)$$

The parameter α enables us to put more (or less) emphasis on coverage w.r.t. to dispersion; by default (like in WRAcc) it is set to 1. d_{def} is the *default dispersion*, i.e., the dispersion of the entire learning set E , and the first factor of Equation 7 can be regarded as the relative dispersion loss. Rules with larger heuristic function values are better.

3.3 Modifying the Learning Set

Within the main loop of the ‘LearnRuleSet’, the current learning set E_c must be modified, otherwise the ‘FindCandidateRule’ procedure would continuously find the same rule. Learning set modification is done by the ‘ModifyLearningSet’ procedure presented in Table 1c.

The most common approach to modifying the learning set is the covering algorithm [12]. The idea is that we put more emphasis on the learning examples that have not yet been adequately covered. This should force the ‘FindCandidateRules’ procedure to focus on these examples and find rules to describe them. In the original covering algorithm ($M_{mod} = \text{“Std-Covering”}$), examples that are already covered by a rule are removed from the current learning set. Rule learning in the next iteration will therefore focus only on examples that have not yet been covered. This approach is used by the CN2 algorithm [54] for the induction of ordered rules, and ordered PCRs are also induced in this manner.

The weighted covering algorithm [8], on the other hand, assigns a weight to each learning example. Instead of removing the covered example completely, weighted covering only decreases its weight. It does this, however, only for examples that have been correctly classified by the newly added rule. The notion of ‘correctly classified example’ unfortunately only makes sense for single target classification problems. To overcome this limitation, we develop a more general covering scheme, which we call *error weighted covering*, that is applicable to single and multiple target prediction problems ($M_{mod} = \text{“Err-Weight-Covering”}$). Error weighted covering is similar to ‘ordinary’ weighted covering, except that the amount by which example’s weight is reduced is proportional to the error the newly added rule makes when predicting the example’s target attributes’ values. The exact weighting scheme is as follows.

Let every learning example \mathbf{e}_i have an assigned weight w_{ei} . At the beginning, the weights of all examples are set to one. Then, whenever a new rule r is added to the rule set, the weight of each covered example \mathbf{e}_i is multiplied by the value of $g(\mathbf{e}_i, r)$, which is defined as

$$g(\mathbf{e}_i, r) = 1 + (\zeta - 1)k(\mathbf{e}_i, r), \quad (8)$$

where $k(\mathbf{e}_i, r)$ is the proportion of correctly classified target attributes of example \mathbf{e}_i by rule r

$$k(\mathbf{e}_i, r) = \frac{\text{nb. corr. pred. tar. atts of } \mathbf{e}_i \text{ by } r}{\text{nb. all tar. atts}}, \quad (9)$$

and ζ is the *covering weight parameter*, which enables us, together with the *covering weight threshold parameter* ϵ , to control the pace of removing covered examples from the current learning set. It should take values between 0 and 1. Setting ζ to 0 means that examples, whose target attributes are correctly predicted by rule r , are immediately removed from the current learning set, i.e., their weights are set to zero. The parameter ϵ defines the threshold under which the example weights are considered to be too small to be still included in the learning set; when the example weight falls below this value, it is set to zero.

4 Experimental Setup

In the experiments, we investigate two issues. First, we compare the performance of predictive clustering rules (PCRs) to some existing rule learning methods for single target classification in order to show that PCRs are comparable to existing methods on this type of problems, and can be used as a baseline in the second part of the evaluation. For comparison we selected the CN2 rule learner [54] and a modification of CN2, the CN2-WRAcc [15], because our approach is a generalization of these algorithms. Additionally, we compare PCRs to Ripper [6] which is a more advanced rule learner; we used the JRip implementation from the Weka data mining suite [16] which only learns ordered rules.

Second, we compare the PCRs for single target prediction to PCRs for multiple target prediction. The main benefit of multiple target prediction is that a collection of models (rule sets) each predicting one target attribute can be replaced by just one model that predicts all target attributes at once. The task of the second part of experimental evaluation is to investigate this issue.

4.1 Data Sets

In order to evaluate the performance of PCRs, we perform experiments on single target and on multiple target problems. For single target problems, we have selected a collection of 15 data sets from the *UCI Machine Learning Repository* [14] which are widely used in various comparative studies.

Multiple target classification is a relatively new machine learning task and consequently there are few publicly available data sets. Nevertheless, some data sets from the *UCI Repository* can also be regarded as multiple target problems (BRIDGES, MONKS, SOLAR-FLARE, and THYROID-0387). In addition, we use the following five data sets.

The EDM is a data set on electrical discharge machining with 154 examples, 16 descriptive attributes and two target attributes. The MEDIANA data set consists of 7953 questionnaires on the Slovene media space, has 79 descriptive attributes and 5 target attributes. The SIGMEA-REAL is a data set on a field study of a genetically modified oilseed rape. It comprises 817 examples, 6 descriptive, and 2 target attributes. The SIGMEA-SIM data set is also concerned with genetically modified oilseed rape, however, the data are produced by a simulation model. The data consists of 10368 examples with 11 descriptive and 2 target attributes. The WATER-QUALITY data set comprises biological and chemical data that were collected through regular monitoring of rivers in Slovenia. The data consists of 1060 examples with 16 descriptive and 14 target attributes.

4.2 Evaluation Methodology

When evaluating the newly developed method, we are interested in the predictive error of the learned rule sets and their size, i.e., the number of rules within the rule sets. The CN2 and CN2-WRAcc as well as PCR algorithms can induce ordered or unordered rules, therefore we perform experiments for both. JRip can only learn ordered rules. All error rates are estimated using 10-fold cross-validation. The folds for a specific data set are the same for all experiments. As recommended by [7], significance of the observed differences in error rates and rule set sizes of two algorithms was tested with the *Wilcoxon signed-rank* test.

Unless otherwise noted, all algorithm parameters were set to their default values. CN2 can use significance testing for rule pruning, while there is no need for significance testing in CN2-WRAcc, since the number of induced rules by this algorithm is already much smaller. We use the *p-value* of 0.99 for significance testing in the CN2 algorithm.

The default parameter values for the PCR algorithm are as follows: beam width $b_w=10$, minimal number of examples $\mu=2$, coverage heuristic weight $\alpha=1$, covering weight $\zeta=0$, and covering weight threshold $\epsilon=0.1$. These are set so as to emulate the CN2 and CN2-WRAcc algorithms as closely as possible and were not tuned in any way. Ordered rules were induced with the learning set modifying method (M_{mod}) set to “*Std-Covering*”, while for unordered rules it was set to “*Err-Weight-Covering*”.

The comparison of PCRs used for multiple target prediction and PCRs used for single target prediction is performed as follows. For each data set, we have learned one multiple target PCR model and compared it to a collection of single target PCR models. This collection consists of the same number of models as is the number of target attributes in a given domain. The sizes of the single target PCR rule sets for each target attribute are summed and compared to the size of the multiple target PCR rule set. The overall significance of differences is again estimated using the *Wilcoxon signed-rank* test; each target attribute of each data set corresponds to one data point.

5 Experimental Results

5.1 Comparison to Existing Methods

First, we present the results of the comparison of predictive clustering rules (PCRs) to the CN2, CN2-WRAcc, and JRip methods. Table 2a gives the significances of differences for pairs of algorithms for ordered rules and unordered rules. Except for the JRip, we also compared ordered vs. unordered rules. Due to space limits, we have left out the table with complete results for each data set.

For ordered rules, we can see that there are no significant differences between the CN2, CN2-WRAcc, and PCR algorithms in terms of error, but rule sets induced by CN2-WRAcc have a significantly smaller number of rules. JRip rules are better in terms of error than ordered PCRs, but the difference is below the significance threshold. Next, if we compare unordered rules, we see that PCRs have a significantly smaller error than the two CN2 algorithms. However, the PCR rule sets are much larger than the CN2-WRAcc rule sets. There is no significant difference between (ordered) JRip

Table 2. Significances (p-values) of differences in error rates and rule set sizes for the pairs of algorithms: *CN2*, *CN2-WRAcc* (*cn2w*), *JRip*, and *PCR* for ordered (OR) and unordered (UN) rules. The sign < (>) right of a p-value means that the first (second) algorithm tends to induce rule sets with smaller error rate or size. Significant differences are typeset in bold. **a)** Single target classification, **b)** Single target vs. multiple target classification.

a)			b)		
COMPARED ALGORITHMS	ERROR	SIZE	COMPARED ALGORITHMS	ERROR	SIZE
	P-VALUE	P-VALUE	SINGLE : MULTIPLE	P-VALUE	P-VALUE
CN2 OR : CN2W OR	0.188	< <0.001 >	PCR OR	0.066	< <0.001 >
CN2 OR : PCR OR	0.978	< 0.151 >	PCR UN	0.067	> <0.001 >
CN2W OR : PCR OR	0.359	> 0.003 <			
JRIP OR : PCR OR	0.073	< 0.934 >			
CN2 UN : CN2W UN	0.762	< <0.001 >			
CN2 UN : PCR UN	0.002	> 0.804 <			
CN2W UN : PCR UN	0.003	> <0.001 <			
JRIP OR : PCR UN	0.847	> 0.007 <			
CN2 OR : CN2 UN	0.144	< 0.359 <			
CN2W OR : CN2W UN	0.524	< 0.804 >			
PCR OR : PCR UN	0.018	> <0.001 <			

rules and unordered PCRs in terms of error, but JRip tends to produce significantly smaller rule sets than the latter. Finally, if we compare ordered and unordered rules induced by each algorithm, the only significant difference is in the case of PCRs; unordered PCRs have a significantly smaller error, but this accuracy comes at a price, since their size is much larger. From these results, we can conclude that the performance of PCRs on single target problems is comparable to the performance of the CN2 and CN2-WRAcc algorithms for ordered rules, and better for unordered rules. In terms of error, ordered PCRs are somewhat worse than JRip, while unordered PCRs are comparable to JRip.

5.2 Comparison of Single to Multiple Target Prediction

The significances of differences between single target and multiple target PCRs are given in Table 2b, while error rates and rule set sizes are presented in Table 3.

From the Table 2b we can conclude that ordered multiple target prediction models tend to be less accurate than the single target prediction models. In the case of unordered rules, however, the situation is reversed: multiple target prediction models are better than the single target prediction models. In both cases the difference is almost significant (p -value ≈ 0.07). The difference in the rule set sizes, however, is very significant; the size of single target rule sets is roughly twofold in the case of unordered rules, and more than threefold in the case of ordered rules. These results suggest that the multiple target PCRs indeed outperform single target PCRs in terms of rule set size, while the accuracy of both types of models is comparable. In addition, multiple target prediction setting somewhat improves the accuracy of unordered rule sets.

Table 3. Comparison of *error rates of ordered (OR) and unordered (UN) PCRs used for single target and multiple target classification.* For each data set, the average error rate over all target attributes is given first, and then for each target attribute separately. Sizes of single target prediction rule sets and summed and compared to multiple target prediction rule set. In each row, the smallest error rate of ordered and unordered rules is typeset in bold. The final row (next page) gives the average error rate over all target attributes of all data sets and the average rule set size over all data sets.

DATA SET TAR. ATT.	PCR OR SINGLE		PCR OR MULTIPLE		PCR UN SINGLE		PCR UN MULTIPLE	
	% ERROR	# SIZE	% ERROR	# SIZE	% ERROR	# SIZE	% ERROR	# SIZE
BRIDGES	35.0	34	40.5	7	37.3	36	32.2	12
T-OR-D	19.4 ±14.1	4	24.7 ±0.0		19.4 ±14.1	4	10.6 ±8.7	
MATERIAL	21.6 ±13.5	6	20.0 ±10.1		26.5 ±17.8	6	18.8 ±10.4	
SPAN	31.8 ±14.9	6	43.5 ±11.3		35.2 ±13.3	6	40.0 ±11.0	
REL-L	47.5 ±21.5	7	44.7 ±0.0		46.5 ±20.6	8	35.3 ±15.7	
TYPE	54.9 ±17.2	11	69.4 ±0.0		58.8 ±12.7	12	56.5 ±0.0	
EDM	24.7	17	25.0	9	28.2	16	29.2	11
D-FLOW	13.6 ±15.5	7	11.7 ±8.1		16.2 ±15.2	7	12.3 ±9.0	
D-GAP	35.7 ±11.8	10	38.3 ±8.2		40.3 ±0.0	9	46.1 ±13.4	
MEDIANA	18.2	1297	17.2	271	20.1	1505	16.6	685
READ-DELO	23.1 ±0.9	306	22.0 ±1.5		24.0 ±1.3	353	21.7 ±1.2	
READ-DNEVNIK	21.9 ±1.2	436	16.6 ±1.4		20.4 ±1.7	493	15.4 ±0.9	
READ-EKIPA	9.2 ±0.9	296	7.1 ±0.8		7.4 ±0.9	362	6.3 ±0.8	
READ-SL-NOV	26.3 ±1.4	100	28.8 ±1.1		38.0 ±4.6	100	29.2 ±0.9	
READ-VECER	10.3 ±1.7	159	11.6 ±0.9		10.5 ±1.1	197	10.4 ±1.0	
MONKS	3.3	39	21.7	4	17.5	40	23.5	10
MONK-1	0.0 ±0.0	7	30.1 ±9.1		11.1 ±4.9	7	17.6 ±7.3	
MONK-2	10.0 ±6.4	28	33.1 ±8.0		33.3 ±13.1	29	35.9 ±5.9	
MONK-3	0.0 ±0.0	4	1.9 ±3.0		8.1 ±7.2	4	17.1 ±5.4	
SIGMEA-REAL	24.8	76	24.9	38	24.5	91	24.9	72
MFO	26.1 ±5.4	52	24.5 ±5.2		26.2 ±5.8	60	25.1 ±4.6	
MSO	23.5 ±6.1	24	25.3 ±3.8		22.8 ±5.3	31	24.6 ±3.3	
SIGMEA-SIM	0.7	14	2.1	3	1.2	15	2.1	4
DISP-RATE	1.4 ±0.4	12	4.3 ±0.7		2.4 ±0.7	13	4.3 ±0.7	
DISP-SEEDS	0.0 ±0.0	2	0.0 ±0.0		0.0 ±0.0	2	0.0 ±0.0	
SOLAR-FLARE	11.1	58	11.0	23	13.1	79	10.4	39
C-CLASS	15.8 ±7.6	25	15.2 ±6.7		18.3 ±8.2	36	13.6 ±6.9	
M-CLASS	13.0 ±3.7	19	14.6 ±4.7		15.8 ±4.0	27	14.9 ±4.6	
X-CLASS	4.6 ±4.1	14	3.1 ±3.2		5.3 ±5.3	16	2.8 ±3.4	
THYROID-0387	1.8	666	2.4	497	2.1	727	2.5	560
HYPER-THYRO	2.0 ±0.5	107	2.5 ±0.6		1.7 ±0.5	115	2.5 ±0.5	
HYPO-THYRO	0.9 ±0.4	53	3.1 ±0.8		2.4 ±0.7	40	3.7 ±0.5	
BIND-PROT	2.9 ±0.8	135	3.2 ±0.8		3.0 ±0.6	157	3.4 ±0.6	
GEN-HEALTH	2.6 ±0.8	125	3.6 ±0.6		3.0 ±0.7	134	2.7 ±0.9	
REPL-THEORY	2.1 ±0.5	129	2.1 ±0.4		2.2 ±0.7	148	3.0 ±0.8	
ANTITHYRO-TR	0.3 ±0.2	24	0.3 ±0.2		0.4 ±0.2	24	0.4 ±0.2	
DISC-RESULTS	1.6 ±0.3	93	2.0 ±0.5		2.1 ±0.6	109	2.0 ±0.6	

Continued on the next page.

Table 3. (continued)

DATA SET TAR. ATT.	PCR OR SINGLE		PCR OR MULTIPLE		PCR UN SINGLE		PCR UN MULTIPLE	
	% ERROR	# SIZE	% ERROR	# SIZE	% ERROR	# SIZE	% ERROR	# SIZE
WATER-QUALITY	32.1	736	33.3	89	33.9	788	31.8	153
CLAD-SP	38.5 ± 3.7	31	39.5 ± 5.6		39.9 ± 4.9	28	40.4 ± 4.9	
GONG-INC	34.4 ± 4.2	64	29.5 ± 3.6		39.2 ± 6.6	77	28.4 ± 3.2	
OEDO-SP	29.8 ± 3.6	62	29.7 ± 4.5		30.9 ± 4.7	78	29.9 ± 5.1	
TIGE-TEN	25.1 ± 2.6	80	23.2 ± 4.6		23.9 ± 4.8	83	20.8 ± 2.4	
MELO-VAR	34.2 ± 4.0	30	41.7 ± 4.8		38.9 ± 3.3	27	41.4 ± 3.7	
NITZ-PAL	29.6 ± 3.0	23	31.3 ± 2.3		30.0 ± 4.8	25	30.6 ± 4.3	
AUDO-CHA	30.5 ± 5.0	88	29.3 ± 5.0		30.8 ± 5.9	90	24.2 ± 5.2	
ERPO-OCT	31.7 ± 3.7	75	29.0 ± 2.6		32.7 ± 4.3	79	26.5 ± 3.3	
GAMM-FOSS	32.3 ± 3.8	27	38.1 ± 4.5		33.0 ± 5.9	23	37.8 ± 3.7	
BAET-RHOD	32.0 ± 4.8	49	31.8 ± 3.1		33.5 ± 5.4	57	32.5 ± 2.4	
HYDRO-SP	34.9 ± 4.5	29	39.1 ± 4.6		39.1 ± 4.2	33	38.2 ± 4.9	
RHYA-SP	29.1 ± 4.4	59	36.1 ± 5.5		32.8 ± 4.2	69	30.8 ± 6.8	
SIMU-SP	37.6 ± 4.5	59	38.5 ± 5.5		39.3 ± 4.6	55	37.3 ± 4.7	
TUBI-SP	29.2 ± 3.8	60	28.8 ± 4.0		31.0 ± 3.9	64	27.1 ± 3.1	
AVERAGE	20.3	326.3	22.6	104.6	22.7	366.3	21.4	171.8

6 Conclusions and Further Work

A new method for learning rules for multiple target classification, called predictive clustering rules, is proposed in this paper. The method combines ideas from supervised and unsupervised learning and extends the predictive clustering approach to methods for rule learning. In addition, it generalizes rule learning and clustering.

The newly developed method is empirically evaluated in terms of error and rule set size on several single and multiple target classification problems. First, the method is compared to some existing rule learning methods (CN2, CN2-WRAcc, and JRip) on single target problems. These results suggest that PCRs' performance on single target classification problems is good, and they can be used as a baseline in the next part of the evaluation.

The comparison of multiple target prediction PCRs to the corresponding collection of single target prediction PCRs on multiple target classification problems shows that in the case of ordered rules, the single target prediction models are better, while in case of unordered rules, the multiple target prediction PCRs are better. The differences in both cases are almost (but not quite) significant. The difference in the rule set sizes, on the other hand, is very significant. Multiple target prediction ordered and unordered rule sets are much smaller than the corresponding single target prediction rule sets.

The new method therefore compares favorably to existing methods on single target problems, while multiple target models (on multiple target problems) offer comparable performance and drastically lower complexity than the corresponding collections of single target models.

Let us conclude with some guidelines for further work. We have only discussed classification problems in this paper. By defining the dispersion measure used in the search

heuristic for numeric attributes, it should be possible to extend the presented algorithm towards regression problems also. Since there are not many methods for learning regression rules, we see this as a worthwhile direction for further research. In addition, there exist several newer methods, e.g., Ripper [6]; incorporating the ideas from these methods into predictive clustering rules could lead to improved performance.

References

1. Blockeel, H.: Top-down Induction of First Order Logical Decision Trees. PhD thesis, Katholieke Universiteit Leuven, Department of Computer Science, Leuven, Belgium (1998)
2. Blockeel, H., De Raedt, L., Ramon, J.: Top-down induction of clustering trees. In: Proceedings of the Fifteenth International Conference on Machine Learning, July 1998, pp. 55–63. Morgan Kaufmann, San Francisco (1998)
3. Caruana, R.: Multitask learning. *Machine Learning* 28(1), 41–75 (1997)
4. Clark, P., Boswell, R.: Rule induction with CN2: Some recent improvements. In: Proceedings of the Fifth European Working Session on Learning, pp. 151–163. Springer, Berlin (1991)
5. Clark, P., Niblett, T.: The CN2 induction algorithm. *Machine Learning* 3(4), 261–283 (1989)
6. Cohen, W.W.: Fast effective rule induction. In: Proceedings of the Twelfth International Conference on Machine Learning, pp. 115–123. Morgan Kaufmann, San Francisco (1995)
7. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7, 1–30 (2006)
8. Gamberger, D., Lavrač, N.: Expert guided subgroup discovery: Methodology and application. *Journal of Artificial Intelligence Research* 17, 501–527 (2002)
9. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley & Sons, New York (1990)
10. Langley, P.: Elements of Machine Learning. Morgan Kaufmann, San Francisco (1996)
11. Lavrač, N., Flach, P., Zupan, B.: Rule evaluation measures: A unifying view. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 174–185. Springer, Heidelberg (1999)
12. Michalski, R.S.: On the quasi-minimal solution of the general covering problem. In: Proceedings of the Fifth International Symposium on Information Processing (FCIP 1969), Bled, Yugoslavia, Switching Circuits, vol. A3, pp. 125–128 (1969)
13. Michalski, R.S.: Knowledge acquisition through conceptual clustering: A theoretical framework and algorithm for partitioning data into conjunctive concepts. *International Journal of Policy Analysis and Information Systems* 4, 219–243 (1980)
14. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: UCI repository of machine learning databases (1998)
15. Todorovski, L., Flach, P., Lavrač, N.: Predictive Performance of Weighted Relative Accuracy. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 255–264. Springer, Heidelberg (2000)
16. Witten, I.H., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques, 2nd edn. Morgan Kaufmann, San Francisco (2005)
17. Ženko, B., Džeroski, S., Struyf, J.: Learning Predictive Clustering Rules. In: Bonchi, F., Boulicaut, J.-F. (eds.) KDID 2005. LNCS, vol. 3933, pp. 234–250. Springer, Heidelberg (2006)

A Mixture Model for Expert Finding*

Jing Zhang, Jie Tang, Liu Liu, and Juanzi Li

Department of Computer and Technology, Tsinghua University
1-308, FIT Building, Tsinghua University, Beijing, China, 100084
{zhangjing, tangjie, ljz}@keg.cs.tsinghua.edu.cn

Abstract. This paper addresses the issue of identifying persons with expertise knowledge on a given topic. Traditional methods usually estimate the relevance between the query and the support documents of candidate experts using, for example, a language model. However, the language model lacks the ability of identifying semantic knowledge, thus results in some *right* experts cannot be found due to not occurrence of the query terms in the support documents. In this paper, we propose a mixture model based on Probabilistic Latent Semantic Analysis (PLSA) to estimate a hidden semantic theme layer between the terms and the support documents. The hidden themes are used to capture the semantic relevance between the query and the experts. We evaluate our mixture model in a real-world system, ArnetMiner¹. Experimental results indicate that the proposed model outperforms the language models.

1 Introduction

Expert finding, aiming at answering the question: “Who are experts on topic X?”, is becoming one of the biggest challenges for information management [15]. Recent years, expert finding has attracted much attention due to the rapid flourish of the Web 2.0 applications and the advancement of information retrieval technologies from the traditional document-level to the object-level [20]. Many challenging questions arise, for example, How to find the most appropriate collaborators for a project? How to find the important scientists on a research topic? How to find an expertise consultant?

Much research work has been done to deal with the challenges. For example, [2][21] propose using conventional language models for finding experts from an enterprise corpora or a domain-specific document collection. TREC has provided a common platform for researchers to empirically assess methods and techniques devised for expert finding. The task can be described as follows: given a set of documents, a list of candidate names, and a set of topics, the goal then is to find experts from the list of candidate names for each of these topics.

Previously, the language model like method or information retrieval based method is usually used for finding experts for a topic. A relevance score is calculated by

* The work is supported by the National Natural Science Foundation of China (90604025, 60703059), Chinese National Key Foundation Research and Development Plan (2007CB310803), and Chinese Young Faculty Research Funding (20070003093).

¹ <http://www.arnetminer.org>

combining relevance scores between the query and different support documents related to each expert candidate. Based on the combination methods, the approach can be again classified into two categories: ‘*composite*’ and ‘*hybrid*’. *Composite* combines the scores of different documents by aggregation and *hybrid* integrates the scores of different support documents into a single formula (cf. Section 3 for details of the two methods). However, preliminary experiments show that simply applying these two categories of models on the task of expert finding does not achieve satisfactory results. In traditional IR models, documents are taken as the retrieval units and the content of documents are considered reliable. However, the reliability assumption is no longer valid in the expert finding context. This is because:

- (1) Composite model (cf. Section 3.2.1) suffers from the limitation that all the query terms should occur in each support document.
- (2) Hybrid model (cf. Section 3.2.2) is a bit more flexible. However, it still requires that all the query terms should occur in the support documents.

The language model-based methods are lexical-level and suffer from lacking semantics. A question, thus, arises: “Can we search for experts in a semantic-level?”.

In this paper, we focus on the above problems. We propose a mixture model based on Probabilistic Latent Semantic Analysis (PLSA) [16] for the expert finding task. In this model, we do not model the relevance between a query and a document directly. Instead, we propose to use a hidden theme layer to model the semantic relations between the query and the support documents of candidate experts. In this way, an expert whose support documents associated with the same themes as that of a query can be ranked higher, although they may not contain the query terms. We evaluated the proposed approach in ArnetMiner system. We compared our approach with the traditional language models for expert finding. We also carried out the comparison with several existing systems. Experimental results show that our proposed approach performs better than the baseline methods and also outperforms the existing systems.

Our contributions in this paper include: (a) formalization of the expert finding problem in a semantic-level, (b) proposal of a mixture model to the problem based on Probabilistic Latent Semantic Analysis (PLSA), and (c) empirical verification of the effectiveness of the proposed approach. To the best of our knowledge, no previous work has been done on a semantic-level model for expert finding.

The rest of the paper is organized as follows. In Section 2, we formalize the task of expert finding. In Section 3, we briefly introduce the language model and propose our mixture model for expert finding. In Section 4, we give the experimental results and in Section 5, we introduce the related work. We conclude the paper in Section 6.

2 Expert Finding Description

We denote a candidate expert as e and a query as q . A general process of expert finding is to estimate the probability of a person being an expert for a given query, i.e., $P(e|q)$, and then return the experts with the highest probabilities on the top.

Based on the Bayes rule, we can obtain the following formula:

$$P(e|q) = \frac{P(q|e)P(e)}{P(q)} \xrightarrow{P(q) \text{ is uniform}} P(e|q) \propto P(q|e)P(e) \quad (1)$$

where $P(q|e)$ is the generating probability of a query q given an expert e . $P(e)$ and $P(q)$ respectively denote the prior probability of an expert e and a query q . $P(q)$ is usually viewed as uniform and thus can be ignored. The probability $P(e)$ reflects the query-independent expertise. A variety of techniques can be used to compute $P(e)$, for example, we can simply use the number of one’s publications to estimate the probability; more complicated, we can calculate it by using a propagation scheme like the state-of-the-art PageRank algorithm. Also, some work assumes it uniformly and only focuses on estimating the probability $P(q|e)$ using language models [2] [21].

Figure 1 shows an example of expert finding. The left part of the figure gives three queries: “semantic web”, “machine learning”, and “natural language processing” and the right part of the figure shows the found experts for each query.

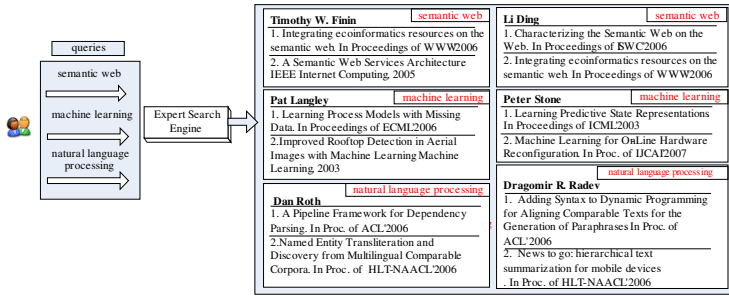


Fig. 1. An example of expert finding

3 Models for Expert Finding

In this section, we will first briefly introduce the language model and then describe several existing language models for expert finding, namely a hybrid model and a composite model. Finally, we propose a mixture model for finding experts.

3.1 Language Models for Document Retrieval

In document retrieval, language model describes the relevance between a document and a query as the generating probability of the query from the document’s model:

$$P(d|q) \propto P(q|d)P(d) \tag{2}$$

For a query q , we usually assume that terms appear independently in it, thus:

$$P(q|d) = \prod_{t_i \in q} P(t_i|d) \tag{3}$$

where t_i is the i -th term in q and $P(t_i|d)$ represents the probability of generating term t_i from the language model of document d . A common method for estimating $P(t_i|d)$ is maximum likelihood estimation and Dirichlet smoothing [1], as follows:

$$P(t_i|d) = \lambda \cdot \frac{tf(t_i, d)}{|d|} + (1 - \lambda) \cdot \frac{tf(t_i, D)}{|D|}, \quad \lambda = \frac{|d|}{|d| + \mu} \tag{4}$$

where $|d|$ is the length of document d ; $tf(t_i, d)$ is the term frequency of term t_i in d ; $|D|$ is the number of documents in the document collection D ; $tf(t_i, D)$ is the term frequency of term t_i in D ; λ is a parameter ranging in $[0, 1]$ and is often set based on the length of document d ; μ is another parameter and is commonly set as the average document length in D .

3.2 Language Models for Expert Finding

The simplest method to apply language model for expert finding is to merge all support documents of a candidate expert together and treat them as a virtual document, then employ the language model described in Section 3.1 to estimate the relevance between the virtual document and the query. However, this model has obvious disadvantages: it cannot differentiate the contributions of different support documents. Based on the consideration, two extended language models have been proposed (we call them as composite language model and hybrid language model).

3.2.1 Composite Language Model

Let $D_e = \{d_j\}$ denotes the collection of support documents related to a candidate e . In the composite language model, each support document d_j is viewed as a unit and the estimations of all the documents of a candidate e are combined. We have:

$$P(q|e) = \sum_{d_j \in D_e} P(q|d_j)P(d_j|e) \quad (5)$$

The model consists of two components: 1) a document that is related to a candidate is selected with probability $P(d_j|e)$; and 2) the query q is generated from the selected document with probability $P(q|d_j)$. The former actually indicates how a document d_j characterizes the candidate e . The probability is often viewed as identical in many language modeling applications. That is, set $P(d_j|e)$ to be 1 if expert e is the author of document d_j , otherwise 0. Let $q = \{t_i\}$, the probability $P(q|d_j)$ is estimated by Equation (3) and (4) based on the independent assumption. Finally, we obtain:

$$P(q|e) = \sum_{d_j \in D_e} P(d_j|e) \prod_{t_i \in q} P(t_i|d_j) \quad (6)$$

We call this model as composite model because it first integrates the probability of document d_j generating each term t_i and then combines the different document models together. The nature of the composite model is that it views documents as a “hidden” variable separating the query from a candidate such that the candidate is not directly modeled. It is based on the assumption that terms are independent in d_j . Accordingly, the model emphasizes the co-occurrence of all the query terms in the same document and gives penalty to the document that does not match the whole query [2] [21]. As for the example in Figure 1, the composite model can find the two experts for the query “semantic web”. However, it does not work well for the other two queries “machine learning” and “natural language processing”.

3.2.2 Hybrid Language Model

The hybrid language model (cf. Equation (7)) is similar to the composite model, except that it describes each term t_i using a combination of support documents models and then uses a language model to integrate them together.

$$P(q|e) = \prod_{t \in q} \sum_{d_j \in D_e} P(t|d_j)P(d_j|e) \tag{7}$$

The two models are not equivalent mathematically since the product and the sum cannot be interchanged. The nature of the hybrid model is that it collects all terms information from all documents associated with the given candidate and models the candidate directly. It is based on the assumption that terms are independent in all support documents of e . Thus the model does not care much about the co-occurrence of the query terms in the same support document [2] [21]. As for the example in figure 1, the hybrid model works well for both the queries “semantic web” and “machine learning”, as the query terms appear in the support documents of experts. Unfortunately, it cannot find the two experts for “natural language processing” because it is still based on lexical-level relevance assumption.

3.3 A Mixture Model for Expert Finding

We propose a mixture model for expert finding. We assume that there is a hidden ‘semantic’ theme layer $\Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$ between query q and document d_j . Each hidden theme θ_m is semantically associated with multiple queries and support documents. Similarly, each support document or query is also associated with multiple themes, respectively. In this way, given a query and a support document, we do not directly model the relevance between them. Instead, we use the hidden themes associated to them as the bridge to model the relevance. More accurately, we have:

$$P(q|d_j) = \sum_{m=1}^k P(q|\theta_m)P(\theta_m|d_j) \tag{8}$$

Here, $P(q|\theta_m)$ denotes the probability of generating a query given a theme and $P(\theta_m|d)$ denotes the probability of generating a theme given a document.

We assume that a query q and a document d are conditional independent given a theme θ_m . Then the problem becomes, for each document, how to estimate the probability $P(\theta_m|d_j)$ and for each query, how to estimate the probability $P(q|\theta_m)$, called parameter estimation. Following we introduce the method for parameter estimation.

Let T as all terms occurring in the whole document collection D . Suppose there are k hidden themes. The generative process of the data set can be described as:

- (1) Select a document d with probability $P(d)$;
- (2) Pick a latent theme θ_m with probability $P(\theta_m|d)$;
- (3) Generate a term t with probability $P(t|\theta_m)$.

As a result, we obtain an observed pair (t, d) without θ_m .

The above generative process can be expressed as a joint probability model:

$$P(t, d) = P(d)P(t|d), \text{ where } P(t|d) = \sum_{m=1}^k P(t|\theta_m)P(\theta_m|d) \tag{9}$$

Equation (9) sums over all θ_m from which the observations could have been generated, which is based on the assumption that t and d are conditional independent on θ_m . We use Bayes’ formula to transform Equation (9) to get its symmetric form:

$$P(t, d) = \sum_{m=1}^k P(t|\theta_m)P(d|\theta_m)P(\theta_m) \tag{10}$$

In order to explain the observations (t, d) , we need to estimate $P(t|\theta_m)$, $P(d|\theta_m)$ and $P(\theta_m)$ by maximizing of the log-likelihood function:

$$L = \sum_{d \in D} \sum_{t \in T} n(d, t) \log \sum_{m=1}^k P(t|\theta_m)P(d|\theta_m)P(\theta_m) \tag{11}$$

where $n(d, t)$ denotes the co-occurrence times of d and t .

We use Expectation-Maximization (EM) algorithm [5] to estimate the maximum likelihood. The EM algorithm begins with some initial values of $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ and runs an iterative process to obtain new values based on updating formulas. The update formulas contain expectation (E) step and maximization (M) step.

In E-Step, we aim to compute the posterior probability of latent theme θ_m , based on the current estimates of the parameters:

$$P(\theta_m | d, t) = \frac{P(t|\theta_m)P(d|\theta_m)P(\theta_m)}{\sum_{m=1}^k P(t|\theta_m)P(d|\theta_m)P(\theta_m)} \tag{12}$$

In M-Step, we aim to maximize the expectation of the log-likelihood of Equation (11). By introducing Lagrange multipliers and solving partial derivative, we can obtain the following equations for re-estimated parameters:

$$P(d|\theta_m) = \frac{\sum_{t \in T} n(d, t)P(\theta_m | d, t)}{\sum_{d \in D} \sum_{t \in T} n(d, t)P(\theta_m | d, t)} \tag{13}$$

$$P(t|\theta_m) = \frac{\sum_{d \in D} n(d, t)P(\theta_m | d, t)}{\sum_{t \in T} \sum_{d \in D} n(d, t)P(\theta_m | d, t)} \tag{14}$$

$$P(\theta_m) = \frac{\sum_{d \in D} \sum_{t \in T} n(d, t)P(\theta_m | d, t)}{\sum_{d \in D} \sum_{t \in T} n(d, t)} \tag{15}$$

The E-step and M-step run iteratively until the log-likelihood function converges to a local maximum. Then we obtain the parameters: $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$.

3.4 Find Experts Using the Model

We can make inferences based on the estimated probabilities. Given a query, the probability $P(q|\theta_m)$ can be estimated by

$$P(q|\theta_m) = \prod_{t \in q} P(t|\theta_m) \tag{16}$$

Then Equation (8) can be rewritten as:

$$P(q|d_j) = \sum_{m=1}^k \prod_{t \in q} P(t_i|\theta_m)P(\theta_m | d_j) \tag{17}$$

Therefore we obtain Equation (18) by substituting $P(q|d_j)$ into Equation (5):

$$P(q|e) = \sum_{d_j \in D} \sum_{m=1}^k \prod_{t_i \in q} P(t_i | \theta_m) P(\theta_m | d_j) P(d_j | e) \quad (18)$$

where $P(\theta_m | d_j)$ can be estimated by Bayes' formula:

$$P(\theta_m | d_j) = \frac{P(d_j | \theta_m) P(\theta_m)}{P(d_j)} \square P(d_j | \theta_m) P(\theta_m) \quad (19)$$

Now, we get the probability $P(q|e)$. We can further obtain $P(elq)$ by $P(elq) \propto P(q|e) P(e)$, where $P(e)$ is often viewed as uniform in previous work such as [3]. However, we have found that final results sometimes are sensitive to the probability. In this work, we employ the propagation approach we have proposed in [25] to estimate $P(e)$. The approach is based on the social relationship analysis. The basic idea is that if a person knows many experts on a topic or if the person's name co-occurs many times with the known experts, then it is more likely that he/she is an expert on the topic. Finally we obtain $P(elq)$ for each candidate and sort the candidates accordingly.

4 Experiments

In this section, we first introduce the experimental setting. Then we present the experimental results. Finally we give some discussions.

4.1 Experimental Setting

We evaluate the work in the context of ArnetMiner[22]. ArnetMiner contains 448,289 researchers and 725,655 publications extracted from the Web database, pages, and files. As performing PLSA on the full data collection will take an extreme long time, we created a subset of the data for evaluation purpose. Specifically, we first selected the most frequent queries from the log of ArnetMiner (by removing the specific queries or too long queries, e.g., 'A convergent solution to tensor subspace learning'). We also removed the similar queries (e.g., 'web service' v.s. 'web services'). Then we obtained seven queries: 'information extraction' (IE), 'machine learning' (ML), 'semantic web' (SW), 'natural language processing' (NLP), 'support vector machine' (SVM), 'planning' (PL), and 'intelligent agents' (IA). Next, for each query, we gathered the top 30 persons from Libra author search, Rexa authors search, and ArnetMiner¹. We merged all the persons together by removing ambiguous names (e.g., L. Liu) and names that do not exist in ArnetMiner. Finally we got 421 person names. We collected 14,550 publications of the 421 persons from ArnetMiner as the support document collection.

For evaluation, it is difficult to find a standard data set as the ground truth. As a result, we use the method of pooled relevance judgments [8] together with human judgments. Specifically, for each query, we first pooled the top 30 results from the above three systems (Libra, Rexa, and ArnerMiner) into a single list. Then, one faculty and two graduates, from the authors' lab, provided human judgments. Assessments were carried out mainly in terms of how many publications he/she has

published, how many publications are related to the given query, how many top conference papers he/she has published, what distinguished awards he/she has been awarded. Finally, the judgment scores were averaged to construct the final ground truth. The data set is available on line.

We conducted evaluation in terms of $P@5$, $P@10$, $P@20$, $P@30$, R -prec, Mean Average Precision (MAP) and P-R curve [8] [10].

We used the language models introduced in Section 3.2 as baselines. Hereafter, we respectively call them CM and HM. For comparison purpose, we also report the results obtained by Libra and Rexa.

We implemented our proposed model (shortly MM) in two stages. In the first stage, we use PLSA algorithm (equations (12)-(15)) to estimate the probabilities $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ for each document, term, and theme. Here, documents denote publications. Terms are extracted from the titles and conference names of the publications after word segmentation and stop words filtering. We empirically set the number of themes as 300 (cf. figure 3 for the effect of the number of themes). In the second stage, we rank experts using equation (18) for each query.

4.2 Experimental Results of Expert Finding

Table 1 shows the performances on the 7 queries by our proposed model, the two language models, and the two systems (Libra and Rexa). Figure 2 shows the average 11-point precision recall curves on the 7 queries for the different approaches. We see that in terms of most of the measures, the proposed model outperforms the two baseline language models. We also present top 9 example experts for “natural language processing” ranked by different approaches in Table 2.

4.3 Discussions

(1) Improvements Over Baselines. Our proposed model outperforms the two language models in terms of $P@5$, $P@10$ and MAP . From the PR curve, we can also see that our model outperforms the language models in most of the 11 points, which confirms the effectiveness of our approach. The proposed model can retrieve experts whose support documents do not contain the query terms but ‘semantically’ related to the query, therefore our approach can improve the performance significantly. For example, in Table 2, our model MM ranks higher for “Raymond J. Mooney” than the language models. This is because many of Mooney’s papers do not exactly contain the query terms although they are related to “natural language processing”. We rank higher for “Dan Roth” and “Dragomir R. Radev” due to the similar reason.

(2) Effect of the Number of Themes. The best number of themes is difficult to determine. In our experiment, we tried to tune the parameter to get better performance. As Figure 3 shows, the number of themes systematically varies from 10 to 100 with interval 10 and from 100 to 1000 with interval 100. In general, the best results were obtained when setting the number of themes as 300.

An intuitive explanation to Figure 3 is that when the number of theme is small, the estimated mixture model prefers to very general queries; with the number increasing,

the model prefers to specific queries. The number 300 seems to be a best balance in our setting. Table 3 show two themes with the representative words, respectively for #theme=10 and #theme=300.

(3) Language Models. We also analyze the retrieval results of two language models. From table 1, we see that for queries “SW”, “IE” and “SVM”, CM performs better than HM, because the word “web”, “information” and “machine” may slightly drive the topic of documents drift away when using HM. For the queries of “PL”, “IA”, “ML”, and “NLP”, HM performs better than CM, due to the limitation in CM that all the query terms should co-occur in one document.

Table 1. Performances of different expert finding approaches (%)

Query	Approach	P@5	P@10	P@20	P@30	R-pre	MAP
SW	Libra	80.00	70.00	80.00	66.67	60.00	71.28
	Rexa	80.00	60.00	55.00	43.33	37.78	52.65
	CM	80.00	80.00	75.00	70.00	62.22	76.70
	HM	80.00	80.00	85.00	76.67	60.00	69.25
	MM	100.00	100.00	75.00	60.00	57.78	72.20
IE	Libra	100.00	60.00	50.00	36.67	50.00	67.76
	Rexa	60.00	60.00	45.00	36.67	45.00	51.88
	CM	80.00	70.00	65.00	56.67	65.00	73.16
	HM	80.00	70.00	60.00	56.67	60.00	71.96
	MM	100.00	70.00	60.00	56.67	60.00	75.03
SVM	Libra	60.00	30.00	25.00	30.00	32.26	37.22
	Rexa	60.00	60.00	40.00	36.67	35.48	43.75
	CM	100.00	90.00	75.00	66.67	64.52	79.47
	HM	100.00	100.00	80.00	60.00	58.06	76.61
	MM	100.00	100.00	80.00	63.33	61.29	81.56
PL	Libra	60.00	60.00	65.00	53.33	48.57	57.02
	Rexa	60.00	70.00	60.00	46.67	42.86	52.50
	CM	80.00	70.00	65.00	56.67	54.29	70.14
	HM	100.00	90.00	75.00	60.00	54.29	73.07
	MM	80.00	90.00	70.00	60.00	54.29	74.04
IA	Libra	80.00	50.00	40.00	26.67	26.67	49.63
	Rexa	60.00	40.00	35.00	40.00	40.00	43.90
	CM	80.00	70.00	60.00	53.33	53.33	70.06
	HM	100.00	80.00	65.00	60.00	60.00	78.18
	MM	100.00	100.00	70.00	50.00	50.00	82.29
ML	Libra	60.00	40.00	35.00	30.00	29.27	33.88
	Rexa	80.00	70.00	60.00	46.67	34.15	52.52
	CM	60.00	60.00	50.00	46.67	46.34	54.96
	HM	60.00	60.00	60.00	56.67	53.66	60.07
	MM	80.00	80.00	65.00	53.33	51.22	66.70
NLP	Libra	40.00	30.00	35.00	43.33	36.59	40.49
	Rexa	20.00	20.00	30.00	26.67	24.39	26.29
	CM	40.00	70.00	65.00	50.00	0.00	61.76
	HM	80.00	70.00	55.00	60.00	48.78	68.93
	MM	100.00	80.00	65.00	60.00	48.78	76.07
AVE	Libra	68.57	48.57	47.14	40.95	40.48	51.04
	Rexa	60.00	54.29	46.43	39.52	37.09	46.21
	CM	74.29	72.86	65.00	57.14	49.39	69.46
	HM	85.71	78.57	68.57	61.43	56.40	71.15
	MM	94.29	88.57	69.29	57.62	54.76	75.41

Table 2. Top 9 experts for “natural language processing” by five expert finding approaches

MM	CM	HM	Libra	Rexa
Raymond J. Mooney	Rebecca F. Bruce	Janyce Wiebe	Eric Brill	W. Addison Woods
Dan Roth	Janyce Wiebe	Michael Collins	Christopher D. Manning	Klaus Netter
Michael Collins	Veronica Dahl	Aravind K. Joshi	Adam L. Berger	Yorick Wilks
Janyce Wiebe	Robert J. Gaizauskas	Raymond J. Mooney	Stephen Della Pietra	Kavi Mahesh
Aravind K. Joshi	Kevin Humphreys	Rebecca F. Bruce	Vincent J. Della Pietra	Robert H. Baud
Rebecca F. Bruce	Aravind K. Joshi	Veronica Dahl	David D. Lewis	Kevin Humphreys
Veronica Dahl	Philippe Blache	Robert J. Gaizauskas	Kenneth Ward Church	Philippe Blache
Claire Cardie	Eric Brill	Thomas Hofmann	Hinrich Schutze	Victor Raskin
Oren Etzioni	Raymond J. Mooney	Eric Brill	Lillian Jane Lee	Lorna Balkan

(4) Decline Over Baselines. In terms of $p@20$, $p@30$ and R -prec, we must note that our model underperforms the two language models. The reason lies in that our model may also bring some noises when estimating the probabilities $P(t|\theta_m)$, $P(d|\theta_m)$, and $P(\theta_m)$ in the first stage.

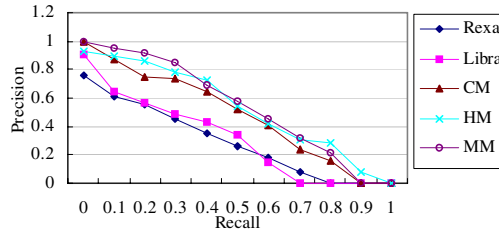


Fig. 2. Average Precision-recall curves of five expert finding approaches for 7 queries

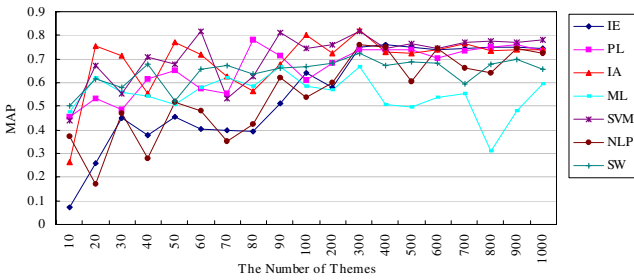


Fig. 3. The effect of the number of themes

Table 3. Example themes discovered by PLSA with #themes=10 and #themes=300. Each theme is shown with 10 representative words.

#Themes = 10										
Theme #2	information	design	framework	intelligent	ontology	management	based	semantic	systems	web
Theme #3	KDD	neural	from	text	selection	networks	Time	data	mining	using
#Themes = 300										
Theme #12	spelling	roadmap	ebl	correction	scoring	question	Directions	answering	ICGA	syntax
Theme #64	zero	variance	manifolds	predictions	principal	transformation	ICPR	matrix	clustering	words

5 Related Work

5.1 Language Model for Expert Finding

With the launch of expert finding task in TREC 2005, more and more researchers begin focusing on the research topic. Previous work for expert finding usually makes use of language models. For example, Cao et al. [9] propose a two-stage language model which combines a co-occurrence model to retrieve documents given a query, and a relevance model to find experts in those documents. Balog et al. [2] propose a model which models candidate using support documents directly and another model which is similar to the model of Cao. [3] studies the expert finding problem in a sparse data environments and proposes several advanced models based on the characteristics of the dataset. Petkova et al. analyze and compare different language models proposed for the task of finding experts [21]. They argue that all the models are probabilistically equivalent and the differences lie in the independent assumptions. As far as we know, expert finding by using latent semantic analysis has not been investigated previously.

5.2 Probabilistic Latent Semantic Analysis and Its Applications

The idea of using latent semantic structure in information retrieval traces back to [13]. They propose latent semantic analysis (LSA) method, which is mostly used in automatic indexing and information retrieval [4]. The main idea is to map data using Singular Value Decomposition (SVD) from a high-dimensional vector space representation to a reduced lower representation, also called latent semantic space.

A new approach to discover latent variables is Probabilistic latent semantic analysis (PLSA) proposed by Thomas Hofmann [16]. The difference between LSA and PLSA is that the latter one is based on the likelihood principle and defines a proper generative model of the data; hence it results in a more solid statistical foundation. The core of PLSA is a statistical model called aspect model, which assumes there exists a set of hidden factors underlying the co-occurrences among two sets of objects. Expectation Maximization (EM) algorithm [5] is used to estimate the probabilities of the hidden factors generating the two sets of objects.

Probabilistic Latent Semantic Analysis has been used to solve problems in a variety of applications on account of its flexibility. Such applications include information retrieval [16], text learning and mining [6] [7] [14] [18] [24], co-citation analysis [11] [12], social annotation analysis [23], web usage mining [17] and personalize web search [19].

6 Conclusion

In this paper, we have proposed a mixture model for expert finding. We assume that there is a latent theme layers between terms and documents and employ the themes to help discover semantically related experts to a given query. A EM based algorithm has been employed for parameter estimation in the proposed model. Experimental results on real data show that our proposed model can achieve better performances than the conventional language models. As future work, we plan to investigate how to automatically determine the number of themes based on the input query.

References

- [1] Baeza-Yates, R., Ribeiro-Neto, B.: *Modern Information Retrieval*. ACM Press, New York (1999)
- [2] Balog, K., Azzopardi, L., de Rijke, M.: Formal Models for Expert Finding in Enterprise Corpora. In: *Proc. of SIGIR 2006*, pp. 43–55 (2006)
- [3] Balog, K., Bogers, T., Azzopardi, L., Rijke, M., Bosch, A.: Broad Expertise Retrieval in Sparse Data Environments. In: *Proc. of SIGIR 2007*, pp. 551–558 (2007)
- [4] Berry, M., Dumais, S., O'Brien, G.: Using Linear Algebra for Intelligent Information Retrieval. *SIAM Review* 37, 573–595 (1995)
- [5] Bilmes, J.A.: A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models. Berkeley, ICSI TR-97-021 (1997)
- [6] Brants, T., Chen, F., Tsochantaridis, I.: Topic-based Document Segmentation with Probabilistic Latent Semantic Analysis. In: *Proc. of CIKM 2002*, pp. 211–218 (2002)
- [7] Brants, T., Stolle, R.: Find Similar Documents in Document Collections. In: *Proc. of LREC 2002* (2002)
- [8] Buckley, C., Voorhees, E.M.: Retrieval Evaluation with Incomplete Information. In: *Proc. of SIGIR 2004*, pp. 25–32 (2004)
- [9] Cao, Y., Liu, J., Bao, S., Li, H.: Research on Expert Search at Enterprise Track of TREC (2005)
- [10] Craswell, N., de Vries, A., Soboroff, I.: Overview of the Trec-2005 Enterprise Track. In: *TREC 2005 Conference Notebook*, pp. 199–205 (2005)
- [11] Cohn, D., Chang, H.: Learning to Probabilistically Identify Authoritative Documents. In: *Proc. of ICML 2000*, pp. 167–174 (2000)
- [12] Cohn, D., Hofmann, T.: The Missing link: A Probabilistic Model of Document Content and Hypertext Connectivity. In: *Neural Information Processing Systems 13*, MIT Press, Cambridge (2001)
- [13] Deerwester, S., Dumais, S., Furnas, G., Landauer, T., Harshman, R.: Indexing by Latent Semantic Analysis. *Journal of the American Society for Information Science* 41(6) (1990)
- [14] Gaussier, E., Goutte, C., Popat, K., Chen, F.: A Hierarchical Model for Clustering and Categorizing Documents. In: Crestani, F., Girolami, M., van Rijsbergen, C.J.K. (eds.) *ECIR 2002*. LNCS, vol. 2291, pp. 229–247. Springer, Heidelberg (2002)
- [15] Hawking, D.: Challenges in Enterprise Search. In: *Proc. of the Fifteenth Conference on Australasian Database*, vol. 27, pp. 15–24 (2004)
- [16] Hofmann, T.: Probabilistic Latent Semantic Analysis. In: *Proc. of UAI 1999* (1999)
- [17] Jin, X., Zhou, Y., Mobasher, B.: Web Usage Mining based on Probabilistic Latent Semantic Analysis. In: *Proc. of SIGKDD 2004*, pp. 197–205 (2004)
- [18] Kim, Y., Chang, J., Zhang, B.: An Empirical Study on Dimensionality Optimization in Text Mining for Linguistic Knowledge Acquisition. In: Whang, K.-Y., Jeon, J., Shim, K., Srivastava, J. (eds.) *PAKDD 2003*. LNCS (LNAI), vol. 2637, pp. 111–116. Springer, Heidelberg (2003)
- [19] Lin, C., Xue, G., Zeng, H., Yu, Y.: Using Probabilistic Latent Semantic Analysis for Personalized Web Search. In: Zhang, Y., Tanaka, K., Yu, J.X., Wang, S., Li, M. (eds.) *AP-Web 2005*. LNCS, vol. 3399, pp. 707–717. Springer, Heidelberg (2005)
- [20] Nie, Z., Ma, Y., Shi, S., Wen, J., Ma, W.: Web Object Retrieval. In: *Proc. of WWW 2007*, pp. 81–90 (2007)
- [21] Petkova, D., Croft, W.B.: Generalizing the Language Modeling Framework for Named Entity Retrieval. In: *Proc. of SIGIR 2007* (2007)

- [22] Tang, J., Zhang, D., Yao, L.: Social Network Extraction of Academic Researchers. In: Proc. of ICDM 2007, pp. 292–301 (2007)
- [23] Wu, X., Zhang, L., Yu, Y.: Exploring Social Annotations for the Semantic Web. In: Proc. of WWW 2006, pp. 417–426 (2006)
- [24] Zhai, C., Velivelli, A., Yu, B.: A Cross-collection Mixture Model for Comparative Text Mining. In: Proc. of SIGKDD 2004, pp. 743–748 (2004)
- [25] Zhang, J., Tang, J., Li, J.: Expert Finding in a Social Network. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 1066–1069. Springer, Heidelberg (2007)

On Privacy in Time Series Data Mining

Ye Zhu¹, Yongjian Fu¹, and Huirong Fu²

¹ Cleveland State University, Cleveland, OH, 44115, USA

{y.zhu61, y.fu}@csuohio.edu

² Oakland University, Rochester, MI 48309, USA

fu@oakland.edu

Abstract. Traditional research on preserving privacy in data mining focuses on *time-invariant* privacy issues. With the emergence of time series data mining, traditional *snapshot-based* privacy issues need to be extended to be multi-dimensional with the addition of *time dimension*. We find current techniques to preserve privacy in data mining are not effective in preserving time-domain privacy. We present data flow separation attack on privacy in time series data mining, which is based on blind source separation techniques from statistical signal processing. Our experiments with real data show that this attack is effective. By combining the data flow separation method and the frequency matching method, an attacker can identify data sources and compromise time-domain privacy. We propose possible countermeasures to the data flow separation attack in the paper.

1 Introduction

With the popularity of data mining, privacy issues have been a serious concern. Most research on privacy issues in data mining focuses on privacy preserving data mining, i.e., how to mine data while protecting the identity of data owners. Various approaches have been proposed to conduct data mining without breaching of privacy [1,2,3,4]. However, privacy issues studied in previous research are on time-invariant data which do not change over time.

Time series data mining becomes popular recently. The goal of time series data mining is to find patterns contained in time series data [5,6,7,8,9,10,11,12]. In time series data mining, the data to be mined is labeled with timestamps. One example is the daily stock price. For time series data, because of the special nature of the data, its privacy goes beyond the protection of data. In this paper, when the meaning of privacy is unclear from context, we call the privacy in time-invariant data mining *time-invariant privacy*, and the privacy in time series data mining *time series privacy*.

We focus on time series privacy issues in this paper. As snapshot privacy issues arise from snapshot based data mining, time series privacy issues arise from time series data mining. Time series privacy issues concern about changes in data over time. We need to protect data, as well as its properties in time and frequency domains. For example, sales data on a car model changes over time, but

the manufacturer of the car model will worry about sharing the sales data with data miners because the sales data may indicate changes in financial situation or marketing strategies of the manufacturer over time. Another example is that a store may not be willing to share its sales data because a data miner may find out promotion periods of the store by checking periodicities contained in the data provided by the store. We argue that privacy in time series data involves protection of properties in time domain such as peak, trough, and trend, and properties in frequency domain, such as periodicity. Such properties reveal lots of information, even though they do not reveal data.

Two common approaches have been proposed to preserve snap-shot privacy in data mining. One approach is data perturbation in which data to be mined is modified to protect privacy. The other approach is data partitioning in which data is split among multiple parties and each party only see its share of the data. One method in data perturbation approach is aggregation in which time series data from different sources are aggregated and given to data miners. This can prevent data miners from finding private information about individual sources. For example, auto manufacturers usually do not want to publish daily, monthly or yearly sales data of individual car model because too much sensitive information is contained in the time series data. Instead, trusted market research companies aggregate sales data of different car models made by different auto manufacturers and publish these aggregated data. These time series data can be aggregated in different ways such as according to vehicle types or vehicle features for different purposes.

In this research, we found that current techniques to protect snap-shot privacy were largely ineffective under data flow separation attack, which can separate aggregated data and separate noise from original data. The data flow separation attack employs the *blind source separation* model [13], which was originally defined to solve *cocktail party problem*: blind source separation algorithms can extract one person's voice given the mixtures of voices in a cocktail party. Our experiments show that data flow separation can separate independent time series data generated by different sources.

The contributions of this paper can be summarized as follows:

- We introduce the concept of privacy in time series data mining. Because of the nature of time series data, privacy issues in time series data mining go beyond these in snap-shot data mining, especially privacy in time and frequency domains. We believe it is important to preserve privacy in time series data as well as in snap-shot data.
- We present *data flow separation attack* and show that aggregation is not always enough to protect time series privacy. We use experiments on real data to show that data flow separation attacks are effective.
- We present *data flow separation attack with noise*, a further attack based on data flow separation attacks, which can fully disclose sensitive information of data sources.
- We discuss the pros and cons of countermeasures to data flow separation attacks.

The rest of the paper is organized as follows: Section 2 reviews the related work in privacy preserving data mining and time series data mining. We list time series privacy issues in Section 3. Section 4 outlines the threat model. In Section 5, we introduce the data flow separation attack. We will also describe frequency matching that can be used as further attacks. In Section 6, we use experiments on real stock data to show the effectiveness of the data flow separation attack. Section 7 discusses countermeasures for data flow separation attack. We conclude this paper in Section 8, with remarks on extensions of this work.

2 Related Work

2.1 Privacy Preserving Data Mining

The main approaches to privacy-preserving data mining can be categorized into two categories: data perturbation and data partitioning.

In data perturbation approaches, original data is modified by data obscuration or by adding random noises. An example of data obscuration is replacing values of a continuous variable with ranges. Distributions of random noises are usually known, such as even distribution or normal distribution. The modified data is given to data miners. Algorithms have been developed to mine decision tree [1] and association rules [2] in data with noise. Techniques for improving randomization are also proposed [14].

In data partitioning approaches to privacy preserving data mining, the original data is distributed among multiple parties, either by the partitioning of centralized data or by the nature of data collection. The data mining process is split into local computation at individual sites and global computation. During the process, each party does not see other party's data, but cooperates to find global patterns. In almost all cases, secure multi-party computation [15] and encryptions are employed. Secure algorithms for decision tree construction [3], association rules mining [4], k-means clustering [16], and Bayesian network learning [17] have been proposed. In these algorithms, all parties were assumed to be semi-honest. That is, every party would faithfully follow the protocol or algorithm, but tried to learn as much as possible about others.

As discussed above, past research in privacy preserving data mining focuses on privacy of raw data. Though privacy of derived data has been mentioned [14], we are not aware of any research in time series privacy. We hope to raise the awareness of time series privacy issues in this paper.

2.2 Time Series Data Mining

Because time series data is usually large and noisy, direct application of data mining algorithms on raw data is time-consuming and gives unreliable results. A lot of attentions have been paid on preprocessing techniques that facilitate data mining tasks. Research in time series data mining mostly focuses on data preprocessing techniques, such as discretization and transformation [10], feature

extraction and feature reduction [12]. Work has been also been done in related techniques such as representation [11] and similarity metric [18].

The data mining tasks studied by researchers include subsequence matching [5], classification [7], clustering [8], time series modeling [9], and association rule mining [6].

It is clear that most research in time series data mining does not address privacy issues, let alone time series privacy issues. While current privacy preserving techniques can be applied to preserve snap-shot privacy in time series data, they are inadequate for protecting time series privacy.

3 Time Series Privacy Issues

We identify privacy issues for time series data in addition to traditional privacy issues in data mining. Time series data from a data source can be regarded as a time-domain signal. All the characteristics of a time-domain signal can be potentially regarded as private information by the data provider. Below, we list common characteristics in time series data that a data provider may need to keep confidential.

- Amplitude: Amplitude indicates the strength of a signal, like the raw data in traditional privacy research.
- Average: The average signal strength over time. For example, for a series of sales data, average amplitude indicates the average sales.
- Peak and trough: Peak and trough indicate extreme situations. They are usually confidential as they may disclose extreme changes in underlying causes such as difficult cash flow.
- Trend: By observing trends of time series data, an adversary may predict future changes of time series data. Thus trend information should be protected from competitors as well.
- Periodicity: Periodical changes in time series data indicate existence of periodically changing factors. For sales data of a store, the factor can be periodical changes in marketing strategies such as promotions which are usually regarded as confidential information for stores. Unlike the previous characteristics which are in time domain, periodicity is in frequency domain.

There are other characteristics which may be regarded as confidential by some data providers. However, as an initial study on time series privacy, we focus on the common characteristics listed above. Since data flow separation attack aims to recover original signal, the attack may be effective to disclose these common characteristics.

4 Threat Model

In this paper we assume that data providers care about the sensitive information contained in their time series data. To protect their privacy, data providers

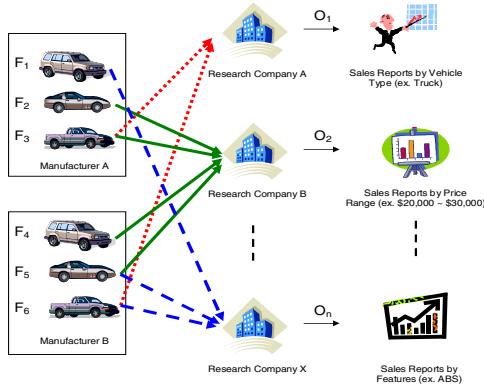


Fig. 1. An Example for Data Flow Model

will only supply their data to trusted research companies. Research companies will aggregate time series data provided by different data providers according to different criteria. An example of aggregating sales data provided by auto manufacturers is shown in Figure 1. In Figure 1, there is only one aggregation layer. In practice there can be many layers of aggregation because some research companies may aggregate data provided by other research companies or aggregate data provided by both original data providers and research companies.

We assume research companies will publish aggregated data for profit or for public usage. The research companies will disclose criteria used in aggregation, but not the information of data sources, specifically identities of data providers to protect the privacy of data providers.

We assume adversaries to have capabilities summarized as follows:

- Adversaries can obtain aggregated data from research companies free or for a small fee.
- Adversaries can not obtain data generated from original data sources because of lack of trust with original data sources. This assumption excludes the possibility of an original data provider being a privacy attacker. We do not study the case of compromised data provider in this paper. But obviously the data flow separation attack will be more effective if an adversary, being a provider of original data, can know part of original data aggregated by research companies.
- Adversaries can obtain data aggregated according to different criteria.
- Research companies have various data providers as their data sources and research companies do not want to disclose the composition of data sources. It is similar to an investment company does not want to disclose the composition of stocks in possess.

The threat model M can be represented as $M = \langle F, G, O \rangle$, where F is a set of original data sources, G is a set of aggregation operations, and O is a set of observations available to adversaries. Though observations are obtained by

applying aggregation operations to data sources, i.e., $O = G(F)$, aggregation operations G and data sources F are unknown to adversaries.

The model assumed in our paper is realistic. Many research companies compile weekly or monthly sales of large items, such as cars, TVs, computers, etc, from retailers or manufacturers. Each research company has its own sources and publishes its reports with aggregated totals. Since these reports are available with a small fee, someone can collect all these reports and try to separate data to recover original data. Manufacturers want to protect their data from third parties, but would like to see the aggregated data to understand their industry.

5 Data Flow Separation Attack

In this section, we will first define the problem in the context of blind source separation and then describe how to apply the data flow separation attack in practice.

5.1 Blind Source Separation

Blind source separation is a methodology in statistical signal processing to recover unobserved “source” signals from a set of observed mixtures of the signals. The separation is called “blind” to emphasize that the source signals are not observed and that the mixture is a black box to the observer. While no knowledge is available about the mixture, in many cases it can be safely assumed that source signals are independent. In its simplest form [19], the blind source separation model assumes n independent signals $F_1(t), \dots, F_n(t)$ and n observations of mixture $O_1(t), \dots, O_n(t)$ where $O_i(t) = \sum_{j=1}^n a_{ij} F_j(t)$. The goal of blind source separation is to reconstruct the source signals $F_j(t)$ using only the observed data $O_i(t)$, with the assumption of independence among the signals $F_j(t)$. The common methods employed in blind source separation are minimization of mutual information [20], maximization of nongaussianity [21] and maximization of likelihood [22].

5.2 Data Flow Separation as a Blind Source Separation Problem

In this paper, we define an *individual data flow* as a series of time-stamped data generated by an original data source. *Aggregate data flow* is defined as the aggregate of individual data flows. Aggregate data flows are generated by research companies. If not specified, the phrase *data flow* in the remaining of this paper means the individual data flow for brevity.

For an attacker who is interested in sensitive information contained in individual data flow, it will be very helpful to separate the individual data flows based on the aggregate data flows. Because the separation of the data flows can recover the pattern of data flows, they can be use for further attack such as frequency matching attack described in Section 5.3.

In this paper, we are interested in patterns carried in the time series data. For example, in Figure 1, the attacker can get a time series $O_1 = [o_1^1, o_2^1, \dots, o_n^1]$ of aggregate data flow from Research Company A. We call n as the sample

size in this paper. The attacker’s objective is to recover the time series $F_i = [f_1^i, f_2^i, \dots, f_n^i]$ for each individual data flow.

In general, with l research companies and m individual data flows, we can rewrite the problem in vector-matrix notation,

$$\begin{pmatrix} O_1 \\ O_2 \\ \vdots \\ O_l \end{pmatrix} = \mathbf{A}_{l \times m} \begin{pmatrix} F_1 \\ F_2 \\ \vdots \\ F_m \end{pmatrix} \tag{1}$$

where $\mathbf{A}_{l \times m}$ is called the mixing matrix in blind source separation problems.

Data flow separation can be achieved using blind source separation techniques. The individual data flows are independent from each other since they are from different sources. Given the observations O_1, O_2, \dots, O_l , blind source separation techniques can be used to estimate the independent individual flows F_1, F_2, \dots, F_m by maximizing the independence among the estimated flows.

The issues about the blind source separation method are summarized as follows.

- Basic blind source separation algorithms require the number of observations to be greater than or equal to the number of independent sources. For data flow separation, it means $l \geq m$. Furthermore, we assume $m = l$ in this paper since it is fairly straightforward to extend our idea to cases where $l > m$.
- The l observations may have redundancy. In other words, the row vectors of the mixing matrix may be linearly dependent. The cost of the redundancy will be that some independent data flows are not separated.
- The data flow estimations by blind source separation algorithms are usually lifted, scaled versions of the actual data flows. Sometimes, the estimated data flow may be of different sign than the actual data flow. However, the attacker can still find characteristics of the actual data flow from the estimated data flow. Also, heuristic approaches can be used to fine tune the estimation, which is an interesting topic for further research.

5.3 Frequency Matching Attack

After the data flows have been separated, a number of data flows, each with a given time series, have been determined to be included in the aggregate.

We choose frequency spectrum matching to do further attack. Frequency spectrum can be generated by applying Discrete Fourier Transform on time series data as below

$$X(k) = \sum_{j=0}^{N-1} f_j' e^{-\frac{2\pi i k j}{N}} \tag{2}$$

where f_j' denotes the j^{th} data point in the time series data and N denotes the length of the time series and then calculating the magnitude of transformed data. We match frequency spectrum by correlation.

The rationale for the use of frequency matching is two-fold: First, the dynamics of many data flows, such as sales, stock price, and weather, are characterized

by their periodicities. By matching the frequency spectrum of a known data flow with the frequency spectrum of estimated data flows obtained by blind source separation techniques, we can identify corresponding flows with high accuracy. Second, frequency matching can easily remove the ambiguities introduced by the lifting and scaling in the estimated time series by removing the zero-frequency component.

Frequency matching can be applied to match data flows separated from different attacks. After collecting a set of aggregate data flows according to different criteria, an attacker can select arbitrary subsets as groups and apply data flow separation techniques on the groups to recover individual data flows. If a data flow separated from one group matches a data flow separated from another group, then these two data flows should be generated from the same source. Moreover, the source generating these two data flows should satisfy at least one aggregation criteria in each group. If the attacker can match a data flow with data flows separated from several groups, the attacker can largely reduce the anonymity or possibly determine the identify of the source generating the data flow since the source should satisfy at least one criteria in each of these groups of aggregate flows. To better utilize the data, the attacker can try all possible combinations to group available aggregate data flows and then match the data flows separated from these groups. Of course, when the number of aggregate flows in a group is too small, the data flow separation technique can not separate all data flows because the number of observations is smaller than the number of independent sources.

6 Evaluation

In this section, we will evaluate the performance of data flow separation. We use the blind source separation algorithm proposed in [23] to separate the data flows. The accuracy of separation will be measured using correlation with actual flows. In our experiments, real stock market data [24] is used.

6.1 Performance Metrics

In the following, we will adopt two metrics to evaluate the accuracy of data flow separation. Both metrics are based on a comparison of the separated data flows with the actual data flows.

As first performance metric, we use the correlation coefficient (ρ), a widely used performance criterion in blind source separation research. Let $F_A = [f_1^A, f_2^A, \dots, f_n^A]$ represent the time series of the actual data flow and $F_B = [f_1^B, f_2^B, \dots, f_n^B]$ represent the time series estimated by the blind source separation algorithm. To match the time series F_A with F_B , we first need to scale and lift F_B so that they have the same mean and variance.

$$F'_B = \frac{std(F_A)}{std(F_B)} \cdot (F_B - mean(F_B) \cdot [1, 1, \dots, 1]) + mean(F_A) \cdot [1, 1, \dots, 1] \quad (3)$$

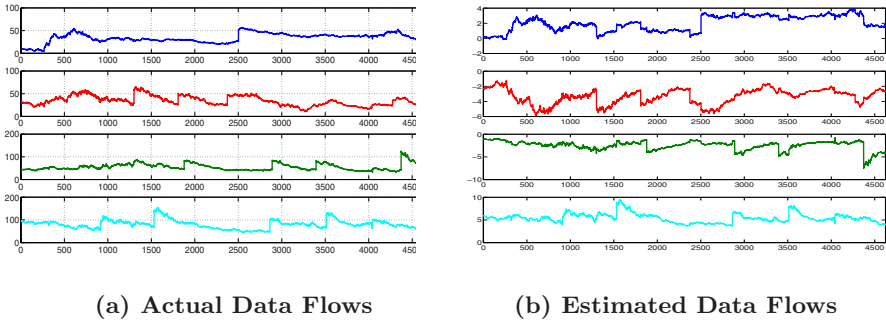


Fig. 2. Example of Data Flow Separation

where $std(F)$ and $mean(F)$ denote the standard deviation and the average of time series F , respectively. The correlation coefficient, $\varepsilon_{A,B}$, is defined as follows:

$$\varepsilon_{A,B} = \frac{\|F_A - F'_B\|^2}{n} \tag{4}$$

Since the times series F_B can also be a flipped version of F_A , we also need to match F_A with $-F_B$.

As the second metric, we use correlation coefficient, R_{F_A,F_B} , between the separated flow F_B and the corresponding actual flow F_A defined as follows:

$$R_{F_A,F_B} = \frac{\sum_i (f_i^A - mean(F_A))(f_i^B - mean(F_B))}{std(F_A)std(F_B)} \tag{5}$$

6.2 A Small Example

In this experiment, four time series of stock price selected from [24] are mixed into four aggregates. Figure 2(a) and Figure 2(b) show the actual data flows and separated data flows from the aggregates.

We can observe for data flows 1, 2, and 3, the separated data flows are flipped, scaled and lifted versions of the corresponding actual data flows. We can also observe the resemblance between separated flow and the corresponding actual flow for data flow 4.

Figure 3 shows the performance of data flow separation in terms of metrics introduced in Section 6.1. As shown in Figure 3(a), the separated data flows are highly correlated to actual data flows. In Figure 3(b), both the separated data flow and its flipped time series are compared against the actual flows and the mean square error for each data flow shown in the figure is the smaller one. From Figure 3(b), we can observe that the reconstructed data flows are off by around 10% in comparison with the actual data flows. Both metrics indicate that the data flow separation is successful. In the following we will use correlation only to evaluate performance because the lifting and scaling in the mean square error metrics may introduce error.

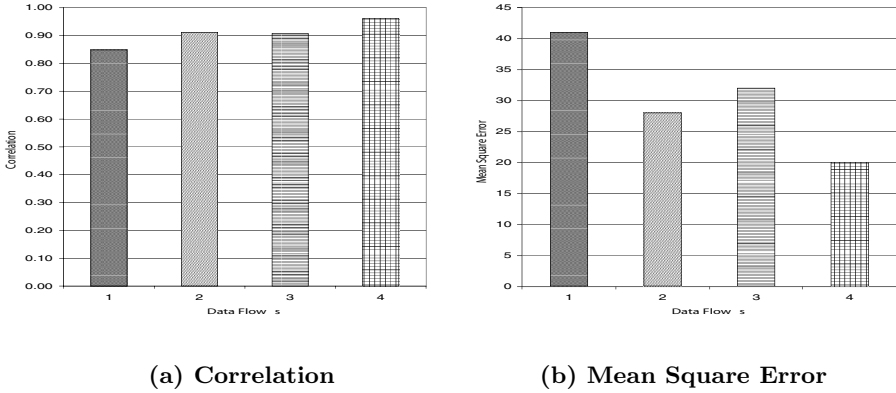


Fig. 3. Performance of Data Flow Separation on a Small Example

6.3 Mixing Degree

In this set of experiments, we would like to study the effect of mixing degree on the performance of data flow separation. We define mixing degree as follows:

$$D_{mix} = \frac{\text{average number of individual data flows mixed in aggregates}}{\text{number of individual data flows}} . \quad (6)$$

It is equivalent as

$$D_{mix} = \frac{\text{number of non-zero entries in } A_{l \times m}}{l \times m} . \quad (7)$$

Ten time series selected from stock data [24] are mixed randomly in this experiment to create ten aggregates. Totally, 10000 randomly-generated 10×10 binary mixing matrices were used in this experiment.

Figure 4(a) shows the effect of mixing degree on the performance of data flow separation. We plot statistics of both average correlation and worst case correlation. In this paper average correlation is defined as mean of correlation between separated data flows and actual data flows for each trial. We use worst case to refer to the most accurately separated data flow in each trial. It corresponds to worst privacy compromising in each trial.

From Figure 4(a), we can observe that data flow separation is effective since the separated flows are highly correlated to actual flows especially for the worst case. We can also observe that the performance of data flow separation is not sensitive to mixing degree for full-rank mixing matrices. This experiment indicates that countermeasure to data flow separation attack by simply increasing mix degree is not effective.

6.4 Redundant Aggregate Data Flows

In this set of experiments, we focus on the cases with redundant aggregate data flows. In our setting, redundant aggregate data flows mean that some aggregate

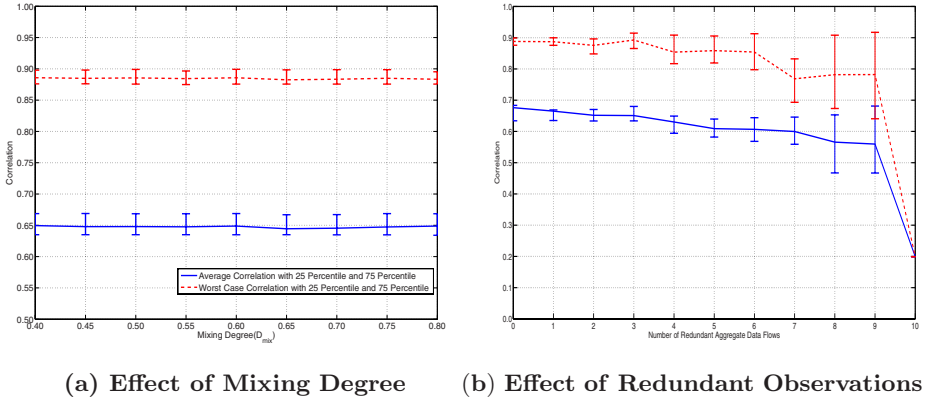


Fig. 4. Effect of Mixing Degree and Redundant Observations (Lower bar: 25 percentile, Upper bar: 75 percentile)

data flows are linear combinations of other aggregate data flows. Redundant aggregate data flows will reduce the number of effective aggregate data flows. Redundant aggregate data flows are caused by rank deficient mixing matrices.

To study the effect of redundant observations, we randomly generate 1000 mixing matrices for each possible rank. Ten data flows randomly selected from the stock data are mixed using the randomly-generated mixing metrics of different ranks.

Figure 4(b) shows the performance of data flow separation with redundant observations. We can observe that the performance of data flow separation decreases as the number of redundant observations increases. The performance degrades because the number of knowns decreases. When the number of aggregate data flows is larger than the number of individual data flows, the data flow separation problem becomes an over-complete base problem in blind source separation literature. In general an over-complete base problem is harder to solve.

6.5 Dependence between Individual Data Flows

In this set of experiments, we study the effect of dependence between individual data flows on data flow separation performance. We did this series of experiments because of the fact that most blind source separation algorithms assume relative independence between actual signals.

Groups of ten data flows are randomly picked from the stock data [24]. These groups have different average correlations among data flows in the same group. The time series in each group are mixed randomly and we apply data flow separation technique on the generated aggregates.

Figure 5 shows that the performance of data flow separation technique decreases when the dependence among individual data flows increases. It is because blind source separation algorithms used in data flow separation assume independence between underlying components. Even for the blind source separation

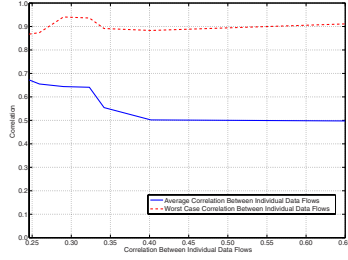


Fig. 5. Effect of Dependence among Individual Data Flows

algorithm [23] which takes advantage of both independence and timing structure of underlying signals, the dependence among individual data flows can still degrade the performance of data flow separation attack. We can also observe that worst case correlation is not sensitive to the dependence between individual data flows.

6.6 Frequency Matching

In this subsection, we show the performance of frequency matching attack proposed in Section 5.3. In this experiment, two groups of ten data flows each are formed by selecting data flows from the stock data. Three data flows in both groups are the same. These two groups of data flows are mixed randomly to form two groups of aggregate data flows. Data flow separation is performed on the two groups of aggregate data flows. We identify common flows in both groups by matching frequency spectrum of separated data flows in two different groups.

Figure 6 shows the correlation between three identified separated data flows in one group and the ten separated data flows in the other group. As shown in Figure 6, we can easily find out the data flows common to both groups.

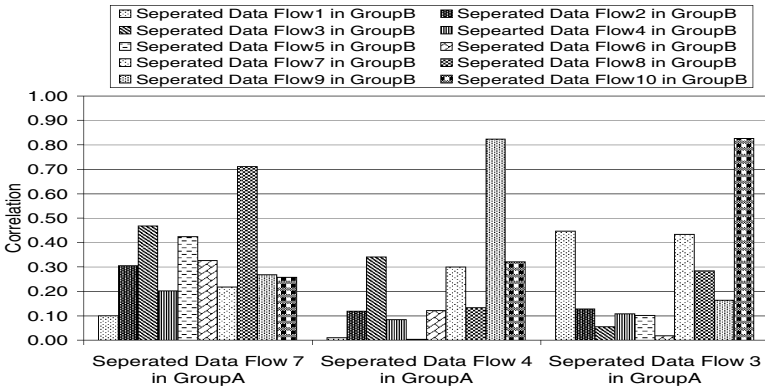


Fig. 6. Performance of Frequency Matching

7 Discussion

From the experiments in Section 6, it is apparent that aggregation methods are not sufficient to effectively counter data flow separation attacks. Additional measures are needed.

One naive countermeasure is adding different noises to a data flow to be supplied to different research companies so that research companies will receive different copies of the data flow. It may not work if the noise and the original data flow are independent, and thus can be separated by blind source separation.

According to our experiments, following countermeasures will be effective against data flow separation attacks:

- Increase the dependence among data flows by adding dependent noises to the data flows. Further research is needed to investigate how to optimally add noise so that privacy can be preserved and the performance of time series data mining will not be significantly affected.
- Limit the number aggregate data flows that can be obtained by an adversary so that the number of observations is much less than the number of independent components. This countermeasure requires cooperation among research companies and it is hard to be enforced.
- Data sources should know from research companies about how the supplied data to be aggregated and restricted the usage of supplied data.

Also, research in blind source separation shows most blind source separation algorithms fail when the signals mixed are Gaussian distributed. Therefore, another countermeasure against data flow separation attack is padding each aggregate data flow so that the distribution of the aggregated data is Gaussian.

As mentioned in [25], aggregation is a major technique used to preserve privacy in data mining. Since data flow separation attack can separate individual data flows from aggregates, aggregation technique based privacy-preserving data mining systems are potentially vulnerable to data flow separation attacks.

8 Conclusion

In this paper, we introduce the concept of privacy in time series data mining. We present a new attack against privacy in time series data mining, called data flow separation attack, which can be used either alone or in conjunctions with other attacks to significantly reduce the effectiveness of privacy-preserving techniques in data mining. Our experiments show that the attack is effective. With the aid of further attack such as frequency matching attack, data flow separation attack can be used to determine data sources of separate data flows.

We discuss countermeasures against data flow separation attack. Our future work will focus on countermeasures to balance privacy-preserving and performance of data mining.

Acknowledgment

We thank Professor Keogh for the data sets used in our experiments and anonymous reviewers for feedbacks on the initial version of this paper.

References

1. Agrawal, R., Srikant, R.: Privacy-preserving data mining. In: SIGMOD Conference, pp. 439–450 (2000)
2. Evfimievski, A.V., Srikant, R., Agrawal, R., Gehrke, J.: Privacy preserving mining of association rules. In: SIGKDD, pp. 217–228 (2002)
3. Lindell, Y., Pinkas, B.: Privacy preserving data mining. In: Bellare, M. (ed.) CRYPTO 2000. LNCS, vol. 1880, pp. 36–54. Springer, Heidelberg (2000)
4. Kantarcioglu, M., Clifton, C.: Privacy-preserving distributed mining of association rules on horizontally partitioned data. *IEEE Trans. Knowl. Data Eng.* 16(9) (2004)
5. Faloutsos, C., Ranganathan, M., Manolopoulos, Y.: Fast subsequence matching in time-series databases. In: SIGMOD Conference, pp. 419–429 (1994)
6. Das, G., Lin, K.I., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: SIGKDD, pp. 16–22 (1998)
7. Geurts, P.: Pattern extraction for time series classification. In: Siebes, A., De Raedt, L. (eds.) PKDD 2001. LNCS (LNAI), vol. 2168, pp. 115–127. Springer, Heidelberg (2001)
8. Keogh, E.J., Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. *Knowl. Inf. Syst.* 8(2), 154–177 (2005)
9. Ihler, A.T., Hutchins, J., Smyth, P.: Adaptive event detection with time-varying poisson processes. In: SIGKDD, pp. 207–216 (2006)
10. Mörchen, F., Utsch, A.: Optimizing time series discretization for knowledge discovery. In: SIGKDD, pp. 660–665 (2005)
11. Keogh, E.J., Pazzani, M.J.: An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In: SIGKDD, pp. 239–243 (1998)
12. Cole, R., Shasha, D., Zhao, X.: Fast window correlations over uncooperative time series. In: SIGKDD, pp. 743–749 (2005)
13. Jutten, C., Herault, J.: Blind separation of sources, part 1: an adaptive algorithm based on neuromimetic architecture. *Signal Process.* 24(1), 1–10 (1991)
14. Huang, Z., Du, W., Chen, B.: Deriving private information from randomized data. In: SIGMOD Conference, pp. 37–48 (2005)
15. Du, W., Atallah, M.J.: Secure multi-party computation problems and their applications: a review and open problems. In: New Security Paradigms Workshop 2001, Cloudcroft, New Mexico, USA, September 10–13, 2001, pp. 13–22 (2001)
16. Jagannathan, G., Wright, R.N.: Privacy-preserving distributed k-means clustering over arbitrarily partitioned data. In: SIGKDD, pp. 593–599 (2005)
17. Wright, R.N., Yang, Z.: Privacy-preserving bayesian network structure computation on distributed heterogeneous data. In: SIGKDD, pp. 713–718 (2004)
18. Keogh, E.J., Pazzani, M.J.: Scaling up dynamic time warping for datamining applications. In: SIGKDD, pp. 285–289 (2000)
19. Cardoso, J.: Blind signal separation: statistical principles. *Proceedings of the IEEE, Special issue on blind identification and estimation* 9(10), 2009–2025 (1998)

20. Comon, P.: Independent component analysis, a new concept? *Signal Process* 36(3), 287–314 (1994)
21. Hyvärinen, A.: Fast and robust fixed-point algorithms for independent component analysis. *IEEE Transactions on Neural Networks* 10(3), 626–634 (1999)
22. Pham, D.T., Garrat, P., Jutten, C.: Separation of a mixture of independent sources through a maximum likelihood approach. In: *Proc. EUSIPCO*, pp. 771–774 (1992)
23. Cruces-Alvarez, S.A., Cichocki, A.: Combining blind source extraction with joint approximate diagonalization: Thin algorithms for ICA. In: *Proc. of the Fourth Symposium on Independent Component Analysis and Blind Signal Separation*, Nara, Japan, April 2003, pp. 463–468 (2003)
24. Keogh, E., Xi, X., Wei, L., Ratanamahatana, C.A.: The ucr time series classification/clustering homepage (2006), http://www.cs.ucr.edu/~eamonn/time_series_data/
25. Zhang, N., Zhao, W.: Privacy-preserving data mining systems. *Computer* 40(4), 52–58 (2007)

Exploiting Propositionalization Based on Random Relational Rules for Semi-supervised Learning

Grant Anderson and Bernhard Pfahringer

Department of Computer Science, University of Waikato, Hamilton, New Zealand

Abstract. In this paper we investigate an approach to semi-supervised learning based on randomized propositionalization, which allows for applying standard propositional classification algorithms like support vector machines to multi-relational data. Randomization based on random relational rules can work both with and without a class attribute and can therefore be applied simultaneously to both the labeled and the unlabeled portion of the data present in semi-supervised learning.

An empirical investigation compares semi-supervised propositionalization to standard propositionalization using just the labeled data portion, as well as to a variant that also just uses the labeled data portion but includes the label information in an attempt to improve the resulting propositionalization. Preliminary experimental results indicate that propositionalization generated on the full dataset, i.e. the semi-supervised approach, tends to outperform the other two more standard approaches.

Keywords: semi-supervised, propositionalization, randomization.

1 Introduction

In supervised classification, training is performed on a set of examples with assigned class labels, and the resulting model is then evaluated on the accuracy of the class labels it assigns to unlabeled data. Semi-supervised classification differs from supervised classification in that additional unlabeled data is available for the algorithm to use in model construction [4]. Krogel and Scheffer [13], for example, experiment with using unlabeled data to augment experiments on the KDD Cup data, and SSVA [17] uses unlabeled data to enhance a support vector machine.

Propositional learning algorithms represent examples as single objects with values for a given set of attributes. This can make it difficult to represent relationships between objects. Relational learning employs richer concept descriptions (such as restricted forms of first-order logic, e.g. like the one used in Foil [18]) to overcome this limitation and allow those relationships to be explicitly represented and used in learning. However, this increased expressivity also causes higher computational cost, resulting in learning algorithms with an exponential time complexity.

Propositionalization [12] – the process of converting a relational representation of data into a propositional one – aims to preserve the relationships within the data while producing a representation for the data that can be used with efficient propositional classification algorithms. This paper presents a two-tiered approach to semi-supervised relational classification that allows for the application of standard propositional learning algorithms to multi-relational data. In the first stage we propositionalize the relational data using randomly generated first-order rules (similar to the relational association rules generated by WarmR [10]), which are then converted into boolean features, based on their coverage. The generation process tries to ensure that generated rules are likely to be useful for classification. This is done by requiring that rules cover a certain number of examples within user-specified minima and maxima, thus avoiding both overly specific and overly general rules. Alternatively, in a class-sensitive setting where class labels are actually present, rules can be selected based on their class-specific coverage similar to the “enrichment” property of stochastic discrimination [11]. In either setting all rules are turned into boolean attributes generating a propositional representation for the second stage, where the resulting propositional dataset is classified using any standard propositional classifier, such as SMO [16] or others.

This procedure holds promise for semi-supervised learning, as one of the main explanations for the success of semi-supervised learning is the so-called cluster assumption [4]. The unlabeled data enables better estimation of cluster boundaries and can therefore also improve classification accuracy. In [1] random relational rules have been shown to work well for the clustering of relational data. Thus their usefulness for semi-supervised learning is investigated in this paper. There does not currently appear to be any directly related work on semi-supervised propositionalisation, which means that there is no outside standard with which the experimental results in this paper could be meaningfully compared.

The next section describes the algorithms in more detail, Section 3 explains and discusses experiments and the final section draws conclusions and outlines future work.

2 Method

The RRP (Randomized Relational Propositionalization) algorithm is composed of two tiers: a first stage generates random rules and a second stage transforms these rules into Boolean features for a propositional representation, which can be used as input for a propositional classification algorithm.

RRP generates definite clauses, which comprise both predicates containing variables and so-called theory constants, as well as tests on and comparisons between these variables and theory constants. Functors and recursion are forbidden. For example, the Mutagenesis dataset comprises the following three predicates:

```
molecule(MoleculeID, Class)
atom(MoleculeID, AtomID, ElementType, QuantaType, Charge)
bond(MoleculeID, AtomID1, AtomID2, BondType)
```

A molecule is described by two parameters: a unique identifier and a class label (active or inactive). An atom is described by five parameters: the identifier of the Molecule it belongs to, a unique identifier, its element type, its quanta type, and its electrical charge. A bond is described by four parameters: the identifier of the molecule that it belongs to, the unique identifiers for the two atoms it is linking, as well as its own bond type. An example of a rule generated on that dataset is:

```
active(MolID):-
  atom(MolID,_,_,_,Charge),
  Charge >= 0.078,
  bond(MolID,_,AtomID1,BondType1),
  bond(MolID,_,AtomID2,BondType2),
  BondType1 != BondType2,
  AtomID1 = AtomID2.
```

This rule describes all compounds that contain an atom with a charge above 0.078 and two bonds of different types that both include a particular atom. Underscores are used here for clarity, to denote variables not used in this rule.

Such random rules are generated in the following way: at each stage a predicate or test is chosen uniformly at random with the following restrictions: for a predicate exactly one variable (or parameter) must already appear in the rule; all other variables are new. This ensures that clauses are linked. Tests on the other hand may not add any new variables. Tests include the usual equal and not-equal comparisons to other variables or theory constants, as well as range comparisons for numeric arguments.

To ensure that the generated rules allow for classification, constraints are imposed on the generation process. For class-blind rule generation, only rules are accepted that cover more than a user-defined minimum number of instances, and also cover less than a user-defined maximum. This prevents both overly specific and overly general rules. For class-sensitive rule generation, rules are required to be ‘enriched’, as per Kleinberg’s definition [11], where a rule is enriched for a particular class if it covers a greater proportion of examples of that class than it does of the other classes:

$$\text{A rule is enriched if } \frac{\#covered_{target}}{\#total_{target}} > \frac{\#covered_{other}}{\#total_{other}} \quad (1)$$

The above constraints operate on individual rules. In addition to this, each example should be covered by roughly the same number of rules. This constraint operates at the ruleset level. This “uniformity of coverage” is produced by generating the random rules in small batches, and then adding the most uniformity-preserving non-zero subset of each batch of rules to the current ruleset. In our experiments the batch size was set to five rules. The coverage of each instance is tracked as rules are added to the ruleset, and the subset that, when added to the current ruleset, gives the smallest standard deviation of instance coverages is determined to be most uniformity-preserving. Algorithm 1 details RRP.

Algorithm 1. Pseudocode for the RRP algorithm

```

while Number of rules in ruleset is less than the minimum do
  while Number of rules in batch is less than the minimum do
    Generate a rule
    if Rule is acceptable with regard to coverage constraints then
      Add rule to rule batch
    end if
  end while
  Calculate the most uniform subset of rules in the current rule batch
  Add those rules to the ruleset
end while
Use ruleset to generate boolean-valued propositional dataset
Apply any propositional classification algorithm

```

The final propositional dataset comprising solely boolean attributes is generated by evaluating each rule on each example in the original dataset. If an example is covered by the rule, the corresponding boolean attribute is set to true, otherwise it is set to false.

The complexity of RRP is the sum of the complexity of both stages. Usually, when using propositionalization in ILP, the propositionalization stage dominates the total complexity, and this is true for RRP as well. Even though generating a random rule is extremely fast, its coverage still has to be determined both for checking the coverage constraints and uniformity of coverage, as well as to generate the propositional dataset. In the worst case this coverage computation requires time exponential in the length of the rule [8]. The complexity of propositional classification algorithms on the contrary is generally polynomial at worst. Still, in practice we find that RRP enjoys very acceptable runtimes.

3 Experiments

An evaluation of RRP on several datasets has been conducted. Rule generation was performed using three different setups:

- Semi-supervised Class-blind - generating rules on the full dataset (labeled and unlabeled) with coverage-range as the criterion (RRP-SS, Algorithm 2)
- Standard Class-blind - generating rules only on the labeled training data, again with coverage-range as the criterion (RRP-CB, Algorithm 3)
- Standard Class-sensitive - generating rules only on labeled training data with enrichment as the criterion (RRP-CS, Algorithm 4)

The propositionalization stage of RRP-SS is the same as that of RRC, described in [1]. RRP-CB and RRP-CS differ from RRC in that the propositionalization is generated on a portion of the data and then applied to the remaining data. RRP-CS differs further in its use of enrichment instead of coverage range.

The resulting propositional data was classified as described in Algorithms [2]. [4] – using SMO [16], with the ‘complexity constant’ parameter determined by

Algorithm 2. RRP-SS process

Generate propositional representation D_p on the full dataset D ,
 using coverage-range as the coverage criterion
 Apply SMO on the labeled portion of D_p to generate a model
 Evaluate the model on the unlabeled portion of D_p

Algorithm 3. RRP-CB process

Generate a propositional representation $D1_p$ using the labeled training data $D1$,
 using coverage-range as the coverage criterion
 Apply SMO on $D1_p$ to generate a model
 Apply the rules generated from $D1$ to $D2$ to produce $D2_p$
 Evaluate the model on $D2_p$

internal ten-fold cross-validation on the training data. All experiments involved random stratified 50 : 50 splits, i.e. 50% of the data was labeled, and 50% was unlabeled. Twenty repetitions were computed for each setup to produce stable average results. Linear support vector machines were used because they proved to be efficient and effective for this type of problems, which comprise at most 2000 examples, but also 1000 attributes, as all setups generated 1000 random rules. Algorithms that are non-linear in the number of attributes (e.g. logistic regression) turned out to be less effective. Competitive alternative algorithms included Random Forests [2] and Alternating Decision Trees [7].

The following standard ILP datasets were used: Mutagenesis (with and without regression-unfriendly instances) [19], Musk1 [5], Cancer [20], and Diterpenes [6]. Mutagenesis and Cancer only had access to low-level structural information as represented by atoms and bonds; additional information such as global properties *lumo* or *logP*, or predefined functional groups were not included. They are known to improve classification accuracy significantly, thereby potentially masking the relational performance of the investigated algorithms.

For the Diterpenes dataset, as the ‘enrichment’ procedure is currently limited to two-class problems, in addition to using the full 23-class dataset with RRP-SS and RRP-CB, three additional two-class versions were generated: all pairwise combinations of the three largest classes (called 3, 52 and 54), which could be used with all three algorithms.

For RRP-SS and RRP-CB, several different ranges for rule coverage were investigated: 0.05-0.5, 0.1-0.5, 0.25-0.5 and 0.25-0.75, as well as “extreme”, which denotes a coverage range limited only by being required to cover at least two

Algorithm 4. RRP-CS process

Generate a propositional representation $D1_p$ using the labeled training data $D1$,
 using enrichment as the coverage criterion
 Apply SMO on $D1_p$ to generate a model
 Apply the rules generated from $D1$ to $D2$ to produce $D2_p$
 Evaluate the model on $D2_p$

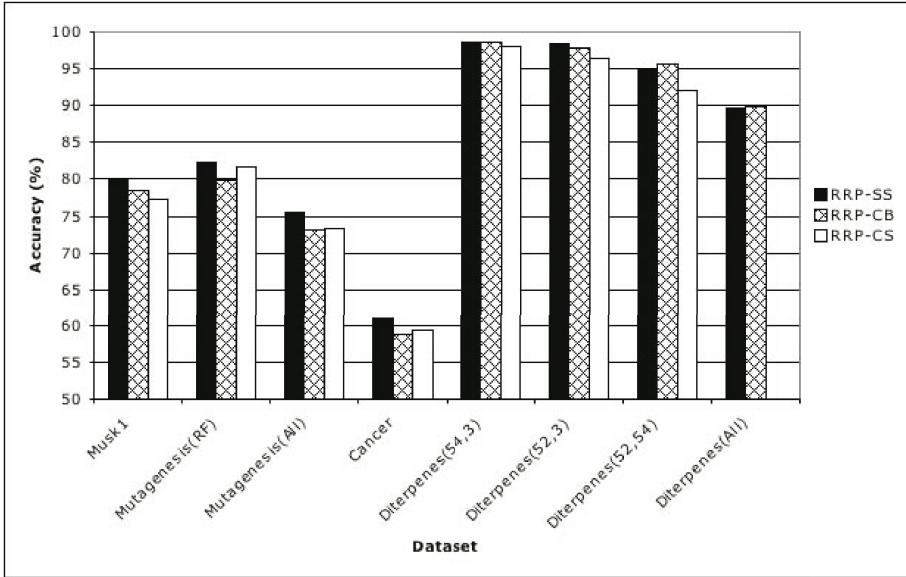


Fig. 1. Accuracy for coverage range 0.05-0.5

instances, and to not cover all instances. All numbers are proportions of the size of the training set.

The results of this evaluation for the 0.05-0.5 coverage range are shown in Figure 1. All other coverage ranges displayed similar properties, except for the “extreme” setting which performed substantially worse than the other ranges on some of the datasets, particularly the Diterpenes subsets. The poorer performance of RRP-CS relative to RRP-CB on Diterpenes is probably related to the poor performance of the “extreme” coverage setting – RRP-CS requires that rules be enriched, but places no other restrictions on their coverage, so its rules have the same coverage limits as the “extreme setting”, and it seems to produce similar results. RRP-SS enjoys a small advantage over both RRP-CB and RRP-CS for Musk1, Cancer and both versions of the Mutagenesis dataset. However, on the Diterpenes datasets, RRP-CB occasionally outperforms RRP-SS.

A possible explanation for this unexpected behavior on the Diterpenes dataset, and in particular the derived two-class subsets thereof, is the fact that they seem to be easier to classify, as the generally high accuracies obtained in classification show. This indicates that the number of labeled examples is sufficient to induce strong classifiers. Additional unlabeled data has previously been found to be either irrelevant or even detrimental under such circumstances [4]. To test this potential explanation, additional experiments were conducted with smaller numbers of labelled training examples: 10%, 4%, 2%, and 1%, and the remainder of the dataset in each case used as test examples. The results for twenty random train-test splits on Diterpenes(52,54) are depicted in Figure 2, in which the error bars show the standard deviations for the accuracy.

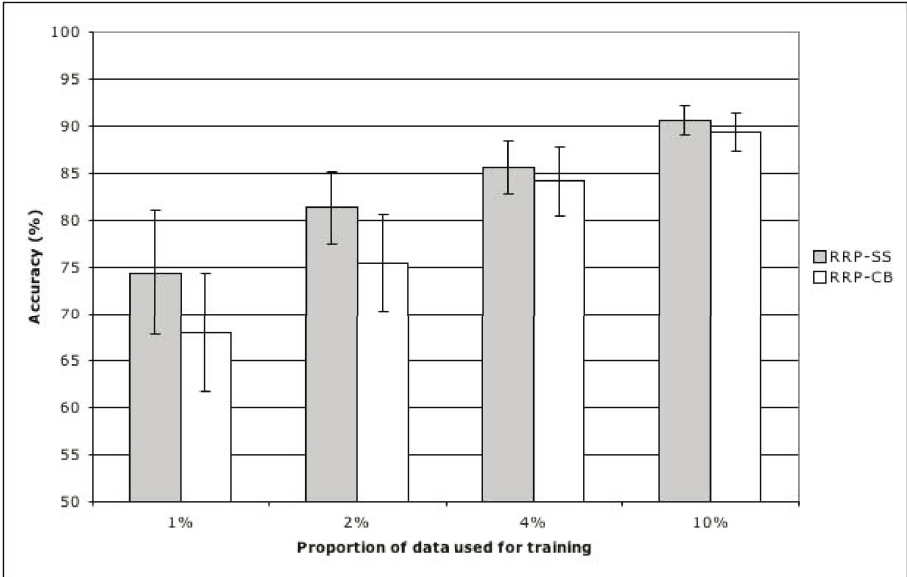


Fig. 2. Accuracy for train-test splits with varying proportions of training data

First of all it is surprising that even 1% of the data used for training (eight examples) can produce a model with 68% or even 74% accuracy, depending on the algorithm. The default accuracy for this problem is 55%. Secondly, when comparing mean accuracies, the semi-supervised approach RRP-SS enjoys the biggest advantage for the smallest number of labeled examples. Furthermore, with larger numbers of labeled data the standard algorithm RRP-CB quickly catches up with the semi-supervised variant.

4 Summary and Future Work

This paper has described a two-tiered approach to semi-supervised relational learning, based on randomized propositionalization and an arbitrary propositional classification algorithm and compared the results to standard train-test learning. The experimental results indicate that the usefulness of the extra information gained from semi-supervised learning in this case depends on the dataset, and that datasets that are already straightforward to classify with smaller proportions of labeled training instances do not benefit greatly from additional unlabeled data. On the other hand, for harder-to-classify datasets and smaller numbers of labeled data the semi-supervised approach RRP-SS enjoys a small, but consistent advantage over both standard learning algorithms RRP-CB and RRP-CS.

There are multiple avenues for future work. The propositionalization phase of RRP could be replaced by other propositionalization tools like RSD [22], or

class-blind variants of relational rule learners like Foil or Progol [14]. It should also be possible to further exploit the unlabeled data in learning: after the generation of the propositional model the mutual coverage of each boolean feature in the train and test set could be computed and compared and taken into account when inducing the propositional classifier. Preliminary experiments have shown that attributes with large differences in coverage between the training and the test set have a detrimental influence on classification accuracy. Such attributes could either be removed from the data, or down-weighted for algorithms capable of dealing with attribute weights.

One of the anonymous reviewers has suggested a intriguing fourth approach: calculating enrichment (which is class-sensitive) on the labeled data, but then determining uniformity on the labeled and unlabeled data. Future work will explore this approach. Contrary to other relational learning algorithms that strive to induce the best possible set of rules, not all random rules have to be evaluated. If rule evaluation time were to exceed a pre-specified time limit, the evaluation could be aborted and the respective rule discarded. Again, future work will explore this efficiency versus potential loss of information trade-off in more detail.

Furthermore, after propositionalization, any standard semi-supervised learning algorithm is applicable to the resulting propositional problem, i.e. the method described in this paper is orthogonal to any such method like, e.g., LLGC [23, 15]. If one were to apply standard semi-supervised learning algorithms, which usually rely on some notion of distance or similarity, directly at the relational representation instead of the propositionalization approach put forward in this paper, then relational notions of distance and similarity [21, 9] will need to be exploited. For clustering applications propositionalization has been found to outperform more direct approaches [1], but that may not be the case for semi-supervised learning, and will therefore also be explored in future work.

Acknowledgments. This work has been funded by a Marsden Grant of the Royal Society of New Zealand.

References

- [1] Anderson, G., Pfahringer, B.: Clustering Relational Data based on Randomized Propositionalization. In: Proceedings International Conference on Inductive Logic Programming 2007 (ILP 2007). Springer, Heidelberg (2007)
- [2] Breiman, L.: Random Forests. *Machine Learning* 45(1), 5–32 (2001)
- [3] de Castro Dutra, I., Page, D., Santos Costa, V., Shavlik, J.: An Empirical Evaluation of Bagging in Inductive Logic Programming. In: Matwin, S., Sammut, C. (eds.) ILP 2002. LNCS (LNAI), vol. 2583. Springer, Heidelberg (2003)
- [4] Chapelle, O., Schölkopf, B., Zien, A.: *Semi-Supervised Learning*. MIT Press, Cambridge (2006)
- [5] Dietterich, T., Lathrop, R., Lozano-Perez, T.: Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence* 89, 31–71 (1997)
- [6] Dzeroski, S., Schulze-Kremer, S., Heidtke, K.R., Siems, K., Wettschereck, D.: Applying ILP to Diterpene Structure Elucidation from ^{13}C NMR Spectra. In: Proc. 6th Int. Workshop on Inductive Logic Programming, pp. 41–54 (1996)

- [7] Freund, Y., Mason, L.: The Alternating Decision Tree Learning Algorithm. In: Proceedings of the Sixteenth International Conference on Machine Learning (ICML 1999). Morgan Kaufmann, San Francisco (1999)
- [8] Giordana, A., Saitta, L.: Phase Transitions in Relational Learning. *Machine Learning* 41(2), 217–251 (2000)
- [9] Horváth, T., Wrobel, S., Bohnbeck, U.: Relational Instance-Based Learning with Lists and Terms. *Machine Learning* 43, 53–80 (2001)
- [10] King, R.D., Srinivasan, A., Dehaspe, L.: Warmr: A Data Mining Tool for Chemical Data. *Journal of Computer Aided Molecular Design* 15, 173–181 (2001)
- [11] Kleinberg, E.M.: On the Algorithmic Implementation of Stochastic Discrimination. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22(5), 473–490 (2000)
- [12] Kramer, S., Lavrac, N., Flach, P.: Propositionalization Approaches to Relational Data Mining. *Relational Data Mining*. Springer, Heidelberg (2001)
- [13] Krogel, S., Scheffer, T.: Multi-Relational Learning, Text Mining, and Semi-Supervised Learning for Functional Genomics. *Machine Learning* 57, 61–81 (2004)
- [14] Muggleton, S.: Inverse entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming* 13(3-4), 245–286 (1995)
- [15] Pfahringer, B., Leschi, C., Reutemann, P.: Scaling Up Semi-supervised Learning: An Efficient and Effective LLGC Variant. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 236–247. Springer, Heidelberg (2007)
- [16] Platt, J.: Machines using Sequential Minimal Optimization. In: Schoelkopf, B., Burges, C., Smola, A. (eds.) *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge (1998)
- [17] Ping, L., Zhe, W., Chunguang, Z.: A Spectrum-Based Support Vector Algorithm for Relational Data Semi-supervised Classification. In: Proceedings of the 13th International Conference on Neural Information Processing, pp. 801–810 (2006)
- [18] Quinlan, J.R.: Learning logical definitions from relations. *Machine Learning* 5, 239–266 (1990)
- [19] Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.E.: Mutagenesis: ILP experiments in a non-determinate biological domain. In: Proceedings of the Fourth Inductive Logic Programming Workshop (1994)
- [20] Srinivasan, A., Muggleton, S., King, R.D., Sternberg, M.J.E.: Carcinogenesis Predictions Using ILP. In: Proceedings of the 7th International Workshop on Inductive Logic Programming (1997), pp. 273–887 (1994)
- [21] Woźnica, A., Kalousis, A., Hilario, M.: Kernels over Relational Algebra Structures. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 588–598. Springer, Heidelberg (2005)
- [22] Zelezny, F., Lavrac, N.: Propositionalization-Based Relational Subgroup Discovery with RSD. *Machine Learning* 62(1-2), 33–63 (2006)
- [23] Zhou, D., Huang, J., Schölkopf, B.: Learning from labeled and unlabeled data on a directed graph. In: Proc. of the 22nd International Conference on Machine Learning (ICML 2005), Bonn, Germany, August 2005, pp. 1041–1048 (2005)

On Discrete Data Clustering

Nizar Bouguila and Walid ElGuebaly

CIISE, Faculty of Engineering and Computer Science, Concordia University,
Montreal, Qc, Canada H3G 2W1

bouguila@ciise.concordia.ca, w_elgue@encs.concordia.ca

Abstract. Finite mixture modeling have been applied for different data mining tasks. The majority of the work done concerning finite mixture models has focused on mixtures for continuous data. However, many applications involve and generate discrete data for which discrete mixtures are better suited. In this paper, we investigate the problem of discrete data modeling using finite mixture models. We propose a novel mixture that we call the multinomial generalized Dirichlet mixture. We designed experiments involving spatial color image databases modeling and summarization to show the robustness, flexibility and merits of our approach.

1 Introduction

Discrete data appear in many machine learning and data mining applications. In this work, we are motivated by the need to construct powerful statistical approaches to model, analyze and cluster this type of data. Different statistical models have been proposed and were generally dedicated to text classification and language processing. It is well-known that the multinomial distribution performs well in the case of discrete data modeling. However, recent researches have shown that even this distribution has some drawbacks such as considering that the events to model are independent [1,2,3]. Different smoothing techniques have been proposed to overcome these problems. The most successful approach is the use of the Dirichlet distribution as a prior to the multinomial which results in a completely formal statistical model [1,3]. This is due to the fact that the Dirichlet is a conjugate prior to the multinomial. Despite this conjugacy property, the consistency of its estimates as a prior, its flexibility and its ease of use, the Dirichlet has a very restrictive negative covariance structure which makes its use as a prior in the case of positively correlated data inappropriate [4,5].

In this paper, we present a discrete finite mixture model based on both a generalization of the Dirichlet distribution and the multinomial. The choice of the generalized Dirichlet is motivated by the excellent results obtained when we have used it as a parent distribution in different pattern recognition and computer vision tasks [5]. The estimation of the parameters of our mixture model is based on the maximum likelihood estimation by invoking the expectation maximization (EM) approach. The proposed mixture model is applied to an important problem in computer vision which is the introduction of spatial constraints in color histograms. Indeed, we propose a generative model for this task. Our generative model is used for image databases categorization.

2 The Multinomial Generalized Dirichlet Mixture

Let $\mathbf{X} = (X_1, \dots, X_D)$ be a discrete vector which means that each element X_l , $l = 1, \dots, D$ in \mathbf{X} is discrete and takes on values $1, 2, \dots, V$. Then, the joint probability of \mathbf{X} is given by

$$p(\mathbf{X}|\mathbf{P}) = \prod_{d=1}^D \prod_{v=1}^V P_v^{\delta(X_d=v)} = \prod_{v=1}^V P_v^{f_v} \tag{1}$$

where $\delta(X_d = v)$ is an indicator function, $\mathbf{P} = (P_1, \dots, P_V)$ is the parameter vector, $P_v = p(X_d = v)$, $\sum_{v=1}^V P_v = 1$, and $f_v = \sum_{d=1}^D \delta(X_d = v)$. Using Eq. 1, the samples will be used to set the probabilities, obtaining

$$\hat{P}_w = \frac{f_w}{\sum_{v=1}^V f_v} \tag{2}$$

which gives poor estimate 3. Then, the majority of the researchers assign a single Dirichlet or a Dirichlet mixture prior to the parameter vector of multinomial distribution to moderate the extreme estimates given by Eq. 2 4. The Dirichlet distribution with V parameters $\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_V)$ is defined by

$$p(\mathbf{P}|\boldsymbol{\alpha}) = \frac{\Gamma(\sum_{v=1}^V \alpha_v)}{\prod_{v=1}^V \Gamma(\alpha_v)} \prod_{v=1}^V P_v^{\alpha_v-1} \tag{3}$$

The Dirichlet distribution depends on V parameters $\alpha_1, \dots, \alpha_V$, which are all real and positive. In spite of its flexibility and the fact that it is conjugate to the multinomial, the Dirichlet has a very restrictive covariance matrix 4. Another restriction of the Dirichlet distribution is that the variables with the same mean must have the same variance as shown in 6. All these disadvantages can be handled by using the generalized Dirichlet distribution. The generalized Dirichlet pdf is defined by

$$p(\mathbf{P}|\boldsymbol{\xi}) = \prod_{v=1}^{V-1} \frac{\Gamma(\alpha_v + \beta_v)}{\Gamma(\alpha_v)\Gamma(\beta_v)} P_v^{\alpha_v-1} (1 - \sum_{l=1}^v P_l)^{\gamma_v} \tag{4}$$

where $\boldsymbol{\xi} = (\alpha_1, \beta_1, \dots, \alpha_{V-1}, \beta_{V-1})$, $\alpha_v > 0$, $\beta_v > 0$, $\gamma_v = \beta_v - \alpha_{v+1} - \beta_{v+1}$ for $v = 1 \dots V-2$ and $\gamma_{V-1} = \beta_{V-1} - 1$. Note that the generalized Dirichlet distribution is reduced to a Dirichlet distribution with parameters $(\alpha_1, \dots, \alpha_{V-1}, \alpha_V = \beta_{V-1})$ when $\beta_l = \alpha_{l+1} + \beta_{l+1}$. Thus, the generalized Dirichlet includes the Dirichlet as a special case. Comparing to the Dirichlet, the generalized Dirichlet has $V - 2$ extra parameters which is a very important advantage. Indeed, as the Dirichlet has V parameters, when constructing a Dirichlet prior and if the mean probabilities of the variables have been fixed, it remains only one degree of freedom (by fixing the value of $\sum_{v=1}^V \alpha_v$) to adjust the distribution 7. For the generalized Dirichlet, however, it remains $V - 1$ degrees of freedom which makes

it more flexible for several applications [5]. The mean of the generalized Dirichlet distribution is given by [8]

$$E(P_v) = \frac{\alpha_v}{\alpha_v + \beta_v} \prod_{k=1}^{v-1} \frac{\beta_k}{\alpha_k + \beta_k} \tag{5}$$

In addition to this property, the generalized Dirichlet is conjugate to the multinomial distribution and we can easily show that the joint distribution of \mathbf{X} and \mathbf{P} is

$$p(\mathbf{X}, \mathbf{P}|\boldsymbol{\xi}) = \prod_{v=1}^{V-1} \frac{\Gamma(\alpha_v + \beta_v)}{\Gamma(\alpha_v)\Gamma(\beta_v)} P_v^{\alpha'_v - 1} (1 - \sum_{l=1}^v P_l)^{\gamma'_v} \tag{6}$$

where $\alpha'_v = \alpha_v + f_v$ and $\beta'_v = \beta_v + f_{v+1} + \dots + f_V$ for $v = 1, \dots, V - 1$, $\gamma'_v = \beta'_v - \alpha'_{v+1} - \beta'_{v+1}$ for $v = 1, \dots, V - 2$ and $\gamma'_{V-1} = \beta'_{V-1} - 1$. Integrating over \mathbf{P} , we obtain the marginal distribution of \mathbf{X}

$$p(\mathbf{X}|\boldsymbol{\xi}) = \int_{\mathbf{P}} p(\mathbf{X}, \mathbf{P}|\boldsymbol{\xi}) d\mathbf{P} = \prod_{v=1}^{V-1} \frac{\Gamma(\alpha_v + \beta_v)}{\Gamma(\alpha_v)\Gamma(\beta_v)} \prod_{v=1}^{V-1} \frac{\Gamma(\alpha'_v)\Gamma(\beta'_v)}{\Gamma(\alpha'_v + \beta'_v)}$$

We call this density the multinomial generalized Dirichlet distribution (MGD). Then, the posterior is given by

$$p(\mathbf{P}|\mathbf{X}, \boldsymbol{\xi}) = \frac{p(\mathbf{X}, \mathbf{P}|\boldsymbol{\xi})}{p(\mathbf{X}|\boldsymbol{\xi})} = \prod_{v=1}^{V-1} \frac{\Gamma(\alpha'_v + \beta'_v)}{\Gamma(\alpha'_v)\Gamma(\beta'_v)} P_v^{\alpha'_v - 1} (1 - \sum_{l=1}^v P_l)^{\gamma'_v} \tag{7}$$

which is a generalized Dirichlet with parameters $(\alpha'_1, \beta'_1, \dots, \alpha'_{V-1}, \beta'_{V-1})$. Then, by taking the generalized Dirichlet as a prior to the multinomial and according to Eq [5] and Eq [7], we obtain

$$\hat{P}_w = E[P_w|\mathbf{X}; \boldsymbol{\xi}] = \frac{\alpha_w + f_w}{\alpha_w + \beta_w + n_w} \prod_{l=1}^{w-1} \frac{\beta_l + n_{l+1}}{\alpha_l + \beta_l + n_l} \tag{8}$$

where $n_l = f_l + f_{l+1} + \dots + f_V$. When $\beta_l = \alpha_{l+1} + \beta_{l+1}$, it is straightforward to verify that this equation is reduced to

$$\hat{P}_w = E[P_w|\mathbf{X}; \boldsymbol{\xi}] = \frac{\alpha_w + f_w}{\sum_{v=1}^V (\alpha_v + f_v)} = \frac{f_w + \alpha_w}{D + \sum_{v=1}^V \alpha_v} \tag{9}$$

where $\alpha_V = \beta_{V-1}$, which represents the expectation when we consider a Dirichlet distribution, with parameters $(\alpha_1, \dots, \alpha_V)$, as a prior.

Suppose now that we select a generalized Dirichlet mixture as a prior to the multinomial. A generalized Dirichlet mixture with M components is defined as

$$p(\mathbf{P}|\Theta) = \sum_{j=1}^M p(\mathbf{P}|\boldsymbol{\xi}_j) p_j \tag{10}$$

where p_j ($0 < p_j \leq 1$ and $\sum_{j=1}^M p_j = 1$) are the mixing proportions and $p(\mathbf{P}|\xi_j)$ is the generalized Dirichlet. The symbol $\Theta = (\xi_1, \dots, \xi_M, p_1, \dots, p_M)$ refers to the entire set of parameters to be estimated. With a mixture prior, the marginal distribution of \mathbf{X} is

$$p(\mathbf{X}|\Theta) = \int_{\mathbf{P}} p(\mathbf{X}, \mathbf{P}|\Theta) d\mathbf{P} = \sum_{j=1}^M p_j \prod_{v=1}^{V-1} \frac{\Gamma(\alpha_{jv} + \beta_{jv})}{\Gamma(\alpha_{jv})\Gamma(\beta_{jv})} \prod_{v=1}^{V-1} \frac{\Gamma(\alpha'_{jv})\Gamma(\beta'_{jv})}{\Gamma(\alpha'_{jv} + \beta'_{jv})}$$

We call this density the multinomial generalized Dirichlet mixture (MGDM). And we can easily show that

$$\hat{P}_w = E[P_w|\mathbf{X}; \Theta] = \sum_{j=1}^M p(j|\mathbf{X}; \xi_j) \frac{\alpha'_{jw}}{\alpha'_{jw} + \beta'_{jw}} \prod_{k=1}^{w-1} \frac{\beta'_{jk}}{\alpha'_{jk} + \beta'_{jk}}$$

where $p(j|\mathbf{X}; \xi_j) = \frac{p_j p(\mathbf{X}|\xi_j)}{\sum_{j=1}^M p_j p(\mathbf{X}|\xi_j)}$ and represents the posterior probability.

3 Maximum Likelihood Estimation

Given a set of independent vectors $\mathcal{X} = \{\mathbf{X}_1, \dots, \mathbf{X}_N\}$, the log-likelihood corresponding to an M -component MGDM is given by

$$L(\mathcal{X}, \Theta) = \sum_{i=1}^N \log \left(\sum_{j=1}^M p_j p(\mathbf{X}_i|\xi_j) \right) \tag{11}$$

The maximization defining the ML estimates is subject to the constraints $0 < p_j \leq 1$ and $\sum_{j=1}^M p_j = 1$. To obtain the ML estimates of the mixture parameters we have used the EM algorithm [9]. In EM, the ‘‘complete’’ data are considered to be $Y_i = \{\mathbf{X}_i, \mathbf{Z}_i\}$, where $\mathbf{Z}_i = (Z_{i1}, \dots, Z_{iM})$ with

$$Z_{ij} = \begin{cases} 1 & \text{if } \mathbf{X}_i \text{ belongs to class } j \\ 0 & \text{otherwise} \end{cases} \tag{12}$$

constituting the ‘‘missing’’ data. The EM algorithm is based on the Q-function (the conditional expectation) of the complete-data log-likelihood

$$Q(\Theta; \Theta^{(t)}) = \sum_{i=1}^N \sum_{j=1}^M \hat{Z}_{ij} \log(p_j) + \sum_{i=1}^N \sum_{j=1}^M \hat{Z}_{ij} \log \left(\prod_{v=1}^{V-1} \frac{\Gamma(\alpha_{jv} + \beta_{jv})}{\Gamma(\alpha_{jv})\Gamma(\beta_{jv})} \prod_{v=1}^{V-1} \frac{\Gamma(\alpha_{jv} + f_{iv})\Gamma(\beta_{jv} + n_{iv+1})}{\Gamma(\alpha_{jv} + \beta_{jv} + n_{iv})} \right) \tag{13}$$

where $\Theta^{(t)}$ is the value of Θ at iteration t and

$$\hat{Z}_{ij} = p(Z_{ij} = 1|\mathbf{X}_i; \Theta^{(t)}) = \frac{p_j^{(t)} p(\mathbf{X}_i|\xi_j^{(t)})}{\sum_{j=1}^M p_j^{(t)} p(\mathbf{X}_i|\xi_j^{(t)})} \tag{14}$$

The first term in Eq. 13 can be maximized by updating p_j as following

$$p_j^{(t+1)} = \frac{\sum_{i=1}^N \hat{Z}_{ij}^{(t)}}{N} \tag{15}$$

The maximization of the second term, however, does not yield to a closed form solution. Thus, we have used Newton-Raphson method which is based on the computation of the first and second derivatives. The iterative scheme of the Newton-Raphson method is given by the following equation:

$$\xi_j^{(t)} = \xi_j^{(t-1)} - H(\xi_j^{(t-1)})^{-1} \frac{\partial Q(\theta; \theta^{(t-1)})}{\partial \xi_j^{(t-1)}} \tag{16}$$

where $H(\xi_j^{(t-1)})$ is the hessian matrix.

4 Experimental Results

Color histograms are widely used as features vectors for images summarization and retrieval [10] and are used in different systems. This can be explained by the fact that histograms provide a stable object recognition in the presence of occlusions and over views change [10]. However, histograms do not include any spatial information which is an important issue in human visual perception. Different approaches have been proposed to integrate spatial information with color histograms [11][12]. In the following, we propose a statistical model based on the MGDM to introduce the spatial information into color histograms. The proposed model is then applied to images databases summarization.

Suppose that we have N labeled images $\mathcal{I}_i, i = 1, \dots, N$ classified in R classes and that the number of labeled images in each class r is equal to n_r ($\sum_{r=1}^R n_r = N$). By associating a distribution and a weight to each class in the training set, we can suppose that each image \mathcal{I}_i is generated by a mixture of R distributions with parameters $\boldsymbol{\pi} = (\boldsymbol{\pi}_1, \dots, \boldsymbol{\pi}_R): p(\mathcal{I}_i | \boldsymbol{\pi}) = \sum_{r=1}^R p(r) p(\mathcal{I}_i | \boldsymbol{\pi}_r)$. The problem now is the determination of $p(\mathcal{I}_i | \boldsymbol{\pi}_r)$. For this, let us introduce some notations. An $L \times K$ image \mathcal{I}_i is considered to be a set of pixels $\{X_{i_{lk}}, l = 1, \dots, L; k = 1, \dots, K\}$, where $X_{i_{lk}}$ is the pixel in position (l, k) of image \mathcal{I}_i . The colors in \mathcal{I}_i are quantized into C colors c_1, \dots, c_C . The distribution $p(\mathcal{I}_i | \boldsymbol{\pi}_r)$ can be described in terms of the features of the image. In our case, the features are the pixels. In order to introduce the spatial information, the probability of a pixel should be conditioned on its neighborhood. By taking the neighborhood consisting of the pixels at a distance $d \in D = \{d_1, \dots, d_D\}$ measured using the L_∞ norm, $p(\mathcal{I}_i | \boldsymbol{\pi}_r)$ will be given by

$$p(\mathcal{I}_i | \boldsymbol{\pi}_r) = \prod_{d=1}^D \prod_{l=1}^L \prod_{k=1}^K p(X_{i_{lk}} | \boldsymbol{\pi}_r; X_{i_{l'k'}}, d) \tag{17}$$

where $|(l, k) - (l', k')| = \max\{|l - l'|, |k - k'|\} = d$. Note that Eq. 17 will represent the classic image histogram, if we suppose that each pixel $X_{i_{lk}}$ is independent of its neighborhood, which is actually the standard naive Bayes assumption.

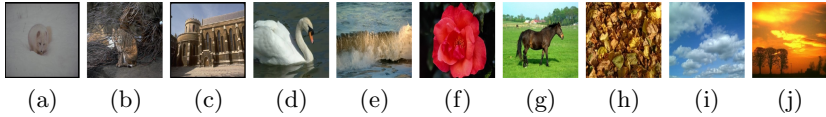


Fig. 1. Sample images from each group. (a) Class1, (b) Class2, (c) Class3, (d) Class4, (e) Class5, (f) Class6, (g) Class7, (h) Class8, (i) Class9, (j) Class10.

According to Eq. 17 the parameters of an individual mixture component are a multinomial distribution over the $C \times C$ possible color pairs and can be written as $\pi_{c_{t_1}, c_{t_2}, d|r}$, where $t_1, t_2 = 1, \dots, C$ and $\pi_{c_{t_1}, c_{t_2}, d|r} = p\left(X_{i_{lk}} = c_{t_1}, X_{i_{l'k'}} = c_{t_2} \mid |(l, k) - (l', k')| = d\right)$, $l, l' = 1, \dots, L$, $k, k' = 1, \dots, K$, which is the probability that a pixel of color c_{t_1} has at a distance d a pixel of color c_{t_2} . Then, Eq. 17 could be written as follows $p(\mathcal{I}_i | \boldsymbol{\pi}_r) = \prod_{d=1}^D \prod_{c_{t_1}=1}^C \prod_{c_{t_2}=1}^C \pi_{c_{t_1}, c_{t_2}, d|r}^{f_{c_{t_1}, c_{t_2}, d}}$, where $f_{c_{t_1}, c_{t_2}, d} \equiv \text{Card}\{(X_{i_{lk}}, X_{i_{l'k'}}) = (c_{t_1}, c_{t_2}) \mid |(l, k) - (l', k')| = d\}$, where $\text{Card}\{\}$ refers to the number of elements of a set. Learning our model consists of estimating the parameters $\pi_{c_{t_1}, c_{t_2}, d|r}$ using the n_r labeled images in class r . By noting that we can associate a C^2 -dimensional vector of frequencies $\boldsymbol{f}_{i,d} = (f_{c_1, c_1, d}, \dots, f_{c_1, c_C, d}, \dots, f_{c_C, c_1, d}, \dots, f_{c_C, c_C, d})$ to each image \mathcal{I}_i for each distance d , the parameters are estimated using Eq. 18.

For our experiments, we used a database containing 45100 images. This database contains 10 homogeneous classes (see Figure 1). We divided the database on two sets. A data set containing 22550 images used for training. The remaining images were used for testing. We considered the RGB space with color quantization into 512 colors ($8 \times 8 \times 8$) and the set of distances $D = \{1, 3, 5, 7, 9, 11\}$. Besides, we have considered only probabilities of pixels having same colors in order to reduce zero frequencies, which is a common approach and used, for instance, in the case of the autocorrelogram proposed by Huang et al. [12].

Table 1. Confusion matrix for image classification using spatial color information modeled by MGDM

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	1961	89	21	45	28	22	14	27	20	23
C2	32	2168	21	89	17	33	45	37	22	36
C3	18	71	2632	23	37	22	28	31	84	54
C4	51	35	14	1700	26	12	15	18	19	10
C5	29	17	21	52	1775	13	14	18	37	24
C6	13	25	7	10	10	1925	31	49	5	25
C7	17	55	16	19	20	23	2033	27	6	34
C8	23	36	4	8	12	25	17	2047	8	20
C9	23	7	3	15	25	2	4	3	1949	19
C10	9	21	9	5	13	12	10	32	18	2171

Table 2. Confusion matrix for image classification using spatial color information modeled by MDM

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	1739	123	43	81	49	41	29	53	41	51
C2	49	1918	49	145	47	51	86	62	39	54
C3	37	118	2347	48	67	37	56	60	135	95
C4	89	67	26	1575	37	17	21	23	28	17
C5	48	29	36	80	1674	18	19	21	46	29
C6	17	39	13	17	19	1822	48	66	16	43
C7	30	79	23	31	29	36	1927	45	8	42
C8	41	57	7	13	23	46	29	1932	15	37
C9	44	11	7	27	46	5	9	15	1859	27
C10	20	43	21	12	25	25	21	59	28	2046

Table 3. Confusion matrix for image classification using spatial color information modeled by multinomial mixtures

	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
C1	1670	131	49	86	55	44	36	61	53	65
C2	53	1858	54	169	51	58	86	67	47	57
C3	43	132	2247	53	76	45	67	72	153	112
C4	98	72	29	1515	45	23	25	29	37	27
C5	51	31	39	89	1624	21	29	25	57	34
C6	26	44	23	25	24	1742	61	78	23	54
C7	37	91	36	42	41	49	1819	63	16	56
C8	44	61	12	17	28	52	35	1877	23	51
C9	56	14	11	31	46	13	14	18	1812	35
C10	27	51	27	17	36	37	29	71	49	1956

accuracy classification produced by our classifier was measured by counting the number of misclassified images, yielding a confusion matrix. In this confusion matrix, the cell (i, j) represents the number of images from category i which are classified as category j . The number of images misclassified when we used MGDM, was 2189, which represents an accuracy of 90.29 percent (See Table 1). Table 2 represents the confusion matrix when we use Multinomial Dirichlet mixtures (MDM) (3711 misclassified images which represents an accuracy of 83.54 percent). Table 3 shows the confusion matrix when we use multinomial mixtures. In this case, the accuracy was 80.35 percent (4430 misclassified images). Note that the improvement achieved by the MGDM, comparing to MDM and multinomial mixtures, is highly statistically significant.

5 Conclusion

We have proposed, discussed and evaluated a novel finite mixture to model discrete data. This mixture model is based on both the generalized Dirichlet

and the multinomial distributions. The recently proposed multinomial Dirichlet mixture has turned out to be a special case. The proposed model is powerful and flexible enough to be adapted to a broad variety of applications where discrete data play an important role such as information retrieval and filtering, natural language processing and bioinformatics.

Acknowledgment

The completion of this research was made possible thanks to the Natural Sciences and Engineering Research Council of Canada (NSERC), a NATEQ Nouveaux Chercheurs Grant, and a start-up grant from Concordia University.

References

1. Bouguila, N., Ziou, D.: Unsupervised Learning of a Finite Discrete Mixture: Applications to Texture Modeling and Image Databases Summarization. *Journal of Visual Communication and Image Representation* 18(4), 295–309 (2007)
2. Rennie, J.D.M., Shih, L., Teevan, J., Karger, D.R.: Tackling the Poor Assumptions of Naive Bayes Text Classifiers. In: *Proc. of the Twentieth International Conference on Machine Learning (ICML 2003)*, pp. 616–623 (2003)
3. Madsen, R.E., Kauchak, D., Elkan, C.: Modeling Word Buristness Using the Dirichlet Distribution. In: *Proc. of the 22nd International Conference on Machine Learning (ICML 2005)*, pp. 545–552. ACM Press, New York (2005)
4. Bouguila, N., Ziou, D., Vaillancourt, J.: Unsupervised Learning of a Finite Mixture Model Based on the Dirichlet Distribution and its Application. *IEEE Transactions on Image Processing* 13(11), 1533–1543 (2004)
5. Bouguila, N., Ziou, D.: A Hybrid SEM Algorithm for High-Dimensional Unsupervised Learning Using a Finite Generalized Dirichlet Mixture. *IEEE Transactions on Image Processing* 15(9), 2657–2668 (2006)
6. Lochner, R.H.: A Generalized Dirichlet Distribution in Bayesian Life Testing. *Journal of the Royal Statistical Society, B* 37, 103–113 (1975)
7. Thall, P.F., Sung, H.G.: Some Extensions and Applications of a Bayesian Strategy for Monitoring Multiple Outcomes in Clinical Trials. *Statistics in Medicine* 17, 1563–1580 (1998)
8. Wong, T.: Generalized Dirichlet Distribution in Bayesian Analysis. *Applied Mathematics and Computation* 97, 165–181 (1998)
9. McLachlan, G.J., Krishnan, T.: *The EM Algorithm and Extensions*. Wiley-Interscience, New York (1997)
10. Swain, M., Ballard, D.: Color Indexing. *International Journal of Computer Vision* 7(1), 11–32 (1991)
11. Bouguila, N.: Spatial Color Image Databases Summarization. In: *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2007)*, Honolulu, HI, USA, vol. 1, pp. I-953–I-956 (2007)
12. Huang, J., Kumar, S.R., Mitra, M., Zhu, W., Zabih, R.: Spatial Color Indexing and Applications. *International Journal of Computer Vision* 35(3), 245–268 (1999)

Automatic Training Example Selection for Scalable Unsupervised Record Linkage

Peter Christen

Department of Computer Science, The Australian National University,
Canberra ACT 0200, Australia
`peter.christen@anu.edu.au`

Abstract. Linking records from two or more databases is an increasingly important data preparation step in many data mining projects, as linked data can enable studies that are not feasible otherwise, or that would require expensive collection of specific data. The aim of such linkages is to match all records that refer to the same entity. One of the main challenges in record linkage is the accurate classification of record pairs into matches and non-matches. Many modern classification techniques are based on supervised machine learning and thus require training data, which is often not available in real world situations. A novel two-step approach to unsupervised record pair classification is presented in this paper. In the first step, training examples are selected automatically, and they are then used in the second step to train a binary classifier. An experimental evaluation shows that this approach can outperform k -means clustering and also be much faster than other classification techniques.

Keywords: data linkage, entity resolution, clustering, support vector machines, data mining preprocessing.

1 Introduction

With massive amounts of data being collected by many businesses and government agencies, techniques that enable efficient sharing of large databases between organisations are of increasing importance in many data mining projects. Data from various sources often has to be linked in order to improve data quality and integrity, or to enrich existing data with additional information [12]. Linking entities is often challenged by the lack of entity identifiers, and thus sophisticated linkage techniques, using the available record attributes, are required [7].

For large databases, it is not feasible to compare each record from one database with all records from another database, as this process is computationally too expensive [7]. Blocking techniques are employed to reduce the number of record pair comparisons [1]. They group records into blocks, and candidate record pairs are then generated only from the records within the same block. Assuming there are no duplicate records in the databases to be linked, then the majority of candidate pairs are non-matches, as the maximum possible number of true matches corresponds to the number of records in the smaller of the databases. Classifying record pairs is thus often a very imbalanced problem.

	Name		Address		
<i>R1</i> :	Christine	Smith	42	Main	Street
<i>R2</i> :	Christina	Smith	42	Main	St
<i>R3</i> :	Bob	O'Brian	11	Smith	Rd
<i>R4</i> :	Robert	Bryce	12	Smythe	Road

$WV(R1,R2)$: [0.9, 1.0, 1.0, 1.0, 0.9]
 $WV(R1,R3)$: [0.0, 0.0, 0.0, 0.0, 0.0]
 $WV(R1,R4)$: [0.0, 0.0, 0.5, 0.0, 0.0]
 $WV(R2,R3)$: [0.0, 0.0, 0.0, 0.0, 0.0]
 $WV(R2,R4)$: [0.0, 0.0, 0.5, 0.0, 0.0]
 $WV(R3,R4)$: [0.7, 0.3, 0.5, 0.7, 0.9]

Fig. 1. The left side shows four example records and the right side the corresponding weight vectors resulting from their comparisons (based on Fig. 2 from [5])

Candidate record pairs are compared using similarity functions applied to selected record attributes. These functions can be as simple as an exact string or a numerical comparison, can take typographical variations into account [2], or they can be specialised, for example, for date or time values. Each similarity function returns a numerical value, often normalised such that 1.0 corresponds to exact similarity and 0.0 to total dissimilarity. For each compared record pair a matching weight vector is formed that contains the pair’s matching weights. Using these weight vectors, record pairs are then classified into matches, non-matches, and uncertain pairs, depending upon the decision model used [7].

A record pair that has equal or very similar attribute values will likely refer to the same entity, as it is very unlikely that two entities have very similar or even the same values in all their record attributes. The matching weights calculated when comparing such a pair will be 1 (or close to 1) in all weight vector elements. On the other hand, weight vectors that contain matching weights of only 0 (or values close to 0) in all vector elements were with high likelihood calculated when two different entities were compared, as it is highly unlikely that two records that refer to the same entity have different values in all their attributes.

Based on these observations, it is normally easy to accurately classify a candidate record pair as a match when its weight vector contains only matching weights close to or equal to 1, and as a non-match when its weights are all close to or equal to 0. It is however much more difficult to correctly classify a pair that has some similar and some dissimilar attribute values. In the examples shown in Fig. 1, records *R1* and *R2* are very similar to each other, and thus very likely refer to the same person. On the other hand, *R1* and *R4* are more different from each other, and it is not obvious if they refer to the same person.

It follows that it is possible, in a first step, to automatically select weight vectors that correspond to good quality training examples. For example, of the weight vectors shown in Fig. 1, $(0.9, 1.0, 1.0, 1.0, 0.9)$ can be selected as a match training example, and $(0.0, 0.0, 0.0, 0.0, 0.0)$, $(0.0, 0.0, 0.5, 0.0, 0.0)$ and $(0.7, 0.3, 0.5, 0.7, 0.9)$ as non-match examples. These training examples can then be used in a second step to train a binary classifier for classification of all weight vectors.

This two-step approach to record pair classification has first been proposed by the author in [5], with initial experiments indicating its feasibility. The contribution of this paper is the investigation of a potential improvement to the basic approach, namely to randomly include additional weight vectors for training.

2 Related Work

In recent years, various techniques have been explored for record pair classification [7]. Among the supervised learning techniques used are decision trees [8,11] and support vector machines [10], while another approach is adaptive string similarity functions [2]. While supervised techniques normally achieve better linkage quality than unsupervised ones, their major drawback is the lack of training data (record pairs with known true match and non-match status) in many real world situations, as manual preparation of training data is time consuming and expensive. Active learning aims to overcome this problem through manual classification of only the most difficult record pairs to classify automatically [11].

Three classification approaches were compared in [8]: decision trees; k -means with three clusters (matches, possible matches and non-matches); and a hybrid approach that first clusters a sub-set of weight vectors (again into three clusters), and then uses the match and non-match clusters for decision tree induction learning. The supervised and hybrid approaches both outperformed k -means.

Methods similar to the proposed two-step approach have been developed for text and Web page classification [9,13], where often only a small number of positive training examples is available besides many unlabeled documents. The aim is to learn a binary classifier from positive and unlabeled examples. PEBL [13] iteratively trains a support vector machine (SVM) using the positive and negative documents furthest away from the decision boundary, while the S-EM [9] approach includes ‘spy’ documents, positive labeled examples, into the set of unlabeled documents to get a more realistic model of their distribution to be used in the EM algorithm. This is similar to the idea of randomly including additional weight vectors into the training sets as presented in this paper.

3 Two-Step Classification

In the first step of the proposed approach, weight vectors that with high likelihood correspond to true matches and true non-matches are selected as training examples. In the second step, these training examples are used to train a binary classifier, which is then employed to classify all weight vectors into matches and non-matches.

3.1 Training Example Selection

There are two different approaches on how to select training examples: threshold or nearest based [5]. In the first approach, weight vectors that have all their vector elements within a certain distance threshold to the exact similarity or total dissimilarity values, respectively, will be selected. For example, using the weight vectors from Fig. 1 and a threshold of 0.2, only $(0.1, 0.1, 0.1)$ will be selected as match training example, and $(0.3, 0.3, 0.3)$ and $(0.4, 0.4, 0.4)$ as non-match training examples. The remaining three weight vectors will not be selected, as at least one of their vector elements is larger than the 0.2 distance threshold.

The second approach is to sort weight vectors according to their distances from the vectors containing only exact similarities and only total dissimilarities, respectively, and to then select the respectively nearest vectors. In Fig. 1, vector $(0.9, 0.9, 0.9)$ is closest to the exact similarities vector, followed by $(0.9, 0.9, 0.8)$. Vectors $(0.9, 0.9, 0.8)$ and $(0.9, 0.8, 0.8)$ only contain total dissimilarity values, and $(0.9, 0.8, 0.9)$ and $(0.8, 0.9, 0.9)$ are the next vectors closest to them.

The notation used below is as follows. It is assumed that candidate record pairs are compared using d similarity functions (with $d \geq 1$), resulting in a set \mathbf{W} of weight vectors \mathbf{w}_i ($1 \leq i \leq |\mathbf{W}|$) of length d , with $|\cdot|$ denoting the number of elements in a set. All comparison functions are assumed to return normalised matching weights between 0 (total dissimilarity) and 1 (exact similarity), i.e. $0 \leq \mathbf{w}_i[j] \leq 1, 1 \leq j \leq d, \forall \mathbf{w}_i \in \mathbf{W}$. The weight vector containing exact similarities in all vector elements (i.e. corresponding to an exact match) is denoted by \mathbf{m} (with $\mathbf{m}[j] = 1, 1 \leq j \leq d$), and the vector with only dissimilarities by \mathbf{n} (with $\mathbf{n}[j] = 0, 1 \leq j \leq d$). In the training example selection step, weight vectors from \mathbf{W} will be inserted into the match training example set, \mathbf{W}_M , and the non-match training example set, \mathbf{W}_N . Generally, not all weight vectors from \mathbf{W} will be selected for training, thus it is likely that $(|\mathbf{W}_M| + |\mathbf{W}_N|) < |\mathbf{W}|$ holds.

Threshold-based Selection. One distance threshold for matches, t_M , and one for non-matches, t_N (with $0 < t_M, t_N < 1$), are used to select weight vectors that have all their similarity values either within t_M of the exact match value \mathbf{m} , or within t_N of the total dissimilarity value \mathbf{n} . Formally, weight vectors from \mathbf{W} will be inserted into \mathbf{W}_M and \mathbf{W}_N , according to: $\mathbf{W}_M = \{\mathbf{w}_i \in \mathbf{W} : (\mathbf{m}[j] - \mathbf{w}_i[j]) \leq t_M, 1 \leq j \leq d\}$, and $\mathbf{W}_N = \{\mathbf{w}_i \in \mathbf{W} : (\mathbf{n}[j] + \mathbf{w}_i[j]) \leq t_N, 1 \leq j \leq d\}$.

Nearest-based Selection. In this approach, the x_M weight vectors closest to \mathbf{m} are selected into \mathbf{W}_M , and the x_N weight vectors closest to \mathbf{n} are selected into \mathbf{W}_N . Both $x_M > 0$ and $x_N > 0$ must hold. The training sets \mathbf{W}_M and \mathbf{W}_N are formed according to: $\mathbf{W}_M = \{\mathbf{w}_i \in \mathbf{W}, \mathbf{w}_k \notin \mathbf{W}_M : dist(\mathbf{m}, \mathbf{w}_i) < dist(\mathbf{m}, \mathbf{w}_k)\}$, and $\mathbf{W}_N = \{\mathbf{w}_i \in \mathbf{W}, \mathbf{w}_k \notin \mathbf{W}_N : dist(\mathbf{w}_i, \mathbf{n}) < dist(\mathbf{w}_k, \mathbf{n})\}$, with $dist$ being a distance function (like Euclidean distance), $x_M = |\mathbf{W}_M|$, and $x_N = |\mathbf{W}_N|$.

Given the number of true non-matches in \mathbf{W} is often much larger than the number of true matches [7], more weight vectors are selected into \mathbf{W}_N than into \mathbf{W}_M . An estimation of the ratio r of matches to non-matches is calculated using the number of records in the two data sets to be linked, \mathbf{A} and \mathbf{B} , and the total number of weight vectors $|\mathbf{W}|$: $r = min(|\mathbf{A}|, |\mathbf{B}|) / (|\mathbf{W}| - min(|\mathbf{A}|, |\mathbf{B}|))$.

3.2 Random Inclusion of Additional Training Examples

The training examples selected in the first step are likely linearly separable, because \mathbf{W}_M and \mathbf{W}_N only contain weight vectors that are either close to \mathbf{m} or close to \mathbf{n} , and also because usually not all weight vectors from \mathbf{W} are selected for training. This will likely result in a ‘gap’ between the training sets, as illustrated in Fig. 2 (a). Similar to the inclusion of ‘spy’ documents for semi-supervised text classification [9], adding a small number of randomly selected weight vectors from this ‘gap’ into \mathbf{W}_M and \mathbf{W}_N should improve classification accuracy, as

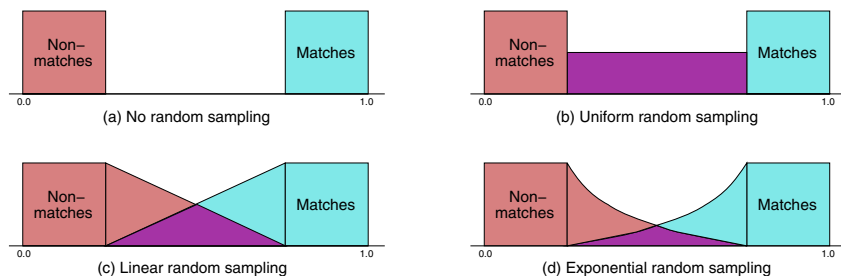


Fig. 2. Possible methods for random sampling of additional weight vectors (assumed to be 1-dimensional vectors)

the training sets will then contain a more realistic distribution of weight vectors. The random sampling of weight vectors should be done such that vectors closer to \mathbf{m} are more likely included into \mathbf{W}_M , while vectors closer to \mathbf{n} are more likely selected into \mathbf{W}_N . Besides no random sampling, the three different sampling methods illustrated in parts (b) to (d) of Fig. 2 are to use either uniform sampling, or a linear or exponential mapping function to randomly sample weight vectors. These sampling methods will be evaluated experimentally below.

3.3 Weight Vector Classification

In the second step of the proposed record pair classification approach, the training sets \mathbf{W}_M and \mathbf{W}_N , as generated in the first step, will be used to train a binary classifier. Once trained, this classifier is then employed to classify all weight vectors in \mathbf{W} . In the experiments presented below, a SVM classifier [3] will be evaluated, because this technique is known to be robust to noisy data.

4 Experimental Evaluation

The proposed two-step approach will be compared with three other classification methods. The first is an ‘optimal threshold’ classifier that has access to the true match status of all weight vectors in \mathbf{W} and can thus find a classification threshold that minimises both false matches and false non-matches. The second is a supervised SVM which also has access to the true match status of all weight vectors. Nine SVM variations were evaluated, three kernels (linear, polynomial and RBF) and three values for the cost parameter, C [3]. The third method is k -means, with the weight vectors being clustered into matches and non-matches. Three distance measures (Manhattan, Euclidean and L_{inf}) were evaluated. All experiments were conducted using 10-fold cross validation.

All classifiers were implemented in the `RecordLinkage` [6] open source record linkage system, which is written in Python. The `libsvm` library was used for the SVM classifier [3]. All experiments were run on a Pentium 3 GHz CPU with 2 GBytes of main memory, running Linux 2.6.20 and using Python 2.5.1.

Table 1. Data sets used in experiments. See Sect. 4 for more details.

Data set	Number of records	Task	Pairs completeness	Reduction ratio	Number of weight vectors
Census	449 + 392	Link	1.000	0.988	2,093
Restaurant	864	Dedup	1.000	0.713	106,875
DS-Gen-A	1,000	Dedup	0.957	0.995	2,475
DS-Gen-B	2,500	Dedup	0.940	0.997	9,878
DS-Gen-C	5,000	Dedup	0.953	0.997	35,491
DS-Gen-D	10,000	Dedup	0.948	0.997	132,532

As summarised in Table 1, experiments were conducted using two real data sets from the `secondstring` toolkit¹ and four synthetic data sets containing names and addresses created with the `secondstring` data set generator [4]. The synthetic data sets are based on real-world frequency tables, and contain 60% original and 40% duplicate records (with randomly generated duplicates [5]). Standard blocking [1] was applied to reduce the number of record pair comparisons, and the Winkler string comparator [12] was used for comparing name and address values.

The quality and complexity of the compared record pairs is shown in Table 2 using the measures $\text{Precision} = \frac{|\mathbf{W}_M|}{|\mathbf{W}|}$ (number of true matched record pairs generated by blocking divided by the total number of true matched pairs) and $\text{Recall} = \frac{|\mathbf{W}_M|}{|\mathbf{W}_M| + |\mathbf{W}_N|}$ (one minus the number of record pairs generated by blocking divided by the total number of pairs) [7,8]. The F-measure, the harmonic mean of precision and recall, is used to measure classifier performance, as accuracy is not suitable for evaluating record pair classification due to the imbalanced distribution of matches and non-matches [7]. The quality of the training example sets generated in step one, as shown in Table 2, is calculated as the percentage of correctly selected weight vectors in the training example sets, i.e. $(|\text{true matches in } \mathbf{W}_M|/|\mathbf{W}_M|)$ and $(|\text{true non-matches in } \mathbf{W}_N|/|\mathbf{W}_N|)$.

4.1 Training Example Quality

As Table 2 shows, the quality of \mathbf{W}_M and \mathbf{W}_N is very good in most cases. For the threshold based approach, setting $t_M, t_N = 0.5$ achieved the best results, while a lower threshold can produce empty training sets (denoted by ‘-’), if all weight vectors have at least one matching weight with a similarity value above the selected threshold. Nearest-based selection overcomes this problem, and generally results in very good quality training example sets [5].

4.2 Classification Performance

Figure 3 shows the F-measure results of two data sets (due to limited space not all results can be shown) over the parameter settings described in Sect. 4. The

¹ <http://secondstring.sourceforge.net>

Table 2. Quality of training examples generated in the first step, adapted from [5]. Each pair of result values shows the quality of $\mathbf{W}_M / \mathbf{W}_N$ as percentages of correctly selected training examples. ‘-’ denotes an empty training set.

Data sets	Thresholds			Nearest		
	0.3	0.5	0.7	1%	5%	10%
Census	100/-	96.2/100	73.4/100	100/100	100/100	100/100
Restaurant	98.5/-	4.5/100	0.19/100	100/100	76.7/100	58.6/100
DS-Gen-A	100/100	100/100	100/99.0	100/100	100/95.9	100/95.5
DS-Gen-B	100/100	100/100	99.8/99.4	100/99.0	100/98.4	100/98.2
DS-Gen-C	100/100	100/100	98.0/99.7	100/99.7	100/99.7	100/99.6
DS-Gen-D	100/99.7	100/100	95.5/99.9	100/99.8	100/99.8	100/99.7

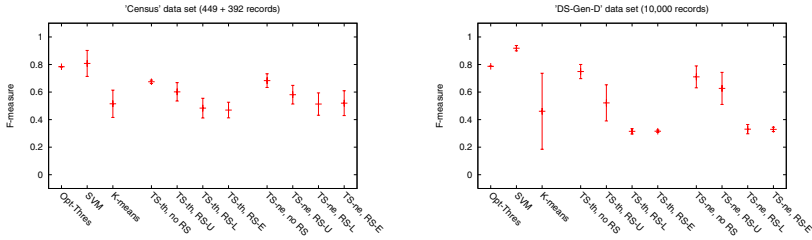


Fig. 3. F-measure results (averages and standard deviations). ‘TS’ stands for two-step, with ‘ne’ for nearest and ‘th’ for threshold based training example selection. ‘RS’ denotes the random selection: ‘U’ for uniform, ‘L’ for linear and ‘E’ for exponential.

four random selection methods described in Sect. 3.2 are shown for the two-step classifier (with both 1% and 10% randomly added weight vectors evaluated). As expected, both supervised classifiers (optimal threshold and SVM) performed best. The two-step classifier outperformed k -means only without random selection of additional weight vectors. Contrary to expectations, all random inclusion methods worsen the quality of the training sets and result in significantly reduced classification performance for the two data sets shown (for other data sets, slightly better classification results were achieved for linear or exponential random selection compared to no random selection). These results indicate that, unlike the random inclusion of ‘spy’ documents for semi-supervised text classification [9], inclusion of additional randomly selected weight vectors in the two-step approach is not improving record pair classification.

Given that normally only a portion of all weight vectors from \mathbf{W} are used for training in the two-step classifier, the training time will be reduced. For example, if only 10% of all weight vectors are selected for training, then the training step should be around ten times faster compared to using all weight vectors in \mathbf{W} , improving the scalability of record pair classification for large data sets.

5 Conclusions and Future Work

The discussed two-step approach allows unsupervised record pair classification with often better linkage quality than k -means clustering. Contrary to expectations, the inclusion of randomly selected additional weight vectors did not increase classification performance. Future work will include the evaluation of an approach that iteratively refines the training example sets by including the strongest classified matches and non-matches in each iteration, similar to the PEBL classifier developed for text and Web page classification [13].

Acknowledgements

This work is supported by an Australian Research Council (ARC) Linkage Grant LP0453463 and partially funded by the New South Wales Department of Health.

References

1. Baxter, R., Christen, P., Churches, T.: A comparison of fast blocking methods for record linkage. In: ACM KDD 2003 workshop on Data Cleaning, Record Linkage and Object Consolidation, Washington DC, pp. 25–27 (2003)
2. Bilenko, M., Mooney, R.J.: Adaptive duplicate detection using learnable string similarity measures. In: ACM KDD 2003, Washington DC, pp. 39–48 (2003)
3. Chang, C.-C., Lin, C.-J.: LIBSVM: A library for support vector machines. Manual, Department of Computer Science, National Taiwan University (2001)
4. Christen, P.: Probabilistic data generation for deduplication and data linkage. In: Gallagher, M., Hogan, J.P., Maire, F. (eds.) IDEAL 2005. LNCS, vol. 3578, pp. 109–116. Springer, Heidelberg (2005)
5. Christen, P.: A two-step classification approach to unsupervised record linkage. In: AusDM 2007, CRPIT, Gold Coast, Australia, vol. 70 (2007)
6. Christen, P.: Febr1 - a freely available record linkage system with a graphical user interface. In: HDKM 2008, CRPIT, Wollongong, Australia, vol. 80 (2008)
7. Christen, P., Goiser, K.: Quality and complexity measures for data linkage and deduplication. In: Quality Measures in Data Mining. Studies in Computational Intelligence, vol. 43, pp. 127–151. Springer, Heidelberg (2007)
8. Elfeky, M.G., Verykios, V.S., Elmagarmid, A.K.: TAILOR: A record linkage toolbox. In: ICDE 2002, San Jose, pp. 17–28 (2002)
9. Liu, B., Lee, W.S., Yu, P.S., Li, X.: Partially supervised classification of text documents. In: ICML 2002, Sydney, Australia, pp. 387–394 (2002)
10. Nahm, U.Y., Bilenko, M., Mooney, R.J.: Two approaches to handling noisy variation in text mining. In: TextML 2002, Sydney, pp. 18–27 (2002)
11. Tejada, S., Knoblock, C.A., Minton, S.: Learning domain-independent string transformation weights for high accuracy object identification. In: ACM KDD 2002, Edmonton, pp. 350–359 (2002)
12. Winkler, W.E.: Methods for evaluating and creating data quality. Elsevier Information Systems 29(7), 531–550 (2004)
13. Yu, H., Han, J., Chang, K.C.C.: PEBL: positive example based learning for Web page classification using SVM. In: ACM KDD 2002, Edmonton (2002)

Analyzing PETs on Imbalanced Datasets When Training and Testing Class Distributions Differ

David Cieslak and Nitesh Chawla

University of Notre Dame, Notre Dame IN 46556, USA
{dcieslak, nchawla}@cse.nd.edu

Abstract. Many machine learning applications like finance, medicine, and risk management suffer from class imbalance: cases of interest occur rarely. Further complicating these applications is that the training and testing samples might differ significantly in their respective class distributions. Sampling has been shown to be a strong solution to imbalance and additionally offers a rich parameter space from which to select classifiers. This paper is concerned with the interaction between Probability Estimation Trees (PETs) [1], sampling, and performance metrics as testing distributions fluctuate substantially. A set of comprehensive analyses is presented, which anticipate classifier performance through a set of widely varying testing distributions.

1 Introduction

Finance, medicine, and risk management form the basis for many machine learning applications. A compelling aspect of these applications is that they present several challenges to the machine learning community. The common thread among these challenges persists to be class imbalance and cost-sensitive application, which has been a focus of significant recent work [2, 3]. However, the common assumption behind most of the related works is that the testing data carries the same class distribution as the training data. This assumption becomes limiting for the classifiers learned on the imbalanced datasets, as the learning usually follows a prior sampling stage to mitigate the effect of observed imbalance. This is, effectively, guided by the premise of improving the prediction on the minority class as measured by some evaluation function. Thus, it becomes important to understand the interaction between sampling methods, classifier learning, and evaluation functions when the class distributions change.

To illustrate, a disease may occur naturally in 15% of a North American population. However, an epidemic condition may drastically increase the rate of infection to 45%, instigating differences in $P(\text{disease})$ between the training and testing datasets. Thus, the class distribution between negative and positive classes changes significantly. Scalar evaluations of a classifier learned on the original population will not offer a reasonable expectation for performance during the epidemic. A separate, but related problem occurs when the model trained from a segment of North American population is then applied to a European population where the distribution of measured features can potentially differ significantly, even if the disease base-rate remains at the original 15%. This issue becomes critical as the learned classifiers are optimized on the sampling distributions spelled out during training to increase performance on minority or positive

class, as measured by some evaluation function. If sampling is the strong governing factor for the performance on imbalanced datasets, can we guide the sampling to have more effective generalization?

Contributions: We present a comprehensive empirical study investigating the effects of changing distributions on a combination of sampling methods and classifier learning. In addition, we also study the robustness of certain evaluation measures. We consider two popular sampling methods for countering class imbalance: undersampling and SMOTE [2,4]. To determine the optimal levels of sampling (under and/or SMOTE), we use a bruteforce wrapper method with cross-validation that optimizes on different evaluation measures like Negative Cross Entropy (*NCE*), Brier Score (*Brier*), and Area Under the ROC Curve (*AUROC*) on the original training distribution. The former focuses on quality of probability estimates and the latter focuses on rank-ordering. The guiding question here is – *what is more effective – improved quality of estimates or improved rank-ordering if the eventual testing distribution changes?* We use the wrapper to empirically discover the potentially best sampling amounts for the given classifier and evaluation measure. This allows us to draw observations on the suitability of popular sampling methods, in conjunction with the evaluation measures, on evolving testing distributions. We restrict our study to PETs [1] given their popularity in the literature. This also allows for a more focused analysis. Essentially, we used unpruned C4.5 decision trees [5] and considered both leaf frequency based probability estimates and Laplace smoothed estimates. We also present an analysis of the interaction between measures used for parameter discovery and evaluation. *Is a single evaluation measure more universal than the others, especially in changing distributions?*

2 Sampling Methods

Resampling is a prevalent, highly parameterizable treatment of the class imbalance problem with a large search space. Typically resampling improves positive class accuracy and rank-order [6,7,8,2]. To our knowledge, there is no empirical literature detailing the effects of sampling on the quality of probability estimates; however, it is established that sampling improves rank-order. This study examines two sampling methods: random undersampling and SMOTE [9]. While seemingly primitive, randomly removing majority class examples has been shown to improve performance in class imbalance problems. Some training information is lost, but this is counterbalanced by the improvement in minority class accuracy and rank-order. SMOTE is an advanced oversampling method which generates synthetic examples at random intervals between known positive examples. [2] provides the most comprehensive survey and comparison of current sampling methods.

We search a large sampling space via wrapper [10] using a heuristic to limit the search space. This strategy first removes “excess” negative examples by undersampling from 100% to 10% in 10% steps and then synthetically adds from 0% to 1000% more positive examples in 50% increments using SMOTE. Each phase ceases when the wrapper’s objective function no longer improves after three successive samplings. We use *Brier*, *NCE*, and *AUROC* [11,12] as objective functions to guide the wrapper and final evaluation metrics. Figure 1 shows the Wrapper and Evaluation framework.

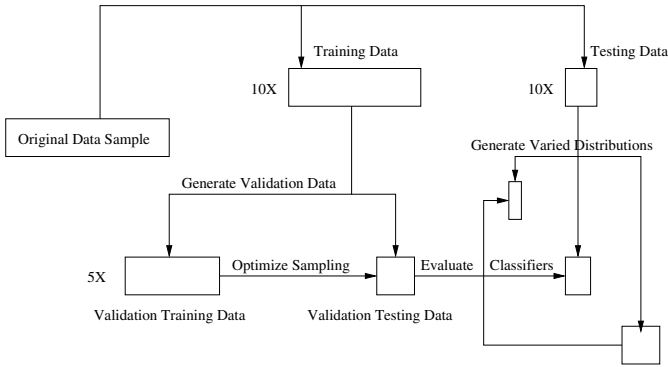


Fig. 1. Wrapper and Evaluation Framework

Table 1. Dataset Distributions. Ordered in an increasing order of class imbalance.

Dataset	Examples	Features	Class Balance
Adult [13]	48,840	14	76:24
E-State [9]	5,322	12	88:12
Pendigits [13]	10,992	16	90:10
Satimage [13]	6,435	36	90:10
Forest Cover [13]	38,500	10	93:7
Oil [14]	937	49	96:4
Compustat [10]	10,358	20	96:4
Mammography [9]	11,183	6	98:2

3 Experiments and Results

We consider performance on different samplings of the testing set to explore the range of potential distributions by exploring samplings for which $P(+)=\{0.02, 0.05, 0.1, 0.2, 0.3, \dots, 0.9, 0.95, 0.98\}$. For example, suppose a given dataset has 2000 examples from class 0 and 1000 examples of class 1 in the testing set. To evaluate on $P(+)=0.5$, 1000 class 0 examples are randomly removed from the evaluation set. We experimented on eight different datasets, summarized in Table 1.

We explore visualizations of the trends in NCE and $AUROC$ as $P(+)$ is varied. Each plot contains several different classifiers: the baseline PET [1]; sampling guided by *Brier* (called Frequency *Brier* Wrapper for frequency based estimates and Laplace *Brier* Wrapper for Laplace based estimates); sampling guided by NCE ; and finally sampling guided by $AUROC$ (the latter two using similar naming convention as *Brier*). In Figures 2 to 9 NCE and $AUROC$ are depicted as a function of increasing class distribution, ranging from fully negative on the left to fully positive on the right. A vertical line indicates the location of the original class distribution. *Brier* trends are omitted as they mirror those of NCE .

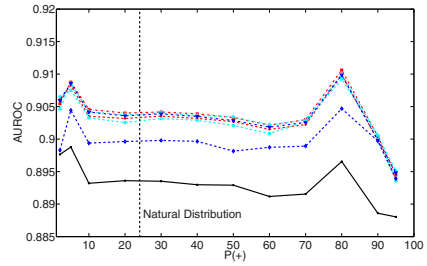
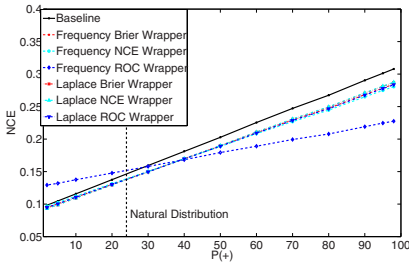


Fig. 2. Adult

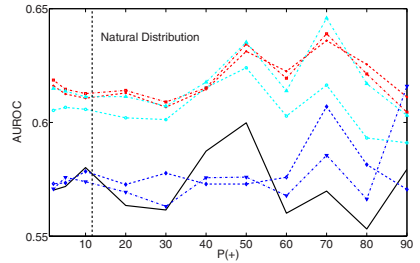
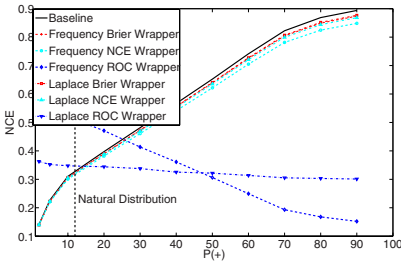


Fig. 3. E-State

Figures 2 through 9 show the experimental NCE and $AUROC$ trends as the class distribution varies. Despite the variety of datasets evaluated, some compelling general trends emerge. Throughout, we note that wrappers guided by losses generally improve NCE at and below the natural distribution of $P(+)$ as compared to $AUROC$ wrappers. This implies that loss does well in optimizing NCE when the testing distribution resembles the training conditions. It is notable that in some cases, such as Figures 2, 3, 5, 6, 7, 8, & 9, that the baseline classifier actually produces better NCE scores than at least the frequency $AUROC$ wrapper, if not both $AUROC$ wrappers. The frequency $AUROC$ wrapper selected extreme levels of sampling. The reduction in NCE at low $P(+)$ indicates that using loss measures within the wrapper lowers the loss estimates for the negative class examples. That is, while the loss from the positive class may actually increase, the lower overall losses are driven by better calibrated estimates on the predominantly occurring majority class. On the other hand, classifiers learned from the $AUROC$ guided wrappers do not result in as well-calibrated estimates. $AUROC$ favors the positive class rank-order, while $Brier$ and NCE tend to treat both classes equally, which in turn selects extreme sampling levels. Thus, if NCE optimization is desired and the positive class is anticipated to occur as rarely or more rarely than in the training data, sampling should be selected according to either $Brier$ or NCE .

However, the environment producing the data may be quite dynamic, creating a shift in the class ratio and causing the minority class to become much more prevalent. In a complete paradigm shift, the former minority class might become larger than the former majority class, such as in an epidemic. Invariably, there is a cross-over point in

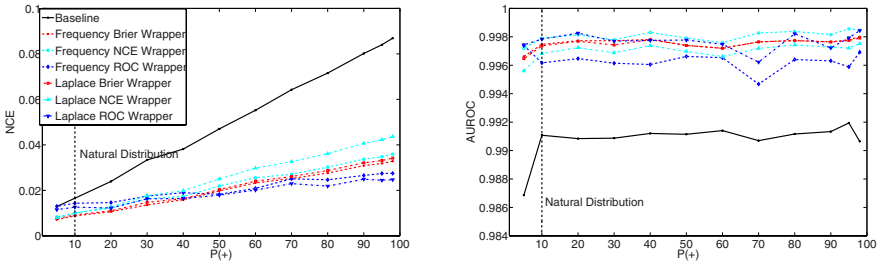


Fig. 4. Pendigits

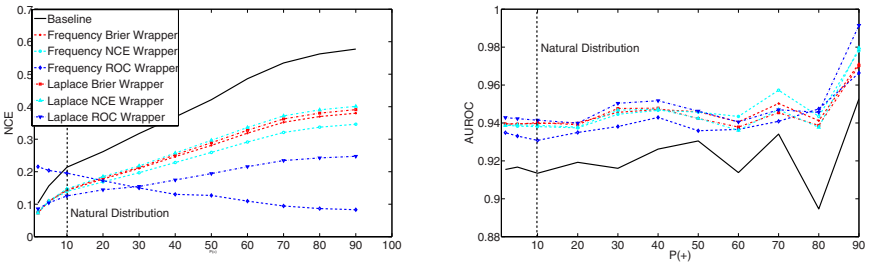


Fig. 5. Satimage

each dataset after which one of the $AUROC$ wrappers optimizes NCE values. This is logical, as $AUROC$ measures the quality of rank-order in terms of the positive class — extra emphasis is placed on correctly classifying positive examples and is reflected by the higher selected sampling levels. As the positive examples eventually form the majority of the evaluation set, classifiers producing on average higher quality positive class probability estimates will produce the best NCE . Therefore, if a practitioner anticipates an epidemic-like influx of positive examples sampling methods guided by $AUROC$ are favored.

Improvement to $AUROC$ under varied testing distributions is not as uniform. We observe that at least one loss function wrapper generally produces better $AUROC$ values in Figures 2, 3, & 4, but that an $AUROC$ wrapper is optimal in Figures 6, 7, & 9. It is difficult to declare a champion in Figures 5 & 8. It is of note that datasets with naturally larger positive classes tend to benefit (in terms of $AUROC$) from a loss wrapper, while those with naturally smaller positive classes benefit more from the $AUROC$ wrapper. As seen before, $AUROC$ guides a wrapper to higher sampling levels than *Brier* or *NCE*. In the cases of relatively few positive examples (such as Forest Cover, Oil, and Mammography), a heavy emphasis during training on these few examples produces better $AUROC$ values. For the datasets with a larger set of positive examples (as in Adult, E-State, and Pendigits) from which to naturally draw, this over-emphasis does not produce as favorable a result. Therefore, in cases where there are very few positive examples, a practitioner should optimize sampling according to $AUROC$. Otherwise, *Brier* or *NCE* optimization is sufficient.

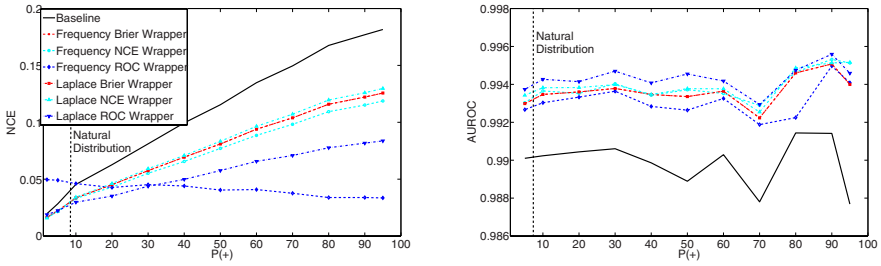


Fig. 6. Forest Cover

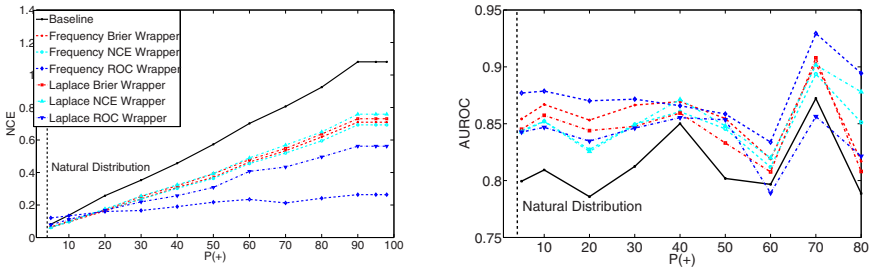


Fig. 7. Oil

The difference of characteristics between the trends in NCE and $AUROC$ is noteworthy. The NCE trends appear stable and linear. By calculating the loss on each class at the base distribution, it appears that one is able to project the NCE at any class distribution using a weighted average. $AUROC$ trends are much more violent, likely owing to the highly perturbable nature of the measure. Adding or removing a few examples can heavily impact the produced ranking. As a measure, $AUROC$ is characteristically less predictable than a loss function.

We also note that sampling mitigates the need of application of Laplace smoothing at the leaves. We can see that the baseline classifier benefits from smoothing, as also noted by other works. However, by treating the dataset for class imbalance first, we are able to counter the bias and variance in estimates arising from small leaf-sizes. The wrapper essentially searches for the ideal training distribution by undersampling and/or injecting synthetic minority class instances that lead to a reduction in loss or improvement in ranking.

Throughout Figures 2 to 9, we also note that $Brier$ and NCE loss wrappers tend to perform similarly across measures and datasets. This is not surprising as the shape of $Brier$ and NCE values are similar. We observe that the optimal sampling levels found by $Brier$ and NCE are similar, certainly more similar than to those samplings of $AUROC$. In general, NCE maintains a slight performance edge. If in the interests of time a practitioner may only experiment using one loss measure, then this study recommends using NCE , although the results found here may not apply to all domains and performance metrics.

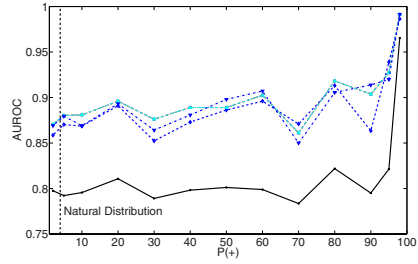
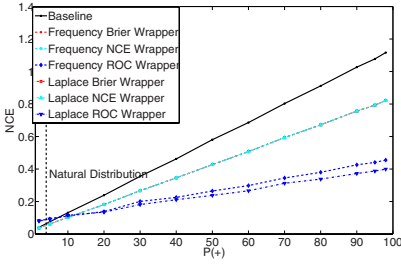


Fig. 8. Compustat

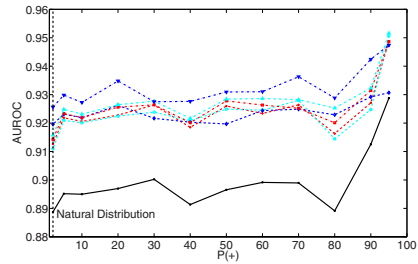
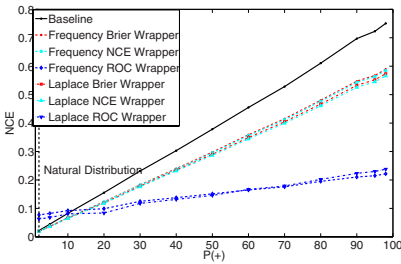


Fig. 9. Mammography

4 Conclusions

The main focus of our paper was to empirically explore and evaluate the interaction between techniques for countering class imbalance, PETs, and corresponding evaluation measures under circumstances where training and testing samples differ. In light of the questions posited in the Introduction, we make the following key observations.

- We demonstrated that it is possible to identify potentially optimal quantities of sampling by optimizing on quality of estimates or rank-order as calculated by AUROC. Almost all the wrappers demonstrated significant improvements in AUROC and reductions in losses over the baseline classifier, irrespective of the dataset. As an evaluation measure, NCE is much more stable and predictable as compared to $AUROC$. We observe NCE to change almost linearly as a function of $P(+)$, while $AUROC$ tends to change as $P(+)$ changes.
- There is a strong inter-play between undersampling and SMOTE. The wrapper determines an interaction between both the approaches by searching undersampling parameters before oversampling via SMOTE.
- It is much more difficult to anticipate the effects of a class distribution shift on $AUROC$ than it is on probability loss functions. When a dataset is highly imbalanced, we recommend guiding sampling through $AUROC$ as this places the necessary emphasis on the minority class. When class imbalance is much more moderate, NCE tends to produce an improved $AUROC$.

- While Laplace smoothing has a profound effect in improving both the quality of estimates and ranking for the baseline classifier, the advantage diminishes with sampling methods. The combination of SMOTE and undersampling improves the calibration at the leaves and thus we observed that wrapper based sampling methods are able to improve performance — lower losses and higher ranking — irrespective of smoothing at the leaves.

References

1. Provost, F., Domingos, P.: Tree Induction for Probability-Based Ranking. *Machine Learning* 52(3), 199–215 (2003)
2. Batista, G., Prati, R., Monard, M.: A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data. *SIGKDD Explorations* 6(1), 20–29 (2004)
3. Chawla, N., Japkowicz, N., Kolcz, A.: Editorial: Special Issue on Learning from Imbalanced Data Sets. *SIGKDD Explorations* 6(1), 1–6 (2004)
4. Estabrooks, A., Jo, T., Japkowicz, N.: A Multiple Resampling Method for Learning from Imbalanced Data Sets. *Computational Intelligence* 20(1), 18–36 (2004)
5. Quinlan, J.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo (1992)
6. Japkowicz, N.: The Class Imbalance Problem: Significance and Strategies. In: *ICAI 2000*, pp. 111–117 (2000)
7. Ling, C.X., Li, C.: Data Mining for Direct Marketing: Problems and Solutions. In: *KDD*, pp. 73–79 (1998)
8. Solberg, A., Solberg, R.: A Large-Scale Evaluation of Features for Automatic Detection of Oil Spills. In: *ERS SAR Images IEEE Symp. Geosc. Rem.*, vol. 3, pp. 1484–1486 (1996)
9. Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P.: SMOTE: Synthetic Minority Over-sampling Technique. *JAIR* 16, 321–357 (2002)
10. Chawla, N.V., Cieslak, D.A., Hall, L.O., Joshi, A.: Automatically countering imbalance and its empirical relationship to cost. *Utility-Based Data Mining: A Special issue of the International Journal Data Mining and Knowledge Discovery* (2008)
11. Buja, A., Stuetzle, W., Sheu, Y.: Loss Functions for Binary Class Probability Estimation and Classification: Structure and Applications (under submission, 2006)
12. Caruana, R., Niculescu-Mizil, A.: An Empirical Comparison of Supervised Learning Algorithms. In: *ICML 2006*, pp. 161–168 (2006)
13. Asuncion, A., Newman, D.: *UCI machine learning repository* (2007)
14. Kubat, M., Holte, R., Matwin, S.: Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning* 30, 195–215 (1998)

Improving the Robustness to Outliers of Mixtures of Probabilistic PCAs

Nicolas Delannay¹, Cédric Archambeau², and Michel Verleysen¹

¹ Université catholique de Louvain, Machine Learning Group - DICE
3 place du Levant, B-1348 Louvain-la-Neuve, Belgium
`michel.verleysen@uclouvain.be`

² Centre for Comput. Statistics and Machine Learning, University College London
Gower Street, London WC1E 6BT, U.K.
`c.archambeau@cs.ucl.ac.uk`

Abstract. Principal Component Analysis, when formulated as a probabilistic model, can be made robust to outliers by using a Student-t assumption on the noise distribution instead of a Gaussian one. On the other hand, mixtures of PCA is a model aimed to discover nonlinear dependencies in data by finding clusters and identifying local linear sub-manifolds. This paper shows how mixtures of PCA can be made robust to outliers too. Using a hierarchical probabilistic model, parameters are set by likelihood maximization. The method is shown to be effectively robust to outliers, even in the context of high-dimensional data.

1 Introduction

Principal Component Analysis (PCA) is a well-known data analysis and visualization tool. It provides a simple, algebraic way to choose axes in the data space that most fit the data, i.e. that maximize the variance after projection on the subspace spanned by these axes, or alternatively that minimize the projection error. A lower-dimensional representation of data is obtained by selecting a restricted number of the principal axes. However, maximal variance and minimal projection error are quadratic measures: a few outliers may dramatically influence the direction of principal axes, especially in high-dimensional spaces.

Probabilistic PCA [10,13] is a way to formalize the PCA problem as a latent variable model into a probabilistic framework. One of the nice features of the probabilistic framework is that non-traditional assumptions can easily be added to the model, the only price to pay being that the optimization of the model may reveal more difficult. For example, the traditional Gaussian noise hypothesis leads to the above detailed quadratic measures of errors and variances; replacing this hypothesis by, for instance, a Student-t noise distribution leads to a robust version of PCA [2]. In contrast to other robust approaches to PCA which usually require to optimize several additional parameters, the probabilistic formalism only requires to choose the dimension of the projection space, the other parameters being set automatically by maximum likelihood (ML). Another advantage is that the probabilistic model provides likelihood measures, which can be used to compute posterior probabilities and eventually to construct a Bayes classifier.

Mixtures of (local) PCA may be used to uncover nonlinear manifolds in data, and are also nicely formalized into a probabilistic framework [12]. The principle is to attribute each observed data to a specific (unknown) local model (or component), through an indicator variable, and then to mix the local models. An expectation-maximization algorithm can be used to set the parameters of the model, including these indicator variables. An advantage of mixtures of PCA, compared to other mixtures models (a.o. Gaussian mixtures), is that the full-rank, possibly ill-conditioned covariance matrices are approximated by low-rank covariance matrices, without having to neglect the correlations between the (local) principal directions to avoid numerical instabilities. The other way to avoid ill-conditioned covariance matrices is to constrain them to be diagonal, leading to suboptimal axis-aligned components [1]. Besides nonlinear manifold uncovering, mixtures models can be used in a straightforward way for clustering, and probability density estimation. In both cases the same limitations related to ill-conditioned covariance matrices apply though.

Mixtures of probabilistic PCA [3] can be made robust to atypical observations by using a Student-t noise distribution hypothesis. This paper shows the complete probabilistic learning procedure for this model. It is shown that all parameters (with the exception of the number of components and their dimensionality) may be easily optimized by an Expectation-Maximization procedure, without additional complexity with respect to the non-robust version.

The following of this paper is organized as follows. The next section first reminds the Probabilistic PCA model and its robust extension, and then introduces the Mixtures of Robust Probabilistic PCA model. Section 3 details how the parameters of the model may be optimized, and Section 4 illustrates the robustness of the model to atypical observations.

2 Robust Probabilistic PCA and Mixtures

PCA can be formulated as the search for an optimal linear projection minimizing a reconstruction error. The principal components are derived from the observations by projecting them on the principal directions. In the probabilistic formulation, the view is inverted in the sense that the observations $\{y_n\}_{n=1}^N$ where $y_n \in \mathbb{R}^D$, are assumed to be generated from a low dimension latent representation $\{x_n\}_{n=1}^N$, where $x_n \in \mathbb{R}^J$, $J < D$.

The principle of probabilistic modeling is to express the uncertainty about (some of) the parameters of the model by prior distributions. Probabilistic PCA (PPCA) was proposed in [10,13]; Gaussian priors are used in PPCA. Maximising the likelihood of the observations in PPCA leads to principal axes that are equivalent to the principal axes found by the standard PCA, up to a rotation and a scaling [13]; the same subspace is thus spanned.

PCA and PPCA are sensitive to atypical observations and observations not well confined in a low-dimensional subspace, because of their quadratic criterion and Gaussian noise model respectively. The robust probabilistic PCA [2] extends PPCA to make it applicable on datasets containing atypical samples. Instead

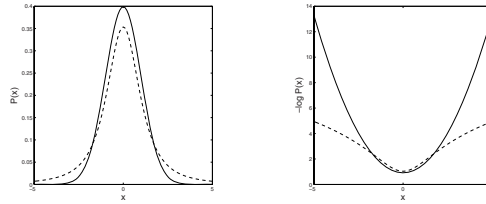


Fig. 1. Left: Probability density functions of a Gaussian (–) and a Student- t with $\nu = 2$ (---). Right: Negative log-likelihood of these same distributions.

of the Gaussian noise assumption, the randomness in observations is modeled by a Student- t distribution with an additional parameter ν (called the *degrees of freedom*), which regulates the thickness of the distribution’s tail. Figure 1(left) shows unit-variance Gaussian and Student- t distributions ($\nu = 2$). Figure 1(right) shows the corresponding negative-log-likelihood which appears in the training criterion of probabilistic models. We see that when ν is small, the Student- t attributes a much smaller cost than the Gaussian to points lying far from the mean. The sensitivity to atypical observations is therefore reduced.

PPCA makes the assumption that atypical samples might come either from the generation of latent vectors \mathbf{x} or from the noise contribution. This is expressed by Student- t distributions on the prior of the latent vectors and on the conditional distribution of observations: $P(\mathbf{x}) = \mathcal{S}(\mathbf{x}|\mathbf{0}, \mathbf{I}_J, \nu)$, $P(\mathbf{y}|\mathbf{x}) = \mathcal{S}(\mathbf{y}|\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \tau^{-1}\mathbf{I}_D, \nu)$. Note that in the traditional PPCA model, the Student- t distributions are replaced by Gaussian ones. To simplify the parameterization, both distributions are attributed the same degree of freedom ν . This choice will be commented below. The Student- t distribution can be reformulated as an infinite mixture of Gaussian distributions $\mathcal{S}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu) = \int_0^\infty \mathcal{N}(\mathbf{y}|\boldsymbol{\mu}, \frac{1}{u}\boldsymbol{\Sigma}) \mathcal{G}(u|\frac{\nu}{2}, \frac{\nu}{2}) du$, $\nu > 0$, where $\mathcal{G}(u|\cdot, \cdot)$ is a Gamma distribution over the precision factor u . Making use of this factorization, the generative model can be represented with an additional level in the hierarchy where the latent precision u appears:

$$P(u) = \mathcal{G}(u|\frac{\nu}{2}, \frac{\nu}{2}) \tag{1}$$

$$P(\mathbf{x}|u) = \mathcal{N}(\mathbf{x}|\mathbf{0}, \frac{1}{u}\mathbf{I}_J) \tag{2}$$

$$P(\mathbf{y}|\mathbf{x}, u) = \mathcal{N}(\mathbf{y}|\mathbf{W}\mathbf{x} + \boldsymbol{\mu}, \frac{1}{u\tau}\mathbf{I}_D) . \tag{3}$$

From this generative formulation, we see that the uncertainty about the observation (i.e. expressed by the variance in (3)) can be amplified by a small latent precision variable u , shared by the \mathbf{x} and \mathbf{y} conditional distributions. According to intuition, this constraint implies that outliers \mathbf{y} in the observation space are also considered as outliers \mathbf{x} in the latent space so their contributions to the identification of the latent space are down-weighted.

For robust PPCA, the marginal distribution of the observations is tractable: $P(\mathbf{y}) = \int_0^\infty \int_{\mathcal{X}} P(\mathbf{y}|\mathbf{x}, u) P(\mathbf{x}|u) P(u) d\mathbf{x} = \mathcal{S}(\mathbf{y}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \nu)$ where $\boldsymbol{\Sigma} \equiv \mathbf{W}\mathbf{W}^\top +$

$\tau^{-1}\mathbf{I}_D$. The training procedure consists in maximizing this (marginal) likelihood with respect to $\theta \equiv (\mathbf{W}, \boldsymbol{\mu}, \tau, \nu)$.

In contrast with previous robust approaches to the PCA (see for example [15] and [7], and the references therein), this probabilistic formalism only requires to select the dimension of the projection space (see Section 3), the other parameters being estimated by the maximum likelihood criterion.

Even in its robust and probabilistic versions, PCA is not adequate for representing clusters or nonlinear dependencies in the data. The mixture of PPCA [12] may solve this problem, but is again too sensitive to atypical samples limiting its use on many real world datasets. It is thus natural to look for a robust formulation of the mixture of PPCA.

The probability distribution of a sample generated from a mixture of K robust PPCA is defined as $P(\mathbf{y}) = \sum_k \pi_k P_k(\mathbf{y})$ where $\{\pi_k\}_{k=1}^K$ is the set of positive mixture proportions, with $\sum_k \pi_k = 1$; the $P_k(\mathbf{y})$ are defined as single robust PPCA components $P_k(\mathbf{y}) = \mathcal{S}(\mathbf{y}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)$ in which $\boldsymbol{\Sigma}_k \equiv \mathbf{W}_k \mathbf{W}_k^T + \tau_k^{-1} \mathbf{I}_D$. The set of parameters of this model is $\theta \equiv \{(\mathbf{W}_k, \boldsymbol{\mu}_k, \tau_k, \nu_k, \pi_k)\}_{k=1}^K$.

Using a latent indicator variable $\mathbf{z} = [z_1, \dots, z_K]$ (with $z_k = 1$ if the k th component generated the observation \mathbf{y} , otherwise $z_k = 0$) simplifies the derivation of an EM algorithm. The factorized mixture of robust PPCA is then

$$P(\mathbf{z}) = \prod_k \pi_k^{z_k} \tag{4}$$

$$P(\mathbf{u}|\mathbf{z}) = \prod_k \mathcal{G}(u_k | \frac{\nu_k}{2}, \frac{\nu_k}{2})^{z_k} \tag{5}$$

$$P(\boldsymbol{\chi}|\mathbf{u}, \mathbf{z}) = \prod_k \mathcal{N}(\mathbf{x}_k | \mathbf{0}, \frac{1}{u_k} \mathbf{I}_J)^{z_k} \tag{6}$$

$$P(\mathbf{y}|\boldsymbol{\chi}, \mathbf{u}, \mathbf{z}) = \prod_k \mathcal{N}(\mathbf{y} | \mathbf{W}_k \mathbf{x}_k + \boldsymbol{\mu}_k, \frac{1}{u_k \tau_k} \mathbf{I}_D)^{z_k} \tag{7}$$

where $\mathbf{u} = [u_1, \dots, u_K]$ and $\boldsymbol{\chi} = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$; the different components could also have different latent dimensionalities $\{J_k\}_{k=1}^K$.

Increasing the robustness by replacing Gaussian densities with Student- t ones was also proposed for finite mixture models [8,11]. The main advantage of mixtures of PPCA resides in the fact that the full-rank, possibly ill-conditioned covariance matrices are approximated by constrained covariance matrices $\boldsymbol{\Sigma}_k$, strongly reducing the number of free parameters per component. By contrast, constraining the covariance to be diagonal leads to axis-aligned components which does not take the dominant correlations into account [1].

3 Learning Procedure

The factorization of the model (4)-(7) allows us to derive an exact Expectation-Maximization (EM) algorithm. Note that this algorithm encompasses the optimization of the (mixture of) probabilistic PCA: one only needs to add the constraint $\nu_k = \infty$ (for all k) such that the Student- t s are in fact Gaussian distributions.

We seek an optimum of the marginal distribution of the observations to estimate the parameters $\theta \equiv \{(\mathbf{W}_k, \boldsymbol{\mu}_k, \tau_k, \nu_k, \pi_k)\}_{k=1}^K$. The simplest way to proceed is by deriving an EM algorithm [6] on the factorised distribution (4)-(7). The

starting point of the algorithm is to bound the marginal likelihood (making use of the Jensen’s inequality):

$$\log P(\{\mathbf{y}_n\}) \geq \mathbb{E}_Q \{ \log P(\{\mathbf{y}_n\}, \{\mathcal{X}_n\}, \{\mathbf{u}_n\}, \{\mathbf{z}_n\}) \} - \mathbb{E}_Q \{ \log Q(\{\mathcal{X}_n\}, \{\mathbf{u}_n\}, \{\mathbf{z}_n\}) \} . \tag{8}$$

Equation (8) is valid for any distribution Q . The bound is tight when the distribution over the latent variable $Q(\{\mathcal{X}_n\}, \{\mathbf{u}_n\}, \{\mathbf{z}_n\})$ coincides with the posterior distribution. Fortunately, the posterior distribution of the mixture of robust PPCA model is still tractable. Indeed, applying the Bayes formula, one can show that the posterior is

$$P(\{\mathcal{X}_n\}, \{\mathbf{u}_n\}, \{\mathbf{z}_n\} | \{\mathbf{y}_n\}) = \prod_n \prod_k P(\mathbf{x}_{nk} | u_{nk}, z_{nk} = 1, \mathbf{y}_n) \cdot P(u_{nk} | z_{nk} = 1, \mathbf{y}_n) P(z_{nk} = 1 | \mathbf{y}_n) \tag{9}$$

where the factor distributions are

$$P(\mathbf{x}_{nk} | u_{nk}, z_{nk} = 1, \mathbf{y}_n) = \mathcal{N}(\mathbf{x}_{nk} | \tau_k \mathbf{C}_k \mathbf{W}_k^\top (\mathbf{y}_n - \boldsymbol{\mu}_k), \frac{1}{u_{nk}} \mathbf{C}_k) \tag{10}$$

$$P(u_{nk} | z_{nk} = 1, \mathbf{y}_n) = \mathcal{G}(u_{nk} | \alpha_k, \beta_{nk}) \tag{11}$$

$$P(z_{nk} = 1 | \mathbf{y}_n) = \frac{\pi_k \mathcal{S}_\cdot(\mathbf{y}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)}{\sum_k \pi_k \mathcal{S}_\cdot(\mathbf{y}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)} \tag{12}$$

and where we have defined $\mathbf{C}_k^{-1} = \tau_k \mathbf{W}_k^\top \mathbf{W}_k + \mathbf{I}_J$, $\alpha_k = (D + \nu_k)/2$, and $\beta_{nk} = ((\mathbf{y}_n - \boldsymbol{\mu}_k)^\top \boldsymbol{\Sigma}_k^{-1} (\mathbf{y}_n - \boldsymbol{\mu}_k) + \nu_k)/2$. Notice that there is only a single observation \mathbf{y}_n appearing in each of these posterior factor distributions.

The EM algorithm then consists in two successive and repeated steps. The *E*-step consists in fixing Q to the distribution given by (9) and developing (8) accordingly. Note that only the first term of (8) (called the log-complete likelihood) has to be computed, as the second one does not depend on the values of the parameters. This leads to a somewhat complex expression, not detailed here for simplicity. Its evaluation necessitates to compute the following expectations:

$$\bar{\rho}_{nk} \equiv \mathbb{E}_Q \{ z_{nk} \} = \frac{\pi_k \mathcal{S}_\cdot(\mathbf{y}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)}{\sum_k \pi_k \mathcal{S}_\cdot(\mathbf{y}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k, \nu_k)} , \tag{13}$$

$$\bar{u}_{nk} \equiv \mathbb{E}_Q \{ u_{nk} \} = \frac{\alpha_k}{\beta_{nk}} , \tag{14}$$

$$\log \bar{u}_{nk} \equiv \mathbb{E}_Q \{ \log u_{nk} \} = \psi(\alpha_k) - \log(\beta_{nk}) , \tag{15}$$

$$\bar{\mathbf{x}}_{nk} \equiv \mathbb{E}_Q \{ \mathbf{x}_{nk} \} = \tau_k \mathbf{C}_k \mathbf{W}_k^\top (\mathbf{y}_n - \boldsymbol{\mu}_k) , \tag{16}$$

$$\bar{\mathbf{S}}_{nk} \equiv \mathbb{E}_Q \{ z_{nk} u_{nk} \mathbf{x}_{nk} \mathbf{x}_{nk}^\top \} = \bar{\rho}_{nk} \mathbf{C}_k + \bar{\omega}_{nk} \bar{\mathbf{x}}_{nk} \bar{\mathbf{x}}_{nk}^\top , \tag{17}$$

where $\bar{\omega}_{nk} \equiv \bar{\rho}_{nk} \bar{u}_{nk}$ and $\psi(\cdot) \equiv \Gamma'(\cdot)/\Gamma(\cdot)$ is called the digamma function.

The log-complete likelihood of course depends on the model parameters; the *M*-step then consists in maximizing it with respect to the parameters, leading

to a set of update rules for all k ($\text{tr}\{\cdot\}$ is the trace operator):

$$\pi_k \leftarrow \frac{1}{N} \sum_n \bar{\rho}_{nk} \quad (18)$$

$$\boldsymbol{\mu}_k \leftarrow \frac{\sum_n \bar{\omega}_{nk} (\mathbf{y}_n - \mathbf{W}_k \bar{\mathbf{x}}_{nk})}{\sum_n \bar{\omega}_{nk}} \quad (19)$$

$$\mathbf{W}_k \leftarrow \left(\sum_n \bar{\omega}_{nk} (\mathbf{y}_n - \boldsymbol{\mu}_k) \bar{\mathbf{x}}_{nk}^\top \right) \left(\sum_n \bar{\mathbf{S}}_{nk} \right)^{-1} \quad (20)$$

$$\tau_k^{-1} \leftarrow \frac{1}{DN\pi_k} \sum_n \left(\bar{\omega}_{nk} \|\mathbf{y}_n - \boldsymbol{\mu}_k\|^2 - \text{tr}\{\mathbf{W}_k \bar{\mathbf{S}}_{nk} \mathbf{W}_k^\top\} \right) \quad (21)$$

In these updates rules, the contribution of each data point is weighted according to $\bar{\omega}_{nk}$, which accounts for both the effect of the responsibilities $\bar{\rho}_{nk}$ and the expected latent precision variables \bar{u}_{nk} . The latter ensures robustness as its value is small for \mathbf{y}_n lying far from $\boldsymbol{\mu}_k$, such that the contribution in the M-step is small. For the non robust formulation ($\nu_k \rightarrow \infty$) we have $\bar{u}_{nk} = 1$ for all n and all k . Note also that these updates are coupled: one could cycle through these updates between each E-step until the M-step has converged.

There is no closed form update for $\{\nu_k\}_{k=1}^K$. Nevertheless, a solution can be computed by line search at each EM iteration [2]. Alternatively, a heuristic was proposed by Shoham [11] in the context of mixture modeling.

As the marginal likelihood of mixture models has local optima, it is recommended to repeat the optimization with different initializations. A good strategy to initialize the components is to set the centers $\boldsymbol{\mu}_k$ with a quantization algorithm and initialize the subspace orientation \mathbf{W}_k from the first Principal directions in the Voronoi region of $\boldsymbol{\mu}_k$.

Two parameters still need to be set: the number of components and the dimensionalities of the latent representations. They can be set in a traditional way by cross-validation, or added in a Bayesian way to the probabilistic formulation; in the latter case however MCMC sampling techniques [9] or (mean field) variational approximation [14,4] must be used instead of the exact EM algorithm. Finally Automatic Relevance Determination was used in [5] to select the dimensionality of latent subspaces.

4 Experiments

In this section, the (robust) probabilistic models are applied first on two artificial examples, and then on the USPS high-dimensional real dataset, using the software available from <http://www.ucl.ac.be/mlg/>.

Figures 2(a)-(b) show an example where samples have been generated along a one-dimensional manifold, with higher density in the right end and higher noise at the other end. The PPCA estimates a global principal direction; the mean of the component lies in an empty region and is thus not representative of typical samples. On the other hand, the robust PPCA discards samples in order to concentrate on the higher density region of the manifold. Using three components in the model (Figures 2(c)-(d)), both the mixture of PPCA and robust PPCA estimate quite well the local principal directions. However one of the components of the mixture of PPCA (Figure 2(c)) tries to account for the noisy samples, forcing its mean to move away from the manifold.

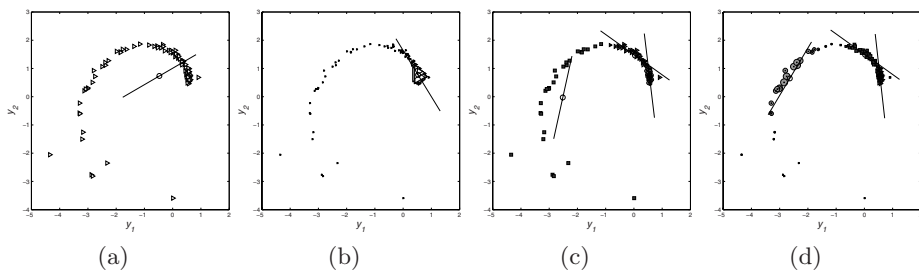


Fig. 2. Samples generated along a 1-dimensional manifold with additional atypical points. (a) Probabilistic PCA, (b) robust probabilistic PCA, (c) 3 components mixture of PPCA, (d) 3 components mixture of robust PPCA. The sizes of the markers represent their contribution to the estimation of the component parameters.

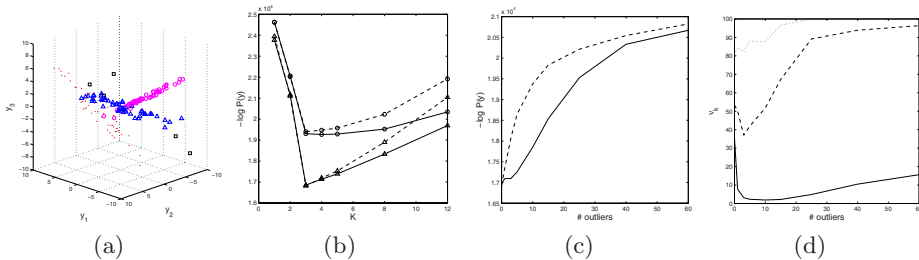


Fig. 3. (a) Synthetic example with 3 Gaussian clusters. The squares represent outliers. (b) Negative log likelihood of a validation set with respect to the number of components K and the dimensionality of the latent space (\circ : $J = 1$, \triangle : $J = 2$). Dashed line: standard. Plain line: robust. (c) Negative log likelihood with respect to the number of outliers. (d) Degree of freedom parameters for the three components in the robust mixture model with respect to the number of outliers.

The next example consists in data arranged in three 3-dim. Gaussian clusters (see Figure 3(a)), with diagonal covariance matrices equal to $\text{diag}\{[5, 1, 0.2]\}$ before rotation around the second coordinate axis. Each component lies on an intrinsic two dimensional space as the variance in the third direction is significantly smaller. The two outer clusters make an angle of ± 30 degrees with the middle one and are respectively shifted by ± 5 units along the axis of rotation. For the first experiment, 30 data are generated for each cluster. The generalisation performances, measured as the log likelihood on a validation set averaged on 50 experiments, are plotted in Figure 3(b) for $K \in \{1, \dots, 12\}$ components and $J \in \{1, 2\}$ latent space dimensions. As expected, the true model with $K = 3$ and $J = 2$ performs the best. Interestingly, we see that the standard and robust mixture models have comparable performances when the model underfits the data (i.e. $K < 2$) while the robust mixture has the edge when K increases. Overfitting is thus reduced with the robust formulation.

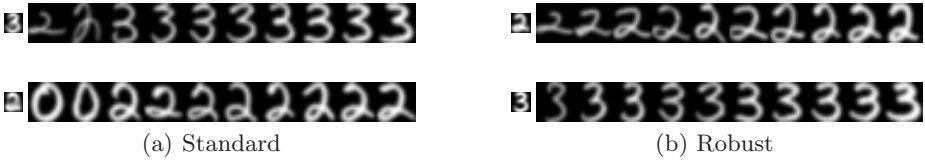


Fig. 4. Mixture of 2 component PPCAs with 1-dimensional latent space to cluster USPS digit 2 and 3, and outliers digit 0. (a) standard; (b) robust.

For the second experiment, K is set to 3 and J to 2 (their optimal values); we look at the sensitivity of the model to the number of outliers. The outliers are generated uniformly in the $[-10, 10]^3$ box. Again, 30 points are generated from each component; 1 to 60 outliers are added. The performances measured on a validation set without outliers, and averaged over 50 repetitions as above, are shown on Figure 3(c). Again, we see the increased robustness of the proposed model, in particular when there are few outliers. When the number of outliers increases to a significant proportion of the learning data the down-weighting of the outliers in the robust model is reduced, and the gap between the performances decreases. Figure 3(d) shows the average value of the degree of freedom parameters (ν_k for $k = 1 \dots 3$). We note that the down-weighting of the outliers obtained with small value of ν_k , comes mainly from a single component.

The last example illustrates the robustness of the proposed method on high-dimensional data. The USPS handwritten digit dataset consists in 16×16 pixels images of digits (0 to 9). Only the (respectively 731 and 658) images of digits 2 and 3 are kept (they form the two dominant clusters), as well as 100 (randomly chosen) images of digit 0. We compare the mixtures of PPCAs and of robust PPCAs in their ability to find the two main clusters (thereby identifying the 0 as outliers) and to identify the main variability in these clusters with a one-dimensional latent space. Figure 4 shows sample images close to the one-dimensional subspace. The mixture of robust PPCAs completely ignores the smaller cluster of digits 0. On the other hand, the mixture of PPCAs cannot down-weight the contribution of the digits 0, influencing the two components.

5 Conclusion

This paper introduces the Mixture of Robust Probabilistic PCA. The method is aimed to represent nonlinear manifolds and possibly identify clusters in data. All parameters of the method, with the exception of the number of clusters and the dimensionality of the latent space, are learned through the use of a probabilistic latent formulation, and the optimization of the likelihood of the data. Compared to its non-robust parent, the method shows a strongly reduced sensitivity to outliers, even in high-dimensional spaces.

References

1. Archambeau, C.: Probabilistic Models in Noisy Environments - And their Application to a Visual Prosthesis for the Blind. PhD thesis, Université catholique de Louvain, Louvain-la-Neuve, Belgium (2005)
2. Archambeau, C., Delannay, N., Verleysen, M.: Robust probabilistic projections. In: Cohen, W.W., Moore, A. (eds.) 23rd International Conference on Machine Learning (ICML), pp. 33–40. ACM Press, New York (2006)
3. Archambeau, C., Delannay, N., Verleysen, M.: Mixtures of robust probabilistic principal component analysers. In: ESANN 2007, European Symposium on Artificial Neural Networks Advances in Computational Intelligence and Learning, Bruges, Belgium (2007)
4. Attias, H.: Inferring parameters and structure of latent variable models by variational bayes. In: Proceedings of the 15th Annual Conference on Uncertainty in Artificial Intelligence (UAI-1999), pp. 21–30. Morgan Kaufmann, San Francisco (1999)
5. Bishop, C.M.: Bayesian pca. In: Proceedings of the 1998 conference on Advances in neural information processing systems II, pp. 382–388. MIT Press, Cambridge (1999)
6. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via EM algorithm. *J. Royal Statistical Soc. B* 39(1), 1–38 (1977)
7. Huang, K., Ma, Y., Vidal, R.: Minimum effective dimension for mixtures of subspaces: A robust GPCA algorithm and its applications. In: 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), vol. 2, pp. 631–638 (2004)
8. Peel, D., McLachlan, G.J.: Robust mixture modelling using the t distribution. *Statistics and Computing* 10, 339–348 (2000)
9. Richardson, S., Green, P.: On bayesian analysis of mixtures with an unknown number of components. *J. Roy. Statist. Soc.* 59, 731–792 (1996)
10. Roweis, S.T.: EM algorithms for PCA and SPCA. In: Jordan, M.I., Kearns, M.J., Solla, S.A. (eds.) *Advances in Neural Information Processing Systems 10 (NIPS)*. MIT Press, Cambridge (1998)
11. Shoham, S.: Robust clustering by deterministic agglomeration EM of mixtures of multivariate t -distributions. *Pattern Recognition* 35(5), 1127–1142 (2002)
12. Tipping, M.E., Bishop, C.M.: Mixtures of probabilistic principal component analyzers. *Neural Computation* 11(2), 443–482 (1999)
13. Tipping, M.E., Bishop, C.M.: Probabilistic principal component analysis. *Journal of the Royal Statistical Society B* 61, 611–622 (1999)
14. Waterhouse, S., MacKay, D., Robinson, T.: Bayesian methods for mixtures of experts. In: Touretzky, D.S., et al. (eds.) *Advances in Neural Information Processing Systems*, vol. 8, pp. 351–357. MIT Press, Cambridge (1996)
15. Xu, L., Yuille, A.L.: Robust principal component analysis by self-organizing rules based on statistical physics approach. *IEEE Transactions on Neural Networks* 6(1), 131–143 (1995)

Exploratory Hot Spot Profile Analysis Using Interactive Visual Drill-Down Self-Organizing Maps

Denny^{1,2}, Graham J. Williams^{3,1}, and Peter Christen¹

¹ Department of Computer Science, The Australian National University, Australia
denny@cs.anu.edu.au, peter.christen@anu.edu.au

² Faculty of Computer Science, University of Indonesia, Indonesia

³ The Australian Taxation Office

graham.williams@ato.gov.au

Abstract. Real-life datasets often contain small clusters of unusual sub-populations. These clusters, or ‘hot spots’, are usually sparse and of special interest to an analyst. We present a methodology for identifying hot spots and ranking attributes that distinguish them interactively, using visual drill-down Self-Organizing Maps. The methodology is particularly useful for understanding hot spots in high dimensional datasets. Our approach is demonstrated using a large real life taxation dataset.

Keywords: self-organizing maps, hot spot analysis, attribute ranking, imbalanced data, interactive drill-down visualization.

1 Introduction

The complexity of knowledge contained in large datasets is often easier to explore by grouping similar entities together, which is known as cluster analysis. For example, clustering of customers sharing similar characteristics generally makes it easier to devise marketing strategies. Self-Organizing Maps (SOMs) [1] are popularly used in cluster analysis for several reasons. First, SOMs topologically map high-dimensional data into a two-dimensional map with similar entities being placed close to each other. Second, SOMs produce a smaller but representative dataset that exhibits the distribution of the original dataset. Third, SOMs offer various map visualizations that allow non-technical users to explore a dataset.

In real datasets cluster sizes are normally not equal and clusters do not have the same level of interest for a user. The cluster distribution is often very skewed with interesting clusters being a small fraction of the full dataset. Also, variance of the items at the tail/margin of the normal distribution of a population is also larger compared to the center of the distribution. Thus it is common to find large dense clusters for common sub-populations, and small sparse clusters that might be of interest. In a taxation context, for example, this could be a small

group of tax entities who have unusual tax debts, while in an insurance context this may be a small group of high claiming clients.

Hot Spots aims to identify important or interesting groups in very large datasets [2] using a combination of clustering and rule induction. By understanding attributes that distinguish these small and interesting clusters (hot spots), businesses can improve their processes, such as the choice of treatment strategies for ensuring tax compliance. We advance the hot spots methodology using attribute selection measurement and visualization. With our methodology, analysts can identify and understand distinguishing characteristics of hot spots through interactive visualizations and by performing drill-down exploration.

2 Hot Spots Analysis

Hot Spots data mining identifies key areas in very large datasets that are interesting to an analyst [2]. A dataset is clustered to identify between 10 and 1,000 clusters. Each entity is then labelled with the cluster it is assigned to. Supervised learning (e.g., tree induction) is used to generate distinguishing descriptions for each cluster. The resulting tree is pruned and transformed into a rule set. Finally, the interestingness of the clusters are evaluated. As it is difficult to formalize interestingness, this is domain dependent and therefore, such an analysis is often exploratory and evolutionary [3].

There are several drawbacks with the Hot Spots methodology. When correlated attributes exist in a dataset only one of them will be used in the rule set to describe a cluster, reducing the description of the clusters. Also, the supervised learning step is highly dependent on the results of the previous clustering step, and also on the clustering technique employed (usually k -means). When a large number of clusters is chosen some clusters might have quite similar characteristics, yet a small number of clusters would reduce the required detail extracted from the dataset. Exploring for the right number is difficult.

3 Self-Organizing Maps

A SOM is an artificial neural network that performs unsupervised competitive learning [1]. Importantly, SOMs can be visualized and be used to explore high-dimensional data spaces through a non-linear projection onto a lower-dimensional manifold, most commonly a 2-D plane [4]. Artificial neurons are arranged on a low-dimensional grid, with each neuron represented by an n -dimensional prototype vector (with n the dimension of the input data) and connected to its neighbouring neurons.

Exploring for Hot Spots we find that interesting clusters are usually located at the border of the map because of the topological ordering property. However, SOMs have a *border problem* [4] where the neighbourhood definition is not symmetric at the borders of the map—the number of neighbours per unit on the borders and corners of the map is not equal to the number of neighbours in the

middle of the map. As the density estimation for the border units is different to the units in the middle of the map, the tails of the marginal distributions of variables (normally located at border units) are less well represented than their centers [4]. A visual drill-down approach using a SOM can alleviate this [5]. Here, several nodes of a region can be selected by an analyst for interactive drill down to target regions of interest.

Furthermore, SOMs tend to merge small sparse clusters. This further reduces the detail in the analysis. Increasing the map size of a SOM gives a better resolution map but with significant additional computational cost.

4 SOM Hot Spot Profile Analysis Methodology

The contribution of this paper is the development of a methodology to perform profile analysis of hot spots. We present this as data pre-processing, map training, hot spots identification, profile analysis, drill-down, and sub-map analysis.

4.1 Data Pre-processing and Map Training

Data pre-processing is important prior to training any maps [5]. SOMs only handle numeric attributes—each non-numeric (categorical) attribute is transformed into a set of numeric attributes, encoding each categorical value into a binary indicator (1 or 0). Normalization of the numeric attributes ensures that attributes with larger ranges won't have an unduly larger influence on the distance calculations [6].

Linear initialization is recommended for initialising a SOM, resulting in an order of magnitude improvement in time taken for learning compared to random initialization [4]. Also, we train a SOM in two phases using batch training [4]. This combined linear initialization and batch training produces the same map each time the learning process is repeated (random initialization might produce different orientations of the map). Batch training can also utilize multi-processor environments to speed up the training process. The map size, training length, initial and final radius are chosen by considering a best practice approach [7].

4.2 Identifying Hot Spots in Self-Organizing Maps

Hot spots in SOMs can be identified by two approaches: first by using the distance matrix visualizations and second by analysts' feedback based on component plane visualizations. Noting that entities in hot spots are usually less homogeneous because they are often located at the tail of distributions, these regions can be identified using the distance matrix. Distance-matrix based visualizations, such as u-matrix visualization [8], show distances between neighbouring nodes using a colour scale representation on a map grid. As shown in Fig. 1, white indicates a small distance between a node and its neighbouring nodes while black indicates a large distance between a node and its neighbours [1].

¹ SOM graphics are best in colour but printing requirements necessitated gray scale.

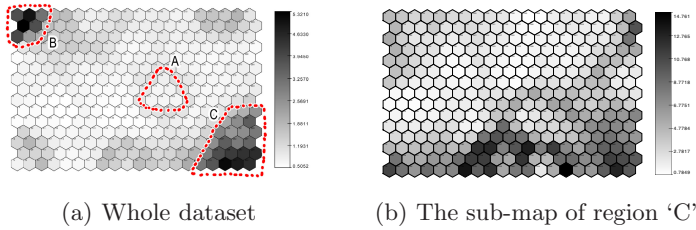


Fig. 1. Distance matrix (median of a node to its neighbours [5]) visualization

The distance matrix visualization can be used to identify borders between clusters. Large distances that show highly dissimilar features between neighbouring nodes divide clusters, i.e. the dense parts of the map with similar features (white regions) [8].

Distance-matrix visualizations can be used to acquire the initial cluster structure of the dataset. By using this visualization, an analyst can see the cluster structure of the dense part of a map. An example is the cluster in the center of the map (marked ‘A’) in Fig. 1(a). However, it is difficult to see the cluster structure of the sparse regions of the lower-right and the upper-left corners of the map (marked ‘B’ and ‘C’).

The distance matrix visualizations in Fig. 1 show homogeneous (low variation) groups with smaller neighbour distances (white regions) and high variation groups (dark regions). Regions with larger neighbour distances can be further investigated through component plane visualizations. In Fig. 1(a) two hot spots are identified according to the above criteria (the regions marked ‘B’ and ‘C’).

4.3 Profile Analysis of Hot Spot

Descriptive statistics (e.g., average values) of entities mapped to a hot spot provides a simple characterization. However, this approach does not provide an analyst with insight, as it is difficult to find the average value with respect to the spread of the values of the whole dataset.

Component plane visualizations can be used to show the spread of values of a certain component of all prototype vectors in a SOM [9]. The value of a component in a node is the ‘average’ value of entities in the node and its neighbours according to the neighbourhood function. The colour coding of the map is created based on the minimum (white) and the maximum values (black) of the component of the map. When analyzing the characteristics of hot spots in high dimensional datasets, it is difficult to identify components which distinguish hot spots from the remaining population by visualizing all component planes, except by ranking their importance, as shown in Fig. 2(a).

An analyst is supported in our methodology by sorting the component planes by the importance of the attributes that distinguish a hot spot from the rest of the population. This ranking can be done using an attribute selection measure [6], such as information gain or gain ratio. As attributes in a SOM are

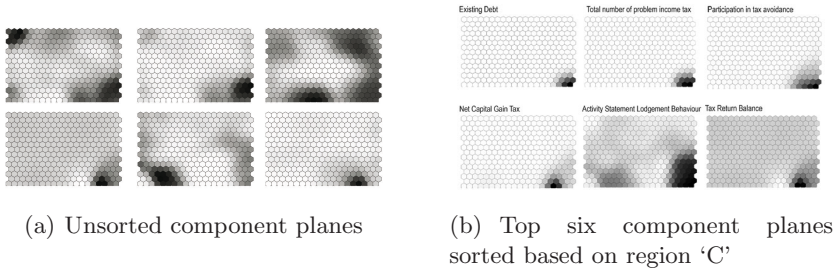


Fig. 2. Component planes. Six of 90 attributes are shown.

numeric, a supervised discretization measure [6], such as entropy-based discretization, should be applied to the numeric attributes before ranking. To rank attributes by their importance, the nodes of the selected region are labeled as ‘hot spot’ and the rest as ‘non-hot spot’. An analyst can then choose an attribute selection measure for attribute importance based on the prototype vectors. The component planes are then ordered by this rank. Fig. 2(b) shows the sorted component planes of the hot spot of region ‘C’ in Fig. 1(a) using the Gain Ratio. With this ordering, an analyst is able to identify the attributes that distinguish a hot spot from the rest of the population.

As a SOM produces a smaller but representative dataset, the prototype vectors can be used as an approximation of the whole dataset. Efficient computation allows an analyst to explore the profile of any region of the map interactively.

4.4 Drill Down and Visualizing Hot Spots

The analyst has chosen the region of the top level map of interest, allowing a sub-map to be trained to gain more detail for these sparse regions. In training the sub-map, consistency of interpretation of the visualization of the sub-map needs to be preserved while maintaining the sub-map quality with respect to the sub-population [5].

For consistent interpretation of the visualization of the sub-map, the orientation of the map is preserved and the colour coding is made consistent [5]. A drawback of using linear initialization for the sub-map based on the entities in the sub-map is that its orientation might be different to the orientation of the top level map. For example, entities located at the bottom-right corner of the top level map might be located at the top-left corner as we drill down, particularly when the two largest principal components of the whole population and the sub-population are different.

We propose that the top level map be used as the initial map of the sub-map [5]. The radius of the rough phase training must be wide enough to avoid subregions of the map becoming empty. We find that setting the initial radius of the rough phase to be half of the longest side and the initial radius of the fine tune phase to be a quarter of the longest side works well.

4.5 Visualization and Analysis of the Sub-map

Sub-maps are also visualized using the distance matrix and component plane visualizations introduced above. To display the distribution of values of the sub-map with respect to the whole population, we use the colour map for the whole population to visualize the component planes of the sub-map. In other words, black in the sub-map visualizations is used for the maximum value of the component of the top level map, not necessarily the maximum value of the component of the sub-map. As the sub-map has better quality in terms of quantization error (more homogeneous within a node), the component value in the sub-map might exceed the maximum value of the top-level map. The colour for such values are also black and this needs to be kept in mind in reviewing the visualization.

With sub-regions consisting of considerably fewer data vectors the training of the sub-map is considerably faster. An analyst is thus able to interactively explore hot spots once the top level map has been trained. The sub-map can be further explored using the methods introduced in Sects. 4.2 and 4.3.

5 Results and Discussion

Our new visual SOM drill-down approach has been applied to the task of exploring taxpayer compliance for the Australian Taxation Office (ATO), using a de-identified taxpayer dataset. Here, we provide aggregate indicative results that demonstrate the effectiveness of our methodology, without breaching the confidentiality of the data or the discoveries made.

The analysis is motivated by the need to understand the logic and structures that drive taxpayers' compliance behaviour (behavioural archetypes). The idea is to construct 'psychographic groups' [10] by using data mining. Understanding the difference between low and high risk taxpayers is important.

The dataset consists of 6.5 million entities with 90 attributes that reflect taxpayer behaviour. The attributes can be categorized into: income profile (details of income sources), propensity to lodge correctly and on time (lodgement profile), propensity to pay (debt profile), market segments, demographics, socio-economic indicators for areas (SEIFA) [11], and participation in tax avoidance schemes. These attributes were selected by domain specialists. The dataset was normalized and categorical attributes were transformed into numerical attributes.

A map size of 15x20 units with a hexagonal lattice structure [4] was chosen. The initial radius of the rough phase was 8 and for the fine tuning phase it was 4. The training length for the rough phase was 6 iterations and for the fine tuning phase 10 iterations. The training of the top-level map took about 5 hours under Debian GNU/Linux with two AMD64 dual-core 3GHz processors and 16 GB memory using our Java-based SOM Toolbox.

In interpreting multiple visualizations it must be understood that the visualizations are linked by position or by colour. A visualization of the same map is linked by position so that the position of each entity remains the same in each visualization. Figs. 1(a), 3(a), and 3(b) are linked by position. The visualization

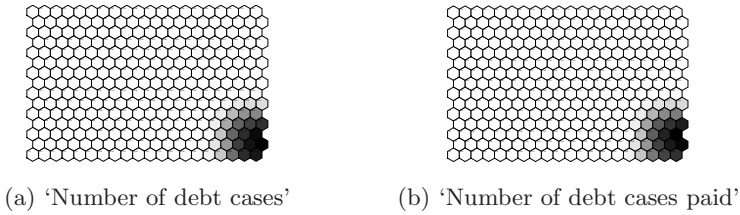


Fig. 3. Component plane of the whole population [5](#)

of the top-level map is linked by colour to the sub-map so that the colours of the top level map are directly used for the sub-maps.

The visualization of the dataset distance matrix can be seen in Fig. [1\(a\)](#). The ‘common’ population in real life datasets is usually located in the center of a map. In Fig. [1\(a\)](#), the entities in the center of the map of the whole population are relatively homogeneous. According to the criteria presented in Sect. [4.2](#), there are two hot spots, located in the top-left corner (‘B’) and in the bottom-right corner (‘C’). Based on the ranking of the component planes (Sect. [4.3](#)) using gain ratio as the attribute selection measure, hot spot ‘C’ can be distinguished by the following attributes in decreasing importance: existing debt, total number of problem income tax returns, participation in tax avoidance schemes, net capital gain tax, activity statement lodgement behaviour, and the balance of the tax return (Fig. [2\(b\)](#)). Hot spot ‘B’ can be distinguished by the attributes: allowances, dividends, and total income. Based on these rankings, ‘C’ is more interesting, and further explored.

The entities in ‘C’ have highly dissimilar characteristics (Fig. [1\(a\)](#)). However, at this level, it is difficult to differentiate the debt behaviour, as shown in Figs. [3\(a\)](#) and [3\(b\)](#). Therefore, to see the debt behaviour in detail, we drill down into the lower-right corner of the top level map (Sect. [4.4](#)).

At this level we can also use the distance matrix visualization (Fig. [1\(b\)](#)) to highlight the hot spots in this sub-map, which are located along the bottom of the map. It is also interesting to note that the hot spot of the sub-map consists of entities that are involved in tax avoidance activities. Furthermore, this group has characteristics of longer debt age, higher levels of compliance enforcement, and lower percentage of cases paid.

6 Conclusion and Future Work

We have introduced a methodology for understanding characteristics of hot spots in large real world datasets, such as from the taxation domain. Based on our experiments, the methodology is effective for hot spots exploration, offering interactive visualizations that are easy to understand. An analyst is able to identify discriminating characteristics of hot spots. As a SOM produces a considerably smaller-sized set of prototype vectors, it allows an efficient use of attribute selection measurements. In using the methodology introduced here analysts have

the flexibility to explore regions or clusters based on map visualization, and are able to drill-down into sparse regions or clusters. Analysts are now able to select regions or clusters based on their business needs.

This work is part of a larger research project where we are interested in observing the dynamics of hot spots over time, such as to find entities who are moving in or out of hot spots. Such knowledge will be valuable as an analyst can derive strategies to encourage or deter moves in or out of the hot spots (which might be regions of non-compliance or of high compliance). It can also be used to evaluate the effectiveness of such business strategies over time.

Acknowledgement

This research has been supported by the Australian Taxation Office. The authors express their gratitude to Grant Brodie, Georgina Breen, Nicole Wade, and Warwick Graco for providing data and domain expertise.

References

1. Kohonen, T.: Self-organized formation of topologically correct feature maps. *Biological Cybernetics* 43, 59–69 (1982)
2. Williams, G.J., Huang, Z.: Mining the knowledge mine: The hot spots methodology for mining large real world databases. In: Sattar, A. (ed.) *Canadian AI 1997*. LNCS, vol. 1342, pp. 340–348. Springer, Heidelberg (1997)
3. Williams, G.J.: Evolutionary hot spots data mining - an architecture for exploring for interesting discoveries. In: Zhong, N., Zhou, L. (eds.) *PAKDD 1999*. LNCS (LNAI), vol. 1574, pp. 184–193. Springer, Heidelberg (1999)
4. Kohonen, T.: *Self-Organizing Maps*, 3rd edn. Springer Series in Information Sciences, vol. 30. Springer, Heidelberg (2001)
5. Denny, Williams, G.J., Christen, P.: Exploratory multilevel hot spot analysis: Australian Taxation Office case study. In: *AusDM 2007*, Gold Coast, Australia, ACS. CRPIT, vol. 70, pp. 73–80 (2007)
6. Han, J., Kamber, M.: *Data Mining: Concepts and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2006)
7. Vesanto, J., Himberg, J., Alhoniemi, E., Parhankangas, J.: SOM toolbox for Matlab 5. Report A57, Helsinki University of Technology, Neural Networks Research Centre, Espoo, Finland (April 2000)
8. Iivarinen, J., Kohonen, T., Kangas, J., Kaski, S.: Visualizing the clusters on the Self-Organizing Map. In: *Proceedings of the Conference on AI Research in Finland*, vol. 12, pp. 122–126, Helsinki, Finland, Finnish AI Society (1994)
9. Tryba, V., Metzen, S., Goser, K.: Designing basic integrated circuits by Self-Organizing Feature Maps. In: *International Workshop on Neural Networks and their Applications*, Nanterre, France, ARC, SEE, EC2, November 1989, pp. 225–235 (1989)
10. Wells, W.D.: Psychographics: A critical review. *Journal of Marketing Research (JMR)* 12(2), 196–213 (1975)
11. Trewin, D.: Socio-economic indexes for areas: Australia 2001. Technical Report 2039, Australian Bureau of Statistics (2003)

Maintaining Optimal Multi-way Splits for Numerical Attributes in Data Streams

Tapio Elomaa and Petri Lehtinen

Department of Software Systems, Tampere University of Technology
P.O. Box 553 (Korkeakoulunkatu 1), FI-33101 Tampere, Finland
elomaa@cs.tut.fi, petri.lehtinen@tut.fi

Abstract. In the batch learning setting it suffices to take into account only a reduced number of threshold candidates in discretizing the value range of a numerical attribute for many commonly-used attribute evaluation functions. We show that the same techniques are also efficiently applicable in the on-line learning scheme. Only constant time per example is needed for determining the changes on data grouping. Hence, one can apply multi-way splits, e.g., in the standard approach to decision tree learning from data streams. We also briefly consider modifications needed to cope with drifting concepts. Our empirical evaluation demonstrates that often the reduction in threshold candidates obtained is high for the important attributes. In a data stream logarithmic growth in the number of potential cut points and the reduced number of threshold candidates is observed.

1 Introduction

By an *on-line learning model* one refers to a learning model in which the training examples arrive on-line. One usually assumes that the examples are received one at a time. However, it is common to update the hypothesis and compute the required sufficient statistics not following each new example, but only after gathering a small set of new examples [1,2]. Thus, the data stream could also be bursting. In any case, the stream can be considered to be infinite, because it is unimaginable that all data received could be stored into the main memory of a computer.

One of the basic knowledge representations of machine learning, *decision trees*, is among the first formalisms learning of which has been studied in this setting [1,2,3,4,5]. The streaming data received contains successive training examples each of which consists of values for a pair of random variables $\langle \mathbf{X}, Y \rangle$. The k elements of the instance vector \mathbf{X} are called *attributes*; $\mathbf{X} = \langle A_1, \dots, A_k \rangle$. An attribute may be nominal- or continuous-valued (numerical). The class labels Y usually come from a small nominal set. The aim is to maintain an adaptive anytime model of determining the value of Y based on the attribute values \mathbf{X} . One is allowed to process the data only in the order that it arrives without storing (all of) it. As the intention is to operate in real time, only constant time per example may be used to update the statistics sufficient to determine which attribute is the appropriate one to be tested in a node of the evolving tree.

Usually there is a training phase in which a model of the data is built based on examples, and an application phase in which the model is used to classify

instances whose class is not known. These phases can also overlap and, therefore, we want to have an anytime model. Also the case in which there is no single final concept to be learned, but it changes over time, has been tackled [35].

In decision tree learning an *attribute evaluation function* is used to decide which attribute's value to test in a node of the evolving tree. For a nominal attribute one simply grows a subtree for each of the few separate values that the attribute may take. Numerical attributes with (infinitely) many possible values, on the other hand, need to be processed somehow. It is common to discretize the continuous value range into a small number of disjoint intervals. We will demonstrate that one can efficiently maintain sufficient statistics for obtaining an optimal multi-way split for the value range of the attribute in question in the on-line data stream model even if the data cannot be stored. Optimality in this context means that we need to guarantee that the attribute evaluation function can attain its best value on the data even after the modifications done.

Multi-way splitting of numerical attributes has not often been applied in the streaming data context. Previous work has mostly relied on recursive binarization of the numerical value range. The notable exception is the recent work of Gama and Pinto [6] in which two-layer histogram discretization was introduced. Gama et al. [2] also applied multi-way splits consisting of ten equal-width bins in the leaves of a tree in their VFDTc system. The leaves in VFDTc decision trees are functional—i.e., contain a naïve Bayes classifier—but the internal nodes use the standard binarization approach for numerical attributes.

Our work continues the *research line* of optimal splitting originally initiated by Fayyad and Irani [7]. This line of research has been continued later in the batch learning setting [8,9,10,11]. We will recapitulate the main ideas of this work in Section 3. In the data stream model Jin and Agrawal [4] have examined reducing the number of cut point candidates that need to be taken into account.

2 Learning Decision Trees from Data Streams

Domingos and Hulten [1] introduced the VFDT system, which learns *efficient* *decision trees*—decision trees with a similarity guarantee to those learned by conventional batch algorithms such as CART [12] and C4.5 [13]. The standard Hoeffding inequality is used to show that the attribute chosen to a node in a tree is, with a high probability, the same as the one that would have been chosen by the batch learner with access to all of the data. VFDT chooses an attribute to the root based on n first examples, after which the process continues recursively in the leaves down to which the succeeding examples are passed. Hoeffding bounds allow to solve the required n for reaching the user-requested confidence level.

Domingos and Hulten [1] did not elaborate on how to handle numerical attributes in Hoeffding trees. They just proposed that the commonly-used thresholded binary splitting of the numerical value range be applied; i.e., only tests of the form $A_i < t_j$ are used. The value range of a numerical attribute may get multi-way splitted through subsequent binary splits of the induced subintervals.

Gama et al. [2] put forward an instantiation of VFDT in which *the* *function* () of C4.5 was used to evaluate attributes. For each numerical

attribute A_i a (balanced) binary search tree (BST) is maintained. It records for every potential threshold t_j the class distribution of the binary partition induced by the test $A_i < t_j$. Exhaustive evaluation over the tests is used to choose the one to be placed to the evolving decision tree. However, all threshold candidates for one attribute can be evaluated during a single traversal of the BST.

Obviously, updating the BST takes $O(\lg V)$ time, where V is the number of different values that the attribute in question takes. This price needs to be paid in order to be able to choose the best binary split. One has to sort the values and—as the value range is unknown from the outset—a time proportional to the number of potential cut points needs to be paid per example, unless one is willing to risk finding the optimal cut point.

Jin and Agrawal [4] proposed an approach for pruning intervals from the range of a numerical attribute. They first discretize a numerical value range into equal-width intervals after which a statistical test decides which intervals appear unlikely to include a split point and can, thus, be pruned. In addition they showed that Hoeffding bounds can be reached for χ^2 function [12] with a lesser number of samples than in the original VFDT.

Gama et al. [2] also used functional leaves in the tree instead of the simple majority class strategy. Before the algorithm has decided which attribute test to assign to a leaf in the evolving tree, Naïve Bayes can be used to give predictions for instances that arrive needing to be classified. For numerical attributes the common approach of discretization into ten equal-width bins (when possible) is used in the naïve Bayes classifiers.

CVFDT [3] adapts the VFDT system to concept drift. With the concept changing over time, it is necessary to incrementally update the model built for the examples. The real-time operation requirement does not allow to rebuild the model for examples in a sliding window from scratch. Instead, Hulten et al. [3] proposed to build an alternative subtree for those nodes that do not pass the Hoeffding test in light of the sufficient statistics maintained for a sliding window of examples. When the alternate subtree's performance on new examples overtakes that of the old one, it is inserted to the tree. To grow the shadow tree one uses the standard techniques of VFDT, thus ensuring real-time operation.

In the UFFT system [5] concept drift is detected through the reducing accuracy of the naïve Bayes classifier installed into an internal node. Whenever, a change in the target concept is identified, the subtree rooted at the corresponding node and its associated statistics is pruned into a (functional) leaf, and building of a new subtree may begin anew. Numerical attributes are handled using the common normality assumption. The sufficient statistics for each numerical attribute in this case are simply the mean and variance per class.

Gama and Pinto [6] use histograms for data stream discretization. They induce either an equal-width or an equal-frequency discretization for the value range. The approach uses two layers of intervals; layer 1 maintains statistics for an excessive number of intervals and layer 2 composes the final discretization based on these statistics. Layer 1 is maintained on-line by updating the counters in the appropriate interval whenever a new example is received. If a user-defined

condition is met, an interval may be split in two; e.g., to keep the intervals of approximately equal width. Layer 2 produces, on need basis, the final histograms by merging intervals of layer 1. Building of the second level discretization is confined on the cut points of layer 1, and may thus be inexact.

3 Optimal Multi-way Splits for Numerical Attributes

The shortcomings of recursive binary splitting of the value range of a numerical attribute could potentially be avoided if one-shot multi-way splits were used instead [14][15]. However, multi-way splitting can be computationally expensive. Hence, the most popular approaches are based on multi-way splitting through successive binary splits [14][15]. Such partitions cannot, though, be guaranteed to be optimal with respect to the attribute evaluation function being used.

Without loss of generality, let us consider only one real-valued attribute X . A training sample $S = \{(x_1, y_1), \dots, (x_n, y_n)\}$ consists of n labeled examples. For each $(x, y) \in S$, $x \in \mathbb{R}$ and y is the class label of x in $C = \{c_1, \dots, c_m\}$. A k -interval discretization of the sample is generated by picking $k - 1$ or $T_1 < T_2 < \dots < T_{k-1}$. Let $T_0 = -\infty$ and $T_k = \infty$, then the set of $k - 1$ thresholds defines a partition $\biguplus_{i=1}^k S_i$ of the set S so that $S_i = \{(x, y) \in S \mid T_{i-1} < x \leq T_i\} \neq \emptyset$ for all $1 \leq i \leq k$.

The simplest attribute evaluation function is Training Set Error (. . .). Let $\delta_j(S_i)$ denote the . . . , or the . . . , with respect to class c_j in the set S_i . That is, if all instances in S_i were predicted to belong to class c_j , we would make $\delta_j(S_i)$ errors. Furthermore, let $\delta(S_i)$ denote the minimum error on S_i . A class $c_j \in C$ is one of the . . . of S_i , if predicting class c_j leads to minimum number of errors on S_i , i.e., $\delta_j(S_i) = \delta(S_i)$.

Given a k -interval partition $\biguplus_{i=1}^k S_i$ of S , where each interval is labeled by a majority class, its . . . is the minimum number of training instances falsely classified in the partition. The . . . minimum error discretization problem is to find a partition $\biguplus_{i=1}^k S_i$ that has the minimum attribute evaluation function value over all partitions of S . The maximum number of intervals k may also be given as a parameter. Then the problem is to find the optimal partition among those that have at most k intervals. This is called . . . discretization.

If one could make its own partition interval out of each data point, this discretization would have zero training error. However, one cannot discern between all data points. Only those that differ in their value of X can be separated from each other. E.g., in the data set shown in Fig. 1 there are 27 integer-valued instances of two classes; α and β . Interval thresholds can only be set in between those points where the attribute value changes. Therefore, one can process the data into . . . , one for each existing attribute value. Within each bin we record its class distribution. This information suffices to evaluate the goodness of the partition; the actual data set does not need to be maintained.

The sequence of bins has the minimal . . . misclassification rate for However, the same rate can usually be obtained with a smaller number of intervals. Fayyad and Irani's [7] analysis of the entropy function has shown that

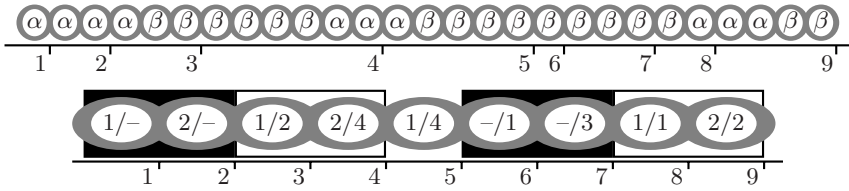


Fig. 1. A sequence of examples sorted according to their numerical values (above). The class labels (α and β) of the instances are shown. The sequence of data bins with their respective class distributions (below). The blocks (black rectangles) and segments (black and white rectangles) in the sample.

cut points embedded into class-uniform intervals need not be taken into account, only the end points of such intervals—the \dots —need to be considered to find the optimal discretization. Thus, optimal splits of \dots and \dots fall on boundary points. Hence, only they need to be examined in optimal binary partitioning of the value range of a numerical attribute.

Elomaa and Rousu [10] showed that the same is true for many commonly-used evaluation functions. By this analysis we can merge together adjacent class uniform bins with the same class label to obtain example \dots (see Fig. 1). The boundary points of the value range are the borders of its blocks. Block construction still leaves all bins with a mixed class distribution as their own blocks. A dynamic programming algorithm lets one find optimal arity-restricted multi-way partitions efficiently in these cases [8,9,10].

Subsequently, a more general property was also proved for some evaluation functions [11]: \dots —points that lie in between two adjacent bins with different relative class distributions—are the only points that need to be taken into account. It is easy to see that segment borders are a subset of boundary points. Example \dots are easily obtained from bins by merging together adjacent bins with the same relative class distribution (see Fig. 1).

4 Maintaining Sufficient Statistics On-Line

Let us now consider how bins, blocks, and segments may change when new examples are received from a data stream. We need to maintain exact counts on instances of different classes observed for each numerical attribute. Prior to observing the data we do not have any knowledge of the value range of a numerical attribute; what are its extreme values?, which values in the range are actually observed?, etc. Therefore, it is necessary to maintain a BST recording the observed values and class counts for each numerical attribute as proposed by Gama et al. [2]. The worst-case update time requirement per example is $O(\lg V)$, where V is the number of bins (observed so far). Truly constant-time update would require giving up on recording exact bin counts (cf. [6]).

Effectively, the BST sorts the observed examples. Hence, we have access to bins and through them to segments also within the streaming data model. In the

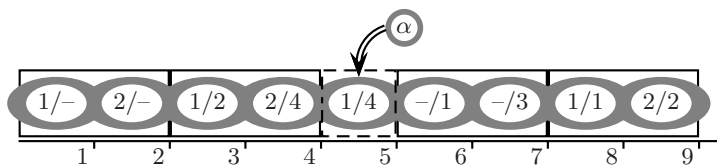


Fig. 2. A new example changes the class distribution in one of the bins leading to the bin being merged together with an adjacent (the left one) segment. Cf. Fig. 1

BST of Gama et al. [2] class frequencies are stored into the internal nodes of the tree. They only want to find the optimal binary split of the value range. Hence, it suffices to know the class distributions at both sides of the split. We, on the other hand, need to know class frequencies for all bins, which is easier to implement by storing relevant information only to the leaves. This modification does not change the asymptotic time requirement of BST processing; the maximum path length remains at $O(\lg V)$. For segments a linked list with appropriate pointers to and from the BST is a suitable data structure.

Bins are atomic intervals in multi-way splits — they cannot be divided further in univariate discretization. A new example received from the data stream can fall into one of the existing bins, in which case its class distribution changes, except when the bin happens to be class uniform and the new example is of the same class. In any case the bin counts (in the BST) need to be updated. Otherwise, the new example falls outside of the existing bins and makes up a new bin with a trivially uniform class distribution. The new bin may also be the first or the last one observed in the value range.

We can consider bins to correspond to the first layer of intervals in Gama and Pinto’s [6] histogram discretization. However, for exact bin counting we need to maintain the BST instead of being able to just use a simple matrix. For anytime prediction, we need to cover the whole value range of a numerical attribute through binning. In particular, empty intervals are not permitted. Therefore, when we have observed values only sparsely from the range, we need to extend the actual bins to cover the whole range. Such intervals, of course, may get divided contrary to actual bins.

Blocks and segments, then, correspond to the second layer of intervals which can be constructed by merging together intervals of the first layer. In the following let us talk about segments instead of blocks and segments. Both are class coherent in any case and blocks are a special (uniform) case of segments.

What changes do updates on bins bring to segments? Let us consider a bin that made up a segment by itself without being merged together with any of the other bins. The changing class distribution of the bin makes it a candidate for being merged together with one of its neighbor segments. Hence, the distribution in the two adjacent segments needs to be checked. In the best case, the two neighbors and the changing bin all three merge together to make up a new segment. However, the merging cannot propagate further as only the middle bin’s class distribution has changed. The differences that existed between the adjacent segments and their neighbors remain unchanged (see Fig. 2).

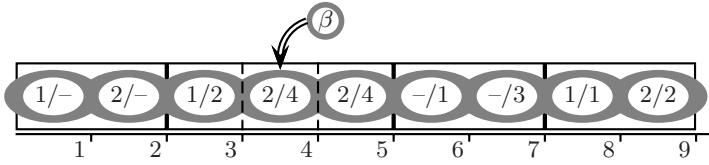


Fig. 3. A new example changes the class distribution in one of the bins leading to the splitting of the segment into which the bin used to belong. Cf. Fig. 2

In the second scenario the bin that receives the new example belongs to a segment. As the bin’s class distribution now changes, it can no longer belong to the same segment as before. Therefore, the bin needs to be taken out of the segment. If it was in the middle of the segment, the segment breaks into three new ones (see Fig. 3). On the other hand, if the bin was the head or the tail bin of the segment, we have to check whether it can be merged together with its other neighbor (if one exists) now that the class distribution has changed.

Because bins are extended to cover the whole value range, a new example cannot actually fall outside of the existing intervals. The interval that receives the previously unseen attribute value, though, splits into two (extended) bins and the relation of class distribution in these two with their adjacent segments needs obvious checking.

In summary, in all the possible cases only local changes to segments are needed due to receiving a new example from the data stream; at most two adjacent segments have to be examined. Hence, the required updates only take a constant time. Prior to these changes, though, the example is directed down the BST at the cost of $O(\lg V)$. Our subsequent experiments will also examine the significance of this cost in real-world domains.

Quite often there is no single concept to track from the data stream, but rather the target changes over time [3]. Then it does not suffice to keep stacking incremental changes to the decision tree, but at some point one needs to forget old examples that are not instances of the current concept. The simplest approach is to have a sliding window of length W of the most recent examples and maintain a decision tree consistent with them. The overhead for using such a window is constant. Let us consider this scenario without paying attention to details of window length selection and updating. We only consider what changes from the point of view of bin and segment maintenance.

The straightforward approach is to delete from the BST the oldest example in the window before (or after) inserting a new example to the BST. Deletion of an example causes similar changes to bins and segments as insertion of an example. Hence, the deletion can be handled with local changes in constant time. Of course, we now traverse the BST twice doubling the time requirement of an update. Asymptotically, though, updates are as efficient as in single concept decision trees. Finally, let us point out that in a window of length W there can be at most W different values for a numerical attribute. Thus, in this scenario the maximum overhead for using the BST is a constant of the order $O(\lg W)$.

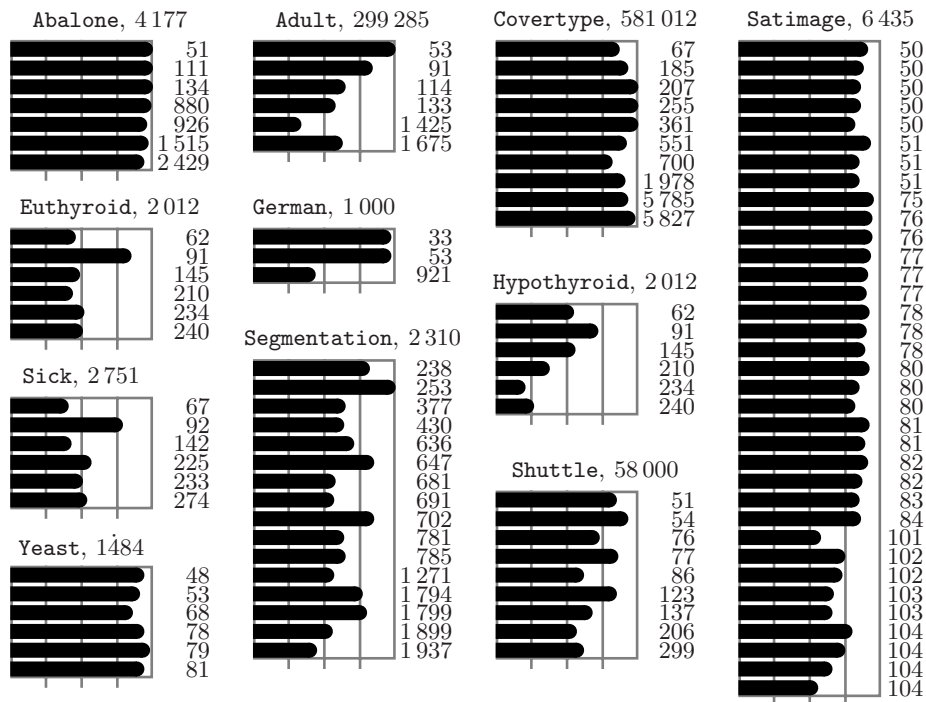


Fig. 4. The relative fraction of segments (bars) out of the bins observed for numerical attributes (the figures on right) for eleven UCI domains. The figures on top denote the number of training examples (without missing numerical values) in the domain.

5 Empirical Evaluation

Let us now experiment with some larger data sets from the UCI repository which contain truly numerical values. Some data sets — like the letter and digit recognition domains — include seemingly numerical attributes, which in reality are nominal ones. In our evaluation we disregard attributes labeled as numerical if they have less than ten different values. We also overlook those examples that have missing values for numerical attributes. We report results individually for all numerical attributes rather than consider the average results (cf. [11]).

Fig. 4 shows for eleven UCI domains the reduction in cut points obtained by moving from bins to segments. The figures on right are the attributes’ bin counts and the bars represent the relative fraction of resulting segments. It is immediate that truly continuous attributes are rare. Usually the number of bins is (clearly) less than 10% of the number of examples. The exceptions are found from domains *Abalone*, *German*, and *Segmentation*. The highest BST search cost that these segment counts yield is $\lg 5\,827 \approx 12.5$ (*Coverttype*).

The reduction percentage varies from 80% to 0%. Usually for attributes with a high number bins large reductions are obtained by moving to operate on

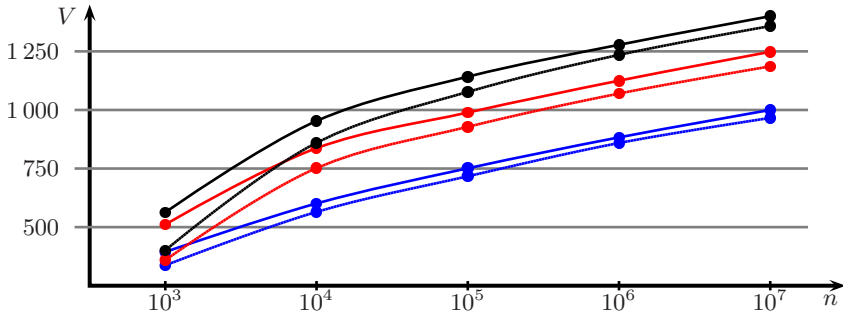


Fig. 5. Evolution of bin and segment counts for three attributes of *Waveform* as more data is generated. Bin counts are denoted by the solid curves and the dashed ones under them are the related segment counts. Note the logarithmic scale of the x -axis.

segments (see e.g., *Adult* and *Segmentation*). However, in domains *Abalone*, *Covertype*, and *Yeast* only quite small reductions are recorded for all attributes.

The domains *Euthyroid* and *Hypothyroid* are equivalent except for labeling of examples. Hence, the attribute bin counts are the same for these two domains. Nevertheless, there are notable differences in the numbers of segments in the two domains. This, of course, follows from the fact that bins only depend on attribute values, while segments also depend on the class distribution.

Our second test monitors for the change of bins and segments using the *Waveform* data generator [12]. Fig. 5 displays for three attributes the evolution of the number of bins and segments when the number of examples grows from one thousand to ten million. The curves eventually stabilize to be more or less linear. Because the x -axis is in logarithmic scale, the true growth rate for bins and segments is also logarithmic. Hence, this experiment indicates that the cost of using the BST to sort the examples by their attribute values is only of the doubly logarithmic order in the number of examples (c. $10 \approx 2 \lg \lg 10^7$ for our largest stream size). Segments slightly lose their advantage as more and more examples are received; the relative fraction of segments grows closer to the number of bins. Quite understandably, when segments contain many examples, the probability that two adjacent bins have the same class distribution reduces.

6 Conclusion

This work has extended the applicability of cut point analysis [7] to the streaming data context. However, our method only allows to solve the unbounded case efficiently. When an arity bound needs to be enforced, the approach would seem to require quadratic processing [8,9,10]. Some form of relaxation is needed to overcome this computational burden.

Our empirical evaluation showed that for some attributes blocks and segments can be of much help with only meager costs. On the other hand, some attributes — even whole domains — resist taking advantage of cut point analysis.

Acknowledgments

This work has been supported by Academy of Finland projects INTENTS (206280), ALEA (210795), and “Machine learning and online data structures” (119699).

References

1. Domingos, P., Hulten, G.: Mining high-speed data streams. In: Proc. 6th ACM SIGKDD Conf. on Data Mining and Knowl. Discovery, pp. 71–80. ACM Press, New York (2000)
2. Gama, J., Rocha, R., Medas, P.: Accurate decision trees for mining high-speed data streams. In: Proc. 9th ACM SIGKDD Conf. on Data Mining and Knowledge Discovery, pp. 523–528. ACM Press, New York (2003)
3. Hulten, G., Spencer, L., Domingos, P.: Mining time-changing data streams. In: Proc. 7th ACM SIGKDD Conf. on Data Mining and Knowledge Discovery, pp. 97–106. ACM Press, New York (2001)
4. Jin, R., Agrawal, G.: Efficient decision tree construction for streaming data. In: Proc. 9th ACM SIGKDD Conf. on Data Mining and Knowledge Discovery, pp. 571–576. ACM Press, New York (2003)
5. Gama, J., Medas, P., Rodrigues, P.: Learning decision trees from dynamic data streams. In: Proc. 2005 ACM Symp. on Appl. Comput., pp. 573–577. ACM Press, New York (2005)
6. Gama, J., Pinto, C.: Discretization from data streams: Applications to histograms and data mining. In: Proc. 2006 ACM Symp. on Applied Computing, pp. 662–667. ACM Press, New York (2006)
7. Fayyad, U.M., Irani, K.B.: On the handling of continuous-valued attributes in decision tree generation. *Mach. Learn.* 8, 87–102 (1992)
8. Fulton, T., Kasif, S., Salzberg, S.: Efficient algorithms for finding multi-way splits for decision trees. In: Proc. 12th ICML, pp. 244–251. Morgan Kaufmann, San Francisco (1995)
9. Zighed, D., Rakotomalala, R., Feschet, F.: Optimal multiple intervals discretization of continuous attributes for supervised learning. In: Proc. 3rd Intl. Conf. on Knowledge Discovery and Data Mining, pp. 295–298. AAAI Press, Menlo Park (1997)
10. Elomaa, T., Rousu, J.: General and efficient multisplitting of numerical attributes. *Mach. Learn.* 36, 201–244 (1999)
11. Elomaa, T., Rousu, J.: Efficient multisplitting revisited: Optima-preserving elimination of partition candidates. *Data Mining Knowl. Discovery* 8, 97–126 (2004)
12. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth (1984)
13. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1993)
14. Catlett, J.: On changing continuous attributes into ordered discrete attributes. In: Kodratoff, Y. (ed.) *EWISL 1991*. LNCS, vol. 482, pp. 164–178. Springer, Heidelberg (1991)
15. Fayyad, U.M., Irani, K.B.: Multi-interval discretization of continuous-valued attributes for classification learning. In: Proc. 13th Intl. Joint Conf. on Artificial Intelligence, pp. 1022–1027. Morgan Kaufmann, San Francisco (1993)

Efficient Mining of High Utility Itemsets from Large Datasets

Alva Erwin¹, Raj P. Gopalan¹, and N.R. Achuthan²

¹ Department of Computing, ² Department of Mathematics and Statistics
Curtin University of Technology, Kent St. Bentley Western Australia
alva.erwin@postgrad.curtin.edu.au, r.gopalan@curtin.edu.au,
n.r.achuthan@curtin.edu.au

Abstract. High utility itemsets mining extends frequent pattern mining to discover itemsets in a transaction database with utility values above a given threshold. However, mining high utility itemsets presents a greater challenge than frequent itemset mining, since high utility itemsets lack the *anti-monotone* property of frequent itemsets. Transaction Weighted Utility (TWU) proposed recently by researchers has *anti-monotone* property, but it is an overestimate of itemset utility and therefore leads to a larger search space. We propose an algorithm that uses TWU with pattern growth based on a compact utility pattern tree data structure. Our algorithm implements a parallel projection scheme to use disk storage when the main memory is inadequate for dealing with large datasets. Experimental evaluation shows that our algorithm is more efficient compared to previous algorithms and can mine larger datasets of both dense and sparse data containing long patterns.

Keywords: High Utility Mining, Pattern Growth.

1 Introduction

The goal of frequent itemset mining [1] is to find items that co-occur in a transaction database above a user given frequency threshold, without considering the quantity or weight such as profit of the items. However, quantity and weight are significant for addressing real world decision problems that require maximizing the utility in an organization. The high utility itemset mining problem is to find all itemsets that have utility larger than a user specified value of minimum utility. Yao et al [2, 3] proposed a framework for high utility itemset mining. Recent research has focused on improving the efficiency of high utility mining. Liu et al. [4] proposed a *TwoPhase* algorithm based on *Apriori* [1] to mine high utility itemsets, using a transaction weighted utility (TWU) measure to prune the search space. Their algorithm is suitable for sparse data sets with short patterns. We recently developed an algorithm named *CTU-Mine* [5] based on the pattern growth approach [6] that was efficient on dense data with relatively longer patterns. In [7], we proposed an algorithm named *CTU-PRO* for efficient mining of both dense and sparse data sets that fit into main memory.

In this paper, we propose an algorithm named *CTU-PROL* for mining high utility itemsets from large datasets using the pattern growth approach [6]. The algorithm first identifies the large TWU items in the transaction database and if the dataset is relatively

small, it creates a *Compressed Utility Pattern Tree (CUP-Tree)* for mining high utility itemsets. For data sets too large to be held in main memory, the algorithm creates subdivisions using parallel projections that can be subsequently mined independently. For each subdivision, a *CUP-Tree* is used to mine the complete set of high utility itemsets. The *anti-monotone* property of TWU is used for pruning the search space of subdivisions in *CTU-PROL*, but unlike *TwoPhase* of Liu et al. [4], our algorithm avoids a rescan of the database to determine the actual utility of high TWU itemsets. The performance of *CTU-PROL* is compared against the implementation of the *TwoPhase* algorithm [4] available from [8] and also with *CTU-Mine* [5]. The results show that *CTU-PROL* outperforms previous algorithms on both sparse and dense datasets at most support levels.

2 Terms and Definitions

In this Section, we define the basic terms of high utility itemset mining based on [1, 2, 9]. Let $I = \{i_1, i_2, \dots, i_m\}$ be a set of items and $D = \{T_1, T_2, \dots, T_n\}$ be a transaction database where the items of each transaction T_i is a subset of I . The quantity of an item i_p in a transaction T_q is denoted by $o(i_p, T_q)$. The external utility $s(i_p)$ is the value of a unit of item i_p in the utility table, (e.g., profit per unit). The utility of item i_p in transaction T_q , denoted by $u(i_p, T_q)$ is defined as $o(i_p, T_q) \times s(i_p)$. A set X is called an itemset if X is a subset of I . The utility of X in transaction T_q , denoted by $u(X, T_q)$ is defined as:

$$u(X, T_q) = \sum_{i_p \in X} u(i_p, T_q) \quad (1)$$

The utility of itemset X in the database, denoted by $u(X)$ is defined as:

$$u(X) = \sum_{T_q \in D \wedge X \subseteq T_q} u(X, T_q) \quad (2)$$

The task of high utility itemset mining is to find all itemsets that have utility above a user-specified *min_utility*. Since utility is not *anti-monotone*, Liu et al. [4] proposed the concepts of Transaction Utility (TU) and Transaction Weighted Utility (TWU) to prune the search space of high utility itemsets. Transaction Utility of a transaction, denoted $tu(T_q)$ is the sum of the utilities of all items in T_q :

$$tu(T_q) = \sum_{i_p \in T_q} u(i_p, T_q) \quad (3)$$

Transaction Weighted Utility of an itemset X , denoted as $twu(X)$ is the sum of the transaction utilities of all the transactions containing X :

$$twu(X) = \sum_{T_q \in D \wedge X \subseteq T_q} tu(T_q) \quad (4)$$

As shown in [4], any superset of a low TWU itemset is also a low TWU itemset, and so we can prune all supersets of low TWU itemsets. However, since TWU is an over-estimation of the real utility value, high TWU itemsets will need to be pruned further. Consider a transaction database of a retailer as shown in Fig. 1(a), with external utility of each item as in Fig. 1(b). The values in each row of Fig. 1(a) shows the quantities of items bought in a transaction, and the last column contains the transaction utility with the total transaction utility of the database shown in the last row. From this sample data, we can compute, $u(3\ 4, t_1) = \$60$, $u(3\ 4, t_3) = \$60$, $u(3\ 4, t_5) = \$60$, $u(3\ 4) = \$180$ and $twu(3\ 4) = \$262$.

TID	1	2	3	4	5	6	TU.
t_1	2	0	1	1	0	0	80
t_2	2	1	1	0	0	0	195
t_3	0	0	1	1	10	0	110
t_4	0	1	0	0	15	0	225
t_5	1	0	1	0	0	1	37
t_6	2	0	0	1	10	0	105
t_7	2	0	0	0	8	1	62
t_8	1	1	0	1	2	0	205
t_9	1	0	0	1	10	0	95
t_{10}	1	1	0	0	5	0	185
total	12	4	4	5	60	2	1299

(a) Transaction Database

		Profit (\$)
1	Printer Ink	10
2	Colour Laser Printer	150
3	Bubble Jet Printer	25
4	Digital Camera	35
5	Glossy Photo Paper	5
6	Floppy Disk	2

(b) Utility Table

Fig. 1. An example transaction database and utility table

3 Mining High Utility Itemsets in Large Datasets

In this section, we describe the *CTU-PROL* algorithm for mining large datasets, consisting of the following steps: (1) Create a *GlobalItemTable* by scanning the database to identify items of high TWU, (2) Subdivide the database by parallel projection of transactions with high TWU items, and (3) Mine each subdivision for high utility itemsets after constructing a Compressed Utility Pattern-Tree (*CUP-Tree*) for the subdivision.

3.1 Creating Global Item Table

A *GlobalItemTable* is constructed as explained in [5] and maps the item-ids to integers in the descending order of their TWU values. Fig. 2 gives the *GlobalItemTable* for the database of Fig. 1, with a minimum utility of 10% of the total transaction utility (=129.9). Note that item 6 with TWU of 99 is pruned. The mapped new index of 1 to 5 correspond to the original items 5, 1, 2, 4, and 3 respectively. The terms mapped item id and item index are used synonymously in the rest of the paper.

GlobalItemTable

Item index	1	2	3	4	5
Original item id	5	1	2	4	3
Profit	5	10	150	35	25
Quantity	60	12	4	5	4
TWU	987	964	810	595	422

Fig. 2. *GlobalItemTable* of database of Fig. 1

3.2 Database Subdivision by Parallel Projection

We adapt the concept of parallel projection reported in [10] for datasets that are too large for the corresponding *CUP-Tree* to fit into main memory. Using the *GlobalItemTable* of Fig. 2, the original database is transformed into the mapped transaction database. Concurrently the parallel projection scheme is constructed. Fig. 3

illustrates the process for the database of Fig. 1 using the corresponding *GlobalItem-Table*. For item index $i \geq 2$, every transaction t with positive quantity of items up to item i is written into the subdivision p_i with the corresponding quantities and TWU. So there are five entries in p_2 from transactions t_6, t_7, t_8, t_9 and t_{10} . Similarly, in p_3 , we have entries for item index 1 and/or 2 that occur before index 3 in the transactions t_2, t_4, t_8 and t_{10} . The third subdivision (p_4) is from transactions t_1, t_3, t_6, t_8 , and t_9 , and the last subdivision from transactions t_1, t_2, t_3 , and t_5 .

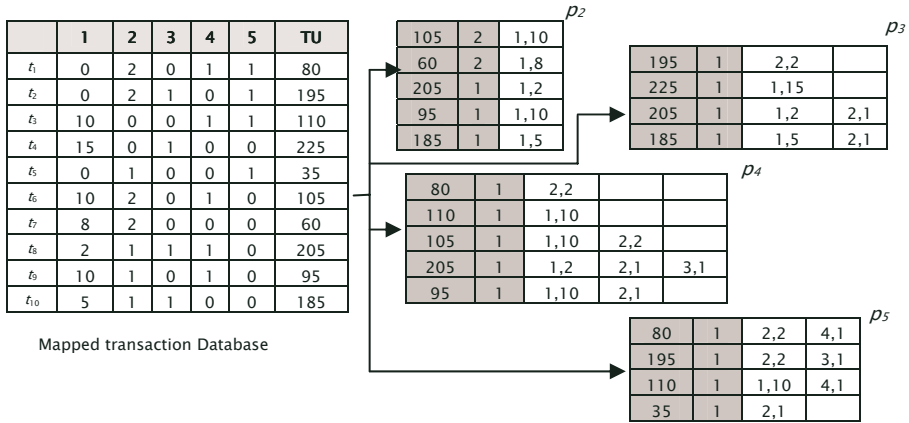


Fig. 3. Parallel projection of transaction database

Once the subdivisions are created, each subdivision p_i is mined separately for high utility patterns.

3.3 Mining Subdivisions Using Compressed Utility Pattern Tree

The Compressed Utility Pattern Tree (*CUP-Tree*) originally proposed by us in [7] is a variant of *CFP-Tree* [11] and *CTU-Tree* [5] data structures. In this paper for a given subdivision p_i of the database, we construct the corresponding *Compressed Utility Pattern Tree (CUP-Tree)* exactly in the same way as in [5]. Fig. 4 illustrates the *CUP-Tree* and the *GlobalItemTable* for subdivision p_5 of the database given in Fig. 1. More explicitly, the first row of subdivision p_5 has mapped item-ids (pattern) 2, 4 and 5 (for the original items 1, 3 and 4) with respective quantities 2, 1, and 1. This pattern is inserted into the tree with the TWU (80) and a pointer to the array of quantities 2, 1 and 1 (node labeled $\textcircled{5}$ in Fig. 4). The nodes which represent the current subdivision p_5 (index 5) are linked by node links to facilitate the traversal in the mining process. All the transactions will be inserted similarly, giving the *CUP-Tree* of Fig. 4.

Now mining for the subdivision p_5 is initiated using the *CUP-Tree* of Fig. 4 as input. Traversing the nodelink of index 5 (Fig. 4.) the associated items are recorded in the projection tree named *ProCUP-Tree*. The information for extracting high utility items is recorded in a *High Utility Pattern Tree (HUP-Tree)* with mapped item id 5 as its root (labeled A in Fig. 6 which shows the *ProCUP-Tree* and *HUP-Tree* of subdivision p_5). The mining of a subdivision consists of three steps: (1) Construction of

ProItemTable, (2) Construction of *ProCUP-Tree*, and (3) Mining by traversing *ProCUP-Tree*. These steps are explained below using subdivision p_5 as an example.

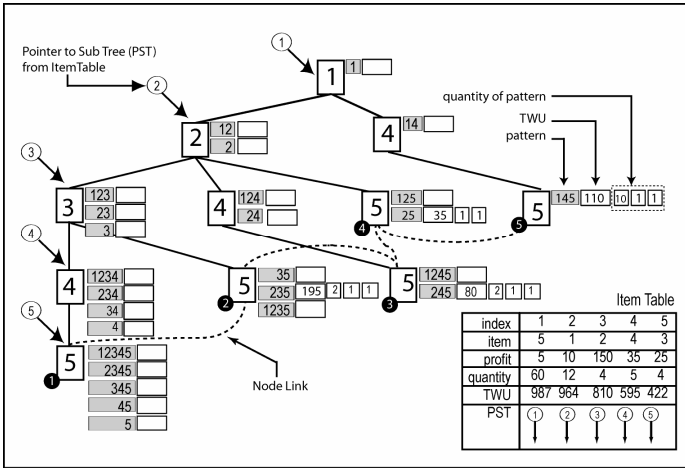


Fig. 4. CUP-Tree and ItemTable for projection p_5

Step 1. Construction of ProItemTable. Traversing the node link of index 5 (Fig. 4), the associated items are arranged as *ProItemTable* in the descending order of the TWU values. This table is constructed in the same manner as the *GlobalItemTable* explained in [5] restricting attention to linked nodes of index 5 in p_5 .

For example, traversing of the linked nodes provides the following indexes and TWU values column 1 (110) from ⑤, 2 (310) from (②, ③, ④), 3 (195) from ②, and 4 (190) from (③, ⑤). Since our *min_utility* is 129.9, index 1 is pruned and indexes 2,3,4 (item ids: 1,2,4) are locally hTWU in *ProItemTable* (see Fig. 6). The mapping of the *GlobalItemTable* item index to the *proItemTable* item index and the corresponding original item-ids are provided in Fig. 5. Concurrently the level 1 children of the *HUP-Tree* root are recorded (indicated by label B in Fig. 6). Furthermore, note that the *proItemTable* includes a column giving the cumulative quantity of the projection item 5.

<i>GlobalItemTable</i> index	1	2	3	4	5
Original item index	5	1	2	4	3
<i>ProItemTable</i> index	-	1	2	3	-

Fig. 5. Mapping projection of index 5 using *ProItemTable*

Step 2. Construction of ProCUP-Tree. Retraversing the *node-link* in the *CUP-Tree* and using the mapping in *ProItemTable* (see Fig. 5), we construct the *ProCUP-Tree*. Note that *ProCUP-Tree* is expressed using the *proItemTable* index.

Step 3. Mining by ProCup-Tree Traversal. For each item in *proItemTable*, the path to the root is traversed computing the other items that are together with the current item. In Fig. 6, traversing the *node-link* of item index 2 will return index 1, and since it

has high TWU, the real utility value will be calculated by multiplying the appropriate quantity with the utility value in *GlobalItemTable*. The quantity at that node is 2 1 1 corresponding to the local pattern of 1 2 5 (original item ids, 1 2 3). So the utility value for the pattern would be $(2 \times 10) + (1 \times 150) + (1 \times 25) = 195$. An entry is created and attached as the child of index 2 (indicated by label C in Fig. 6) in the *HUP-Tree*.

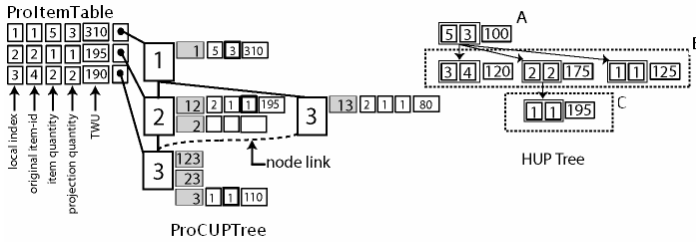


Fig. 6. ProCUPTree and HUP Tree

By traversing the *HUP-Tree* we can print the real utility of itemsets containing item 3 (index 5) as follows: 3 (100), 3 4 (120), 3 2 (175), 3 2 1 (195) 3 1 (125). Since the threshold is 129.9, only itemsets 3 2 and 3 2 1 are reckoned as high utility itemsets. The mining process is continued with the remaining subdivisions p_4, p_3, p_2 . The complete high utility itemsets (after mapping back to the original item-ids) are obtained as the following: {2(600), 4(175), 5(300), 12(490), 14(200), 15(245), 23(175), 24(185), 25(560), 45(300), 123(195), 124(195), 125(355), 145(255), 245(195), 1245(205)}.

4 Performance Study

In this Section, the performance of *CTU-PROL* is empirically compared with the implementation of *TwoPhase* downloaded from [8] and *CTU-Mine* [5]. *CTU-PROL* is written in C++ and compiled using g++ version 4.1.0. The experiments were performed on a Pentium Core Duo, 3 GB RAM, with Linux operating system. We used the real datasets Retail and BMSPOS available from the FIMI Repository [12]. We also generated the synthetic datasets T10N5D100K and T5N5DXM using our program and IBM Quest data generator [13] to test the scalability of our algorithm. Table 1 shows the characteristics of the datasets. Since all these datasets are normally used for traditional frequent itemset mining, we had to add quantity and item utility values to the datasets. We generated utility values from a suitable log-normal distribution, and the quantities randomly from numbers one to ten.

Results of our experiments are shown in Fig. 7. For high thresholds in the Retail dataset, *TwoPhase* runs slightly faster compared to *CTU-PROL*, but when the utility threshold becomes lower, *CTU-PROL* outperforms *TwoPhase*. For very low utility thresholds, the performance of *TwoPhase* got worse. This is due to the limitations of the generation-and-test approach of *TwoPhase* that has to traverse the database many times to enumerate and compute the large number of possible itemsets of high transaction weighted utility. As our algorithm is based on pattern growth using a compact tree, repeated traversal of the database is avoided.

Table 1. Characteristics of Datasets

Dataset	# of Trans	# of different. item	Size of file (MB)
Modified Retail	88,162	1,658	8
Modified BMSPOS	515,597	16,470	30
T10N5D100K	100,000	100	5
T5N5DXM	1,000,000 – 5,000,000	100	35 – 200

Note that for relatively large datasets, *CTU-Mine* ran out of memory and hence *CTU-PROL* is compared with only *TwoPhase*. On the synthetic dataset T10N5D100K, we tested *CTU-Mine*, *CTU-PROL* and *TwoPhase*. When the utility threshold is low, *TwoPhase* is unable to complete within a 10,000 seconds time limit. For scalability, we tested *TwoPhase* and *CTU-PROL* using T10N5DXM datasets with *min_utility* 0.05% of total utility. In general, the results show that *CTU-PROL* outperforms *CTU-Mine* and *TwoPhase* for various utility thresholds and transaction volumes.

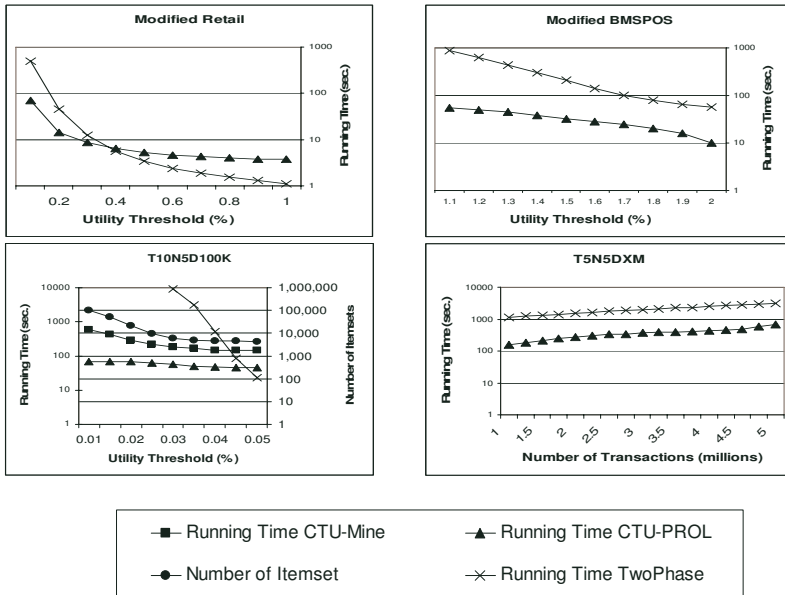


Fig. 7. Execution time with varying minimum utility thresholds and number of transactions on real and synthetic datasets

5 Conclusion

In this paper, we have presented the *CTU-PROL* algorithm to mine the complete set of high utility itemsets from both sparse and relatively dense datasets with short or longer high utility patterns. Our data structure and algorithm extend the pattern growth approach, taking into account the lack of anti-monotone property for pruning utility based patterns. We have compared the performance of *CTU-PROL* against the recent *TwoPhase* algorithm [4] and *CTU-Mine* [5]. The results show that *CTU-PROL*

works more efficiently than *TwoPhase* and *CTU-Mine*. Our algorithm adapts to large data by constructing parallel subdivisions on disk that can be mined independently. The experiments show that *CTU-PROL* is scalable for larger datasets.

Since TWU is an overestimation real utility, resources used are possibly high for these pattern growth algorithms. Further research is needed to determine how the thresholds for TWU may be varied from the user specified utility to reduce this overestimate. As the data for mining is very large in general, we plan to study sampling based approximations to reduce the computation.

Acknowledgement

Alva Erwin is supported by Curtin International Research Tuition Scholarship (CIRTS). We thank Ying Liu for providing the *TwoPhase* program.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Database. In: ACM SIGMOD International Conference on Management of Data (1993)
2. Yao, H., Hamilton, H.J., Buzz, C.J.: A Foundational Approach to Mining Itemset Utilities from Databases. In: 4th SIAM International Conference on Data Mining. Florida USA (2004)
3. Yao, H., Hamilton, H.J.: Mining itemset utilities from transaction databases. *Data & Knowledge Engineering* 59(3), 603–626 (2006)
4. Liu, Y., Liao, W.K., Choudhary, A.: A Fast High Utility Itemsets Mining Algorithm. In: 1st Workshop on Utility-Based Data Mining. Chicago Illinois (2005)
5. Erwin, A., Gopalan, R.P.: N.R. Achuthan.: CTU-Mine: An Efficient High Utility Itemset Mining Algorithm Using the Pattern Growth Approach. In: IEEE CIT 2007. Aizu Wakamatsu, Japan (2007)
6. Han, J., Wang, J., Yin, Y.: Mining frequent patterns without candidate generation. In: ACM SIGMOD International Conference on Management of Data (2000)
7. Erwin, A., Gopalan, R.P., Achuthan, N.R.: A Bottom-Up Projection Based Algorithm for Mining High Utility Itemsets. In: International Workshop on Integrating AI and Data Mining. Gold Coast, Australia (2007)
8. CUCIS. Center for Ultra-scale Computing and Information Security, Northwestern University, <http://cucis.ece.northwestern.edu/projects/DMS/MineBenchDownload.html>
9. Yao, H., Hamilton, H.J., Geng, L.: A Unified Framework for Utility Based Measures for Mining Itemsets. In: ACM SIGKDD 2nd Workshop on Utility-Based Data Mining (2006)
10. Pei, J.: Pattern Growth Methods for Frequent Pattern Mining. Simon Fraser University (2002)
11. Suchayo, Y.G., Gopalan, R.P.: CT-PRO: A Bottom-Up Non Recursive Frequent Itemset Mining Algorithm Using Compressed FP-Tree Data Structure. In: IEEE ICDM Workshop on Frequent Itemset Mining Implementation (FIMI). Brighton UK (2004)
12. FIMI, Frequent Itemset Mining Implementations Repository,
13. <http://fimi.cs.helsinki.fi/>
14. IBM Synthetic Data Generator, <http://www.almaden.ibm.com/software/quest/resources/index.shtml>

Tradeoff Analysis of Different Markov Blanket Local Learning Approaches

Shunkai Fu* and Michel C. Desmarais

Ecole Polytechnique de Montreal,
C.P. 6079, Succ. Centre-ville, Montreal, Quebec, Canada
{shukai.fu, michel.desmarais}@polymtl.ca

Abstract. Discovering the Markov blanket of a given variable can be viewed as a solution for optimal feature subset selection. Since 1996, several algorithms have been proposed to do local search of the Markov blanket, and they are proved to be much more efficient than the traditional approach where the whole Bayesian Network has to be learned first. In this paper, we compare those known published algorithms, including KS, GS, IAMB and its variants, PCMB, and one newly proposed called BFMB. We analyze the theoretical basis and practical values of each algorithm with the aim that it will help applicants to determine which ones to take in their specific scenarios.

Keywords: Markov blanket, feature subset selection, local learning.

1 Introduction

In data mining, a classifier is a function that maps instances described by a set of attributes to a class label of the target variable T of interest. In modern large scale applications, how to identify the minimal, or close to minimal, subset of variables that best predicts T is critical to the success, and this procedure is known as feature subset selection.

A principle solution to the feature selection problem is to determine a subset of features that can render the rest of whole features independent of the variable of interest [1,2,3]. Koller and Sahami (KS) [2] first showed that the Markov blanket (MB) of a given target variable T is the theoretically optimal set of features to predict T 's value, although Markov blanket itself is not a new concept and can be traced back to the work of Pearl[11]. In other words, the Markov blanket of T is the minimal set of variables conditioned on which all other variables are probabilistically independent of the target T , denoted as $MB(T)$. Based on the findings that the full knowledge of $MB(T)$ is enough to determine the probability distribution of T and that the values of all other variables become superfluous, we normally can have a much smaller group of variables in the final classifier, reducing the complexity of learning and resulting with a simpler model, but without scarifying classification performance[2, 3, 4].

* Some of this work was done during the author's time in SPSS Inc.

Since KS's work in 1996, there are several attempts to make the learning procedure more efficient and effective, including GS (Grow-Shrink) [5], IAMB (Iterative Associative Markov Blanket) and its variants [1, 3,4,6], MMPC/MB (Max-Min Parents and Children/Markov Blanket)[3], HITON-PC/MB [7], PCMB(Parent-Child Markov Blanket learning) [1] and the more recent one BFMB (Breadth-First search of Markov Blanket) [8]. To our knowledge, this list contains all the published algorithms. In this article, we will discuss these MB local learning algorithms in terms of theoretical and practical considerations, based on our experience gained from both academic research and industry implementation.

In section 2, a brief introduction to these local learning algorithms is presented. Then, in section 3, we choose some of them, and go a little deeper by comparing their characteristics and pointing out relative merits. We conclude with a short conclusion and what our choice in one project is in SPSS Inc.

2 Brief Review of Related Algorithms

Pearl is the first one to define the concept and study the property of Markov blanket in his early work on Bayesian network [10]. Following this work, Koller and Sahami proved that the Markov blanket of a given variable is the theoretically optimal set of features to predict its value [2]. They also proposed an information entropy-based searching algorithm (generally denoted as KS by their initials) which accepts two parameters: (1) the number of variables to retain, and (2) the maximum number of variables the algorithm is allowed to condition on. Obviously, it is a heuristic and approximate algorithm in its nature, and provides no theoretical guarantees [1,3]. Therefore, although they two pointed out a promising direction, the algorithm itself is not guaranteed to succeed.

The GS algorithm [5] was proposed to induce the Bayesian network (BN) via the discovery of local neighbours. The authors aim to construct a BN by first identifying each node's Markov blankets. Like the constraint-based learning algorithms, e.g. PC[9], GS depends on two basic assumptions, faithfulness(see Definition 1 below) and correct or reliable conditional independence (CI) test. Here, the second assumption is required in practice since only when the number of observations is enough, the result of statistical testing would be trustable. More discussion on this can be found in Section 3.3. Actually, these two assumptions are also the basis of the following algorithms. As its name indicates, GS proceeds in two steps, growing greedily first then shrinking by removing false positives. It is the first algorithm proved correct, but it is not efficient and can't scale to large scale applications. However, the correctness of the algorithm makes it a proven subject for future research.

Definition 1 (Faithfulness). A Bayesian Network G and a joint distribution P are faithful to one another, if and only if every conditional independence relationship encoded by G is also present in P [9].

IAMB [4], was proposed in 2003 for classification problems in microarray research where thousands of attributes are quite common. It is an algorithm based on the same two assumptions of GS, sound in theory and especially simple in implementation. IAMB actually is a variant of GS, consisting of two phases – grow and shrink, but it reorders the set of attributes each time a new attribute enters the blanket in the growing phase based on updated CI testing results, which allows IAMB to perform better than GS since fewer false positives will be added during the first phase [3,6].

In spite of the improvement, IAMB is not data efficient since its CI tests may condition on the whole $MB(T)$ (see more discussion in section 3.3). This point is also noticed by its authors, and several variants of IAMB were proposed, like interIAMB and IAMBnPC [4], for a smaller conditioning set, in its maximum, by interleaving the growing-shrinking phases or using PC for the backward phase, respectively. Empirical study shows that IAMB and its variants outperform GS on average in accuracy [4]. Fast-IAMB [6] is another published known work by the author of GS, but there is only gain in speed as reported, no fundamental difference. Among all the algorithms for local learning of MB, IAMB is the most referred one among the family of MB local search algorithms, which can be explained by the fact that only after its success, more attention was attracted to this topic.

Although several IAMB's variants were proposed to improve IAMB's limit on data efficiency, none of them are known as thorough solution with impressive performance. This situation was finally changed upon the introduction of MMPC/MB and HITON-PC/MB, which work differently from IAMB by including the underlying topology into consideration. As we know, given two nodes of a Bayesian network, e.g. T and someone $X \notin MB(T)$, if they are conditional independent, i.e. d-separated [9], the necessary conditioning set or separating set rarely have to be the whole $MB(T)$. IAMB doesn't recognize this, so the conditioning set may be uselessly big even when the underlying graphical model is just a tree. MMPC/MB and HITON-PC/MB make full use of the topology by dividing the search procedure into finding T 's parents/children first, and then discover the remaining variables belonging to $MB(T)$, i.e. spouses of T . Unfortunately, both algorithms are demonstrated not always correct by [1], but they do suggest a correct approach for following work. PCMB [1] and BFMB [8] are examples of this work.

Following the idea of MMPC/MB and HITON-PC/MB, PCMB was also proposed to conquer the data efficiency problem of IAMB, and, more importantly, it is proved correct theoretically. Like IAMB, PCMB can scale well to thousands of features [1], but it is known as much more time-consuming [8]. Along with PCMB, another derived IAMB algorithm is proposed by the same authors, called KIAMB. It requires no faithfulness assumption, and works in a stochastic manner by allowing users to trade off between greediness and randomness in the search procedure [1].

BFMB is the most recent progress reported on this topic, aiming at even better performance than PCMB. It has a similar framework to MMPC/MB, HITON-PC/MB and PCMB by recognizing firstly all T 's parents and children; then, it repeats the search with each node found as target, looking for their own parents and children which contain the spouses of T . Its name comes from the nature of its search procedure, finding T 's neighbours and then the neighbours' neighbours. In an empirical

study [8], BFMB is demonstrated to outperform PCMB, not only on data efficiency but also on speed. Though it is the latest work in this direction, it is thought as an interesting progress, so we include it in this study as well.

Traditionally, given the faithfulness assumption, a whole Bayesian network (BN) can be learned over T and all other features first, from which we can get the $MB(T)$. This has long been known as tedious procedure that is fought with a number of decades [1,3,4,8], say nothing of making them to scale well in large applications. For instance, the publicly available version of the PC[9] and TPDA[12] algorithms accept datasets with only 100 and 255 variables respectively, indicating their expectations on the scalability. By contrast, these local learning algorithms, like IAMB, PCMB and BFMB, all claim to scale well to thousands of attributes. So, this is a meaningful work worthy of researchers' effort, and it is very promising in real applications.

3 Tradeoff Analysis

In the previous section, we briefly cover the known published algorithms for local learning of Markov blanket in a chronological order, as an introduction to the topic. In this section, a comprehensive and detailed comparison on some of these algorithms will be presented, including their relative advantages. We leave out the algorithms that are not proved correct (KS, MMPC/MB, HITON-PC/MB). IAMB and its variants will be grouped together but still denoted as IAMB in the remaining text because they are not theoretically different and no obvious gain from IAMB's variants, with the exception of KIAMB since it relies on different assumption. Therefore, our discussion will contain IAMB, PCMB, BFMB and KIAMB, and the content will cover their theoretical assumption, time efficiency, data efficiency, and scalability.

3.1 Theoretical Assumption

IAMB, PCMB and BFMB are built on two assumptions: (1) faithfulness and (2) correct CI test. These two assumptions also construct the basis for all CI test based learning algorithms for BN. Different from them, KIAMB only requires correct CI test.

Given the faithfulness assumption, there will be a unique Markov blanket corresponding to given variable T [1, 3, 10]. By removing this assumption, the Markov blanket of T doesn't have to be unique, and KIAMB was proposed to work in such condition where higher complexity is expected due to the introduction of additional uncertainty. It works by choosing the candidate in the growing phase in a random way, not like in IAMB where the best one will always be selected in each iteration. Although its authors say that KIAMB outperforms IAMB and PCMB in their study, we feel skeptical on this result. The algorithm will return different $MB(T)$, but, as we understand, this will only happen when the size of data is not large enough to support correct CI tests with whole $MB(T)$ as conditioning set. Its random choice will indeed allow it to discover a larger number of true positives, because possibly attributes with smaller number of categories can be selected first and this will reduce the number of degree of freedom of the conditioning set and therefore allow the algorithm go further given the same amount of data. In extreme condition, where correct CI test is not a

problem because of enough data, KIAMB will produce the same outcome like IAMB. Actually, KIAMB may work with worse result than IAMB sometimes if, unfortunately, attributes with bigger number of categories may be selected first. Besides, we can't see that KIAMB can outperform PCMB through such a stochastic manner since it is not a fundamentally different solution from IAMB to solve the data efficiency problem.

We reach this conclusion based on our analysis on the algorithm's design. Though it is not completely consistent with that as reported, KIAMB is believed to be an interesting variant to IAMB, and it is expected to have better performance than IAMB averagely.

Conclusion 1: *KIAMB can be viewed as a stochastic variant of IAMB, and it is expected to work with more accurate result, by average, than IAMB when the data is limited. Its requirement of no faithfulness assumption is valuable progress.*

3.2 Time Efficiency

Time efficiency is normally measured in terms of the real length of time consumed, but this measure is known as machine- and implementation-dependent. Therefore, researchers of CI-based learning algorithms often employ the number of CI tests consumed as a measure of time complexity [11]. Among those articles about local learning of Markov blanket, this measure is selected only in [9], plus the number of data passes (“# rounds” in the original paper) required when the authors study the relative time complexity of IAMB, PCMB and BFMB. We believe these two measures are more valuable reference to applicants for reference considering that:

1. They are machine-independent, and are mostly determined by the design of algorithms (though implementation may influence them somewhat);
2. Number of data passes, although rarely be referred, can be much more influential to an algorithm's time complexity than number of CI tests when the data size becomes large;
3. Ideally all necessary statistics can be collected in one data pass, and be cached for future reference, this may cost huge amount of storage space when the number of attributes and the their categories increase;
4. If it exceeds the memory limit, some have to be placed in second-level storage equipment, which will decrease, instead increase as expected, the speed because of (1) the frequent happenings of page missing and switching operation and (2) much slower accessing speed when visit external equipments.

Therefore, in real implementation, we choose to collect the statistics in successive steps with the learning going on, and try to judge and collect whatever we need in each data pass based on the current status. Considering that every additional time of scanning can be very time-consuming, an algorithm requiring fewer data passes should be preferred. Our comparison of IAMB, PCMB and BFMB here will based on these two indexes, number of data passes and CI tests required to finish the learning

of the $MB(T)$. Among them two, the number of data passes should be given more attention based on our discussion above.

IAMB's algorithm is quite simple, not only in theory but in implementation. In its growing phase, it repeatedly adds the most promising candidates into $MB(T)$ in a greedy way. In the shrinking step, it iteratively checks if there is $X \in MB(T)$ that is independent with T conditioned on $MB(T) \setminus \{X\}$, removing it if found. A new data pass is needed to add a new positive in the growing phase since $MB(T)$ changes, so for the CI tests; and only one data pass is necessary for the shrinking step because we can collect all statistics for each possible triple $(T, X, MB(T) \setminus \{X\})$. So, the total number of data passes that IAMB needs scales linearly with the number of true positive in $MB(T)$, and how many false positives are wrongly recognized in the growing phase.

PCMB is known more data efficient than IAMB because the algorithm use the knowledge of underlying topology, given the faithfulness assumption. Readers can refer the original paper for the full algorithm specification, and we will directly use the original function names here for convenience [1]. PCMB firstly depend on $GetPC(T)$ to retrieve all the parents and children of T , i.e. those nodes directly connected to T . It calls $GetPCD$ to find those candidate parents/children of T , of which some possibly are false positive. Within $GetPCD$, the algorithm repeatedly checks the existing of false positives along with the choice of best candidates, which are quite data-pass consuming operation considering possible conditioning sets ranging from 0 to some particular size, and it is not good idea to collect all possible statistics in one data pass. For each candidate $X \in GetPCD(T)$, it has to be double checked if $T \in GetPCD(X)$ too, which prevents false positives from entering $PC(T)$, parents/children of T . So, for a particular $PC(T)$, at least $|PC(T)|+1$ times of $GetPCD()$ are called assuming no false positives are found, and each $GetPCD()$ is known to be time consuming. If we continue this analysis, we find that for a specific $MB(T)$, exactly $|PC(T)|+1$ times of $GetPC$ are needed in total, then we can estimate the time complexity of PCMB to be at least $(|PC(T)|+1)^2$ the time of $GetPCD()$.

BFMB's whole framework is quite similar to MMPC/MB, HITON-PC/MB and PCMB since they follow the same strategy, finding $PC(T)$ first, then iteratively look for PC for each $X \in PC(T)$ since some of them are the remaining part of final $MB(T)$, i.e. spouses of T , who are directly connected to T 's children and dependent with T by conditioning a set including their common children with T [1, 3, 8, 9]. Different from PCMB, BFMB's $GetPC$ procedure, *RecognizePC* as named by the author, works in a smart way by borrowing an idea from known the PC algorithm [9]. It realizes that those directly connected to T are T 's parents/children, so it tries to filter out those nodes which are not directly connected to T by conditional independence test with conditioning set starting from size 0. Those which remain linked to T are $PC(T)$. Since BFMB always conditions on the smallest set first when does CI test to determine if some X is conditionally independent of T , its required statistics can be expected when the conditioning set size is specified, so we can collect all we need in one data pass. Comparing with PCMB, BFMB filters out as many as possible true

negatives given different conditioning set size with one data pass, not several data passes to recognize each single true positive separately as in PCMB. Thus, BFMB is expected to require much fewer number of data passes and CI tests, which is proved by [8] and our work. In summary, BFMB also needs at least $(|PC(T)|+1)^2$ calls to *RecognizePC*, like PCMB, but each *RecognizePC* is known to be much more time efficient than *GetPC*.

Though BFMB is much more time efficient than PCMB, it still loses to IAMB on this point. However, as we will see in next section, BFMB is much more data efficient than IAMB, reaching a higher accuracy rate of learning than IAMB over the same data set [8]. Besides, low data efficiency “helps” IAMB to run “faster” in practice because it has to quit from learning when the conditioning set increases to a certain size and the data available is not enough for reliable CI tests, but PCMB and BFMB can postpone this happening due to their data efficiency, doing more CI tests and going further (see more discussion in section 3.3).

Conclusion 2: *IAMB is the most time efficient, and KIAMB is somewhat slower than IAMB due to its stochastic procedure. Comparing with PCMB, BFMB is more time efficient by requiring much fewer data passes and CI tests. Though IAMB works fast, it doesn't bring as high accuracy as PCMB and IAMB.*

3.3 Data Efficiency

All these Markov blanket learning algorithms require correct CI test. In practice, these algorithms will perform a test if the CI test is reliable and skip it otherwise [1,3,4,8]. Following the approach in [9], this criteria requires that the number of instances in dataset is at least five times the number of degrees of freedom in the test. This means that the number of instances required by IAMB to identify $MB(T)$ is at least exponential in the size of $MB(T)$ since the number of degrees of freedom in a test is exponential to the size of the conditioning set and some tests will be conditioned on at least $MB(T)$. However, depending on the underlying topology, for example if it is a tree, $MB(T)$ can be identified by conditioning on sets much smaller than $MB(T)$. Therefore, IAMB is not data efficient because its data requirements can be high. It is reflected by the fact that IAMB will stop the learning at an early time when the size of $MB(T)$ is still small relative to the true set, given limited data in practice.

IAMB's shortcoming is recognized by its authors, which drove more research work on IAMB's variants, MMPC/MB, HITON-PC/MB, PCMB, and BFMB. [1,3,7,8]. Although MMPC/MB and HITON-PC/MB are the first efforts to propose data efficient algorithm by considering the topology, e.g. a tree, PCMB is the first one that is proven correct. BFMB follows this approach by putting the topology knowledge into consideration during learning, but does even better than PCMB. As we explain in Section 3.2, BFMB always conditions on the smallest set to remove false positives, so it is demonstrated more data efficient than PCMB in [8], where BFMB achieves highest accuracy than PCMB and IAMB by recognizing more correct elements belonging to $MB(T)$, given the same amount of data.

Conclusion 3: *BFMB is the most data efficient; PCMB is much efficient than IAMB on this point, which is consistent with the finding reported in [1]. KIAMB, by average, will beat, or at least tie with, IAMB.*

3.4 Scalability

GS was proposed to learn the BN via the recognition of local neighbours of each variable, aiming at better performance than global learning. IAMB was directly proposed for the microarray research where thousands of attributes are quite common, so did its variants. In [4], we can see the empirical study with 1000 variables involved, among GS, KS, IAMB and its variants. PC is compared to other ones only when the number of variables is 50, which reflects that global learning really can't scale to large problem. Although the cardinality of MB in their empirical studies is quite small, only 6, it still has reference value because filtering out (up to) 994 non-MB elements (or true negatives) within acceptable time itself is a success.

In [1], PCMB is applied to a problem of KDD Cup 2001 in which there are 139351 features. Based on its feature selection result, a Naïve Bayes model is built, and a comparative classification accuracy is reported. IAMB is covered in that comparison as well, producing poorer results than PCMB but in a shorter timing length, which matches the conclusion made in Section 3.2 and 3.3.

BFMB, the latest proposed algorithm, is reported to have highest data efficiency comparing with all previous algorithms of this family; besides, it also runs faster than PCMB [8]. Although no large scale testing is conducted in the study [8], BFMB is expected to have, at least, same scaling ability as PCMB.

Conclusion 4: *IAMB, KIAMB, PCMB and BFMB are all scalable to large scale of applications.*

3.5 Summary

Although IAMB, PCMB, BFMB and KIAMB are all proved correct theoretically, they still demonstrate relative strength or weakness when put them together for a comparison study.

For practical applicants, based on our experience, IAMB is strongly recommended if there are enough data because it is easy to implement and fast in speed. The need for large data samples increases quickly (actually, exponentially) when the number of variables and/or the number of levels per variable increase.

In most cases today, we often face the embarrassing problem of insufficient data, or we are not sure if the data available is enough for reliable analysis. When this happens, PCMB or BFMB appears as the best choice. Among them, BFMB is further suggested because displays greater data and speed efficiency.

In one project done in SPSS Inc (the first author once spent some internship time in SPSS in 2007), we finally chose BFMB as the local learning algorithm of Markov blanket after thorough prototype testing and analysis among IAMB, PCMB and BFMB. IAMB initially appeared the most promising, but it performed quite poorly in most tests due to the limit of data efficiency. Then, we turned to PCMB, and gave it

up finally since it requires too many data passes to collect the statistics data. When the training data is large in size, this is a quite time-consuming procedure. Its design does not leave much room for optimization. Finally, we decided to take BFMB though it is quite new. It works quite well since the testing began couples of weeks as of the writing of this paper. Actually, after our optimization, BFMB requires now even much fewer number of data passes and CI tests than that mentioned in [9], without sacrificing accuracy. For example, with the same settings and testing data (Alarm network) like the experiments done in [8], BFMB needs only about two times the number of data pass needed by IAMB, as shown in Table 1 where we attach the refined performance of BFMB through our test but directly refer the results of IAMB and PCMB from [8].

Table 1. Comparison of time efficiency of IAMB, PCMB and BFMB with data simulated from Alarm network containing 37 features

Instances	Algorithms	# rounds	# CI test
5000	IAMB	211±5	5603±126
5000	PCMB	46702±6875	114295±28401
5000	BFMB	446±15	34073±1996
10000	IAMB	222±4	6044±119
10000	PCMB	46891±3123	108622±13182
10000	BFMB	452±12	37462±1502
20000	IAMB	238±10	6550±236
20000	PCMB	48173±2167	111100±9345
20000	BFMB	460±9	40374±1803

Conclusion 5: (1) Given faithfulness and correct CI tests, BFMB is most preferred to IAMB and PCMB overall because of its highest data efficiency and moderate time efficiency; (2) If abundant data is available, IAMB and KIAMB will be the first choice; (3) KIAMB can be viewed a candidate in the exploratory phase.

4 Conclusion

In this paper, we reviewed those known algorithms to do local learning of Markov blanket, which is known as an optimal solution for feature subset selection. Comparing with traditional global learning algorithms, these local learning algorithms are known as much more efficient and able to scalable to large problems. By filtering out those without sound theory basis, we select IAMB, PCMB, BFMB and KIAMB for comparison, in terms of theory basis, time efficiency, data efficiency, and scalability. Empirical studies from published papers and our own experience tell that BFMB is an ideal choice by summarizing those four factors, which explains why it is selected by us in SPSS Inc. as the feature selection algorithm in one of our projects. Of course, when data is not a problem any more, IAMB or KIAMB will be the first choice.

References

1. Pena, J.M., Nilsson, R., Bjorkegren, J., Tegner, J.: Towards scalable and data efficient learning of Markov boundaries. Elsevier Science (2006)
2. Koller, D., Sahami, M.: Toward optimal feature selection. In: Proc. of International Conference on Machine Learning (ICML), pp. 284–292 (1996)
3. Tsamardinos, I., Aliferis, C.F., Statnikov, A.: Time and sample efficient discovery of Markov blankets and direct causal relations. In: Proc. of SIGKDD (2003)
4. Tsamardinos, I., Aliferis, C.F., Statnikov, A.: Algorithms for large scale Markov blanket discovery. In: Proc. of 16th FLAIRS conference, Florida (2003)
5. Margaritis, D., Thrun, S.: Bayesian network induction via local neighbors. In: Proc. of Neural Information Processing Systems (NIPS) conference (1999)
6. Yaramakala, S., Margaritis, D.: Speculative Markov blanket discovery for optimal feature selection. In: Proc. of ICDM (2005)
7. Aliferis, C.F., Tsamardinos, I., Statnikov, A.: HITON: A novel Markov blanket algorithm for optimal variable selection. In: Proc. of American Medical Informatics Association Annual Symposium, pp. 21–25 (2003)
8. Fu, S., Desmarais, M.C.: Local learning algorithm for Markov blanket discovery: Correct, data efficient, scalable and fast. In: Proc. of Australian National Conference on AI (2007)
9. Spirtes, P., Glymour, C., Scheines, R.: Causation, Prediction, and Search. Lecture Notes in Statistics. Springer, Heidelberg (1993)
10. Pearl, J.: Probabilistic Reasoning in Expert Systems. Morgan Kaufmann, San Mateo (1988)
11. Cheng, J., Bell, D., Liu, W.: Learning belief networks from data: An information theory based approach. In: Proc. of CIMM (1997)
12. Cheng, J., Greiner, R.: Learning Bayesian Networks from Data: An Information-Theory Based Approach. *Artificial Intelligence* 137, 43–90 (2002)

Forecasting Urban Air Pollution Using HMM-Fuzzy Model

M. Maruf Hossain, Md. Rafiul Hassan, and Michael Kirley

Department of Computer Science and Software Engineering,
The University of Melbourne,
Victoria, Australia
{hossain, mrhassan, mkirley}@csse.unimelb.edu.au

Abstract. In this paper, we introduce a Computational Intelligence (CI)-based method to model an hourly air pollution forecasting system that can forecast concentrations of airborne pollutant variables. We have used a hybrid approach of Hidden Markov Model (HMM) with fuzzy logic (HMM-fuzzy) to model hourly air pollution at a location related to its traffic volume and meteorological variable. The forecasting performance of this hybrid model is compared with other common tool based on Artificial Neural Network (ANN) and other fuzzy tool where rules are extracted using subtractive clustering. This research demonstrates that the HMM-fuzzy approach is effectively able to model an hourly air pollution forecasting system.

Keywords: Urban air pollution, forecasting, hidden Markov model (HMM), fuzzy logic.

1 Introduction

Urban air pollution (UAP) [1], [2], [3] has great impact on health and lives in society. It is getting a lot of attention because of it is actually changing the environment as we watch. Over the last decade, it has been studied in terms of measurements, physics, chemistry and modeling. The modeling approach is an important tool to study pollution in an urban air shed, and specifically, to measure the air pollution control policies [4]. In most instances, the deterioration of health can be traced back to air pollution [1]. This makes it imperative that air pollution be stopped, or at least, controlled. An effective forecasting tool that can accurately help us control pollution and keep it within acceptable levels is a dire necessity. In order that air quality does not deteriorate any further, scientific plans for analytical methods and pollution control are needed.

Different techniques exist that can forecast environmental events, though only a few have been applied to forecasting air pollution. Air quality phenomena have been traditionally modelled using physical reality as a start, and, for instance, this information has been coded into differential equations. Methods of Computational Intelligence (CI) [5] are a paradigm shift from the current approach which puts a model together based only on measured data. CI is fairly new to

the environmental scientists and engineers, which is based on the hypothesis that reasoning can be realised using computation [6]. Methods used in computational intelligence include a number of forms of computation, the most famous of which are neurocomputing and fuzzy logic. Neurocomputing works on principles that were discovered studying the brain and its organizational structure [7], whereas fuzzy logic is based on the fuzzy set theory, which extends traditional bivalent logic into continuous group membership with truth values between 0 and 1 [8].

Challenges: Amongst the major challenges in forecasting UAP, the prediction of episodes with high pollutant concentration in urban areas in order that authorities can provide appropriate means to counter potential problems. Authorities and the public demand precise forecasts of urban air quality, especially during episodes where the pollution levels are above the threshold values of acute health effects, and this demand has turned into an outcry during the last few years after the introduction of higher air quality standards (EC/92/72, EC/96/62 and EC/99/33, for example) [2]. Authorities in many European cities have established emergence preparedness systems that handle air pollution episodes and it is quite likely that we shall see more such systems in times to come.

UAP forecast quality depends mainly on three factors: the mapping of emissions, the UAP model and the quality of meteorological forecast data.

Contribution: Our study presented here aims at attaining a better understanding of phenomena associated with air pollution at a location related to its traffic volume and meteorological variable. The objective is to show how Hidden Markov Model (HMM) [9] along with fuzzy logic [10] can be used to create a model that can forecast concentrations of airborne pollutant variables.

Organization: The remainder of the paper is organized as follows. In Section 2, we briefly discuss work related to this area. Our model is formally described in Section 3. Section 4 provides a description of the dataset, design of the experiment carried out in this study and the result of the experiment. Finally in Section 5, we discuss the results and conclude the paper.

2 Related Work

Artificial Neural Networks (ANNs) are a common tool used in most of the similar applications. For instance, neural models for ozone concentrations have been constructed [11] and a model that predicts hourly NO_x and NO_2 concentrations has been successfully applied [12]. Most of the work has focused on comparing feed-forward ANN, especially multi-layer perceptrons (MLP), with traditional methods such as the ARIMA model and linear regression. The results show in general that neural models perform as well as these methods if not better [13]. When applied, however, such ‘black box’ modeling offers too little support for understanding the physical phenomena that are being considered.

Kolehmainen et al. [1] introduced a model using the Self-Organizing Map (SOM) algorithm, Sammon’s mapping and fuzzy distance metrics. The MLPs were used to forecast environmental pollution. Actual levels of individual

pollutants were then computed using a combination of MLP models which were appropriate in that situation. Neagu et al. [3] presented a unified approach that integrated implicit and explicit knowledge in neurosymbolic systems as a combination of neural and neuro-fuzzy modules.

3 HMM-Fuzzy Model

We introduce a novel HMM-based fuzzy rule generation tool: HMM-fuzzy model¹ here. In our model, a HMM is used to sort the data vectors in the multivariate dataset and divide the input space into a number of subspaces to form fuzzy rules.

The model comprises three phases:

- **Phase 1:** The HMM is used to partition the input dataset based on the ordering of the calculated HMM-loglikelihood values.
- **Phase 2:** An iterative top-down(divide and conquer) algorithm is used to generate the minimum number of fuzzy rules to meet the pre-defined mean square error(MSE) for the training dataset.
- **Phase 3:** A gradient descent method is applied to fine tune the obtained model parameters.

It should be noted that, before using the HMM to partition the input data vectors, the HMM is trained using the Baum-Welch algorithm [15] and available training data vectors.

3.1 Sorting Training Dataset

To partition the input dataspace, we first sort the data vectors/patterns using a single HMM based on the similarities among the patterns. The initial HMM was built by using random parameter values. Our approach of sorting the data patterns differs from the usual approach using HMM. Therefore, a detailed explanation is presented in the following subsection.

HMM as a data-pattern sorting tool

In our model, a single HMM has been used to sort the available training data based on the HMM-loglikelihood value.

Lemma 1. Let $\lambda = (A, B, \pi)$ be an HMM with k states and n symbols. Let $X = (x_1, x_2, \dots, x_T)$ be a sequence of T symbols. Then the probability of X given λ is:

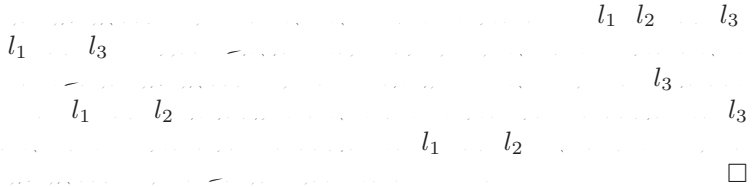
$$\Pr(\mathbf{X}|\lambda) = \sum_{S_1, S_2, \dots, S_k} \pi_{S_1} a_{S_1 S_2} a_{S_2 S_3} \dots a_{S_{T-1} S_T} b_{S_1}(x_1) b_{S_2}(x_2) \dots b_{S_T}(x_k)$$

$$\Pr(X|\lambda) = \sum_{S_1, S_2, \dots, S_k} \pi_{S_1} a_{S_1 S_2} a_{S_2 S_3} \dots a_{S_{T-1} S_T} b_{S_1}(x_1) b_{S_2}(x_2) \dots b_{S_T}(x_k) \quad (1)$$

¹ A prototype of the model was initially developed in our previous study [14].

k

PROOF.



Here, the HMM was used as a pattern matching tool only, where no time dependency is assumed among the data variables (features). Each data vector fed into the HMM are formed using a number of distinct variables. Given the HMM the probability of generating a k -dimensional data pattern, $\langle x_1, x_2, x_3, x_4, \dots, x_k \rangle$, is calculated using the following set of equations [9]:

$$\Pr(X|\lambda) = \sum_Q \Pr(X|Q, \lambda) \Pr(Q|\lambda) \tag{2}$$

- where, Q = State sequence q_1, q_2, \dots, q_k (for a k -state HMM),
- λ = The HMM model,
- X = Input data vector $x_1, x_2, x_3, \dots, x_k$ (Observation Sequence).

The values of $\Pr(X|Q, \lambda)$ and $\Pr(Q|\lambda)$ is calculated using the following equations [9]:

$$\begin{aligned} \Pr(X|Q, \lambda) &= \prod_{i=1}^k \Pr(x_i|q_i, \lambda) \\ &= b_1(x_1)b_2(x_2)...b_k(x_k) \end{aligned} \tag{3}$$

where, $b_i(x_i)$ = Emission probability of the feature x_i from state i .

$$\Pr(Q|\lambda) = \pi_1.a_{1,2}.a_{2,3}...a_k \tag{4}$$

- where, π = Prior probability matrix,
- $a_{i,j}$ = Transition probability from state i to state j .

Bucketing to group similar data vector

The range of log-likelihood values (l_1 to l_m , where l_i = log-likelihood value produced for the i^{th} data vector and m =total data vectors) is split into equal

sized buckets. The data vectors in each bucket produce similar log-likelihood values. Each of the bucket has a start point and an end point corresponding to the log-likelihood values. The size of the bucket, θ , is a parameter of the model that is used to guide the rule extraction process. These buckets were generated so that they can be used to generate fuzzy rules at a later phase.

3.2 Fuzzy Rule Generation

In this phase of the model, we divide the dataset using the buckets and a divide and conquer approach to generate appropriate number of fuzzy rules. To begin with, we create only one fuzzy rule that represents the entire input space of the training dataset. At this point, all log-likelihood values contained in the individual buckets may be perceived as belonging to one global bucket. In the process of rule generation, we calculate the mean μ_{x_i} and standard deviation σ_{x_i} to define the membership function for each features x_i in the dataset as follows:

$$M_{x_i} = e^{-\frac{1}{2}\left(\frac{x_i - \mu_{x_i}}{\sigma_{x_i}}\right)^2} \quad (5)$$

The prediction error for the training data vectors is calculated using the generated fuzzy rule. A mean squared error (MSE) is used to quantify the performance of the developed model for the training dataset. If the prediction error for the training dataset is less than or equals a threshold value ξ the algorithm is terminated and no further rules are extracted. On the other hand, if the prediction error is greater than ξ then the input space is split into two parts with the help of buckets produced in previous section. The splitting of the input space is done by dividing the total buckets into two equal parts. Data in the respective parts constitute the splitted input space. Each splitted partition has individual rules created for it. Finally, the total number of rules is increased by one. The prediction error for the training dataset is recalculated using the extracted rule set. Should the error threshold ξ not be reached then the buckets containing the datasets responsible for the left part of the rule are divided into two rules, and the process is iterated. Again, if the error threshold ξ is still not met, the right part of the rule is partitioned and the process undertaken again. This cycle continues until either the error threshold ξ is met or the number of rules equals the number of buckets.

3.3 Optimization of Extracted Fuzzy Rules

The parameters of the generated fuzzy rules are further fine tuned using a gradient descent algorithm and training dataset. At this stage, the mean and standard deviation for each of the membership functions of all fuzzy rules are fixed more precisely so that it can predict with better accuracy. We follow the gradient descent methodology as in ANFIS [16].

4 Experiment and Result

4.1 Dataset

The dataset used are a subsample of 500 observations from a dataset that was originally put together as part of a study on air pollution related to traffic volume and meteorological variables on a road, conducted by the Norwegian Public Roads Administration. The response variable (column 1) comprised hourly values of the logarithm of the concentration of PM_{10} (particles) measured at Alnabru in Oslo from October 2001 to August 2003. The predictor variables (columns 2 to 8) are the logarithm of the number of cars per hour, wind speed (m/s), temperature 2 meters above the ground ($^{\circ}C$), the temperature difference between 25 and 2 meters above ground ($^{\circ}C$), wind direction (within the range of 0° - 360°), hour of day and day number as counted from October 1, 2001.

4.2 Experiment Design

The experiment was designed using the HMM-fuzzy model. The size of a bucket was chosen to be, $\theta = 0.5$ (while the bucketing was done using log-likelihood values) and the desired MSE was chosen to be 0.001. To optimize the extracted rules, 500 epochs were chosen while executing the gradient descent algorithm.

In this experiment, the number of states in HMM was chosen to be 7 based on a previous study by Hassan et al. [14]. For the HMM, the initial values of transition probability matrix and the prior probability matrix were chosen randomly. As time series datasets are continuous, the observation emission probability matrix of HMM is considered to follow normal distributions where the means and variances are initially chosen randomly. HMM-fuzzy model tool was executed in 10-fold cross validation (CV).

4.3 Results of HMM-Fuzzy Model

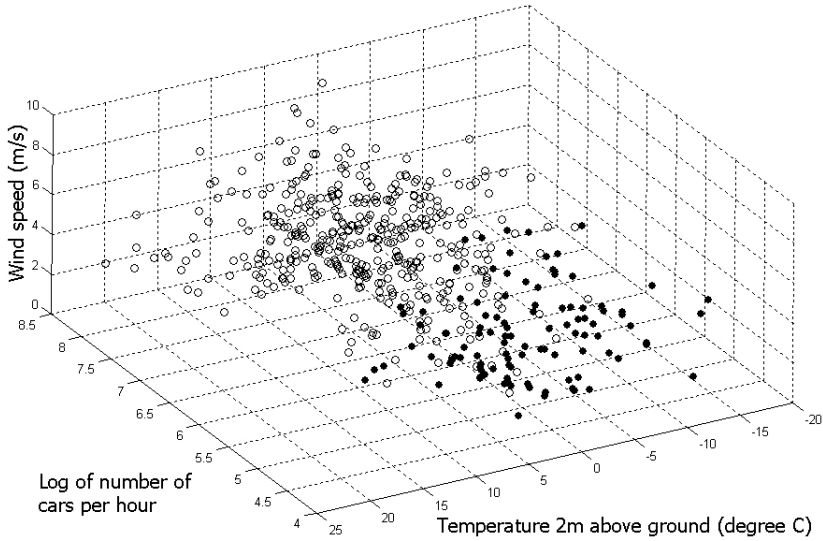
Various numbers of rules were generated in each fold of the execution of HMM-fuzzy model. There were an average of 2.9 ± 1.3703 rules with confidence level of 95% or over.

Figure 1 shows the effect of fuzzy rules in the dataspace. In Fig. 1(a), we see how the dataspace is being divided by the generated rules (figure with respect to the first three attributes only); while in Fig. 1(b), we can see the fuzzy rule that actually divides the dataspace shown in Fig. 1(a). Figure 1(c) shows a membership function of the first attribute shown in Fig. 1(b).

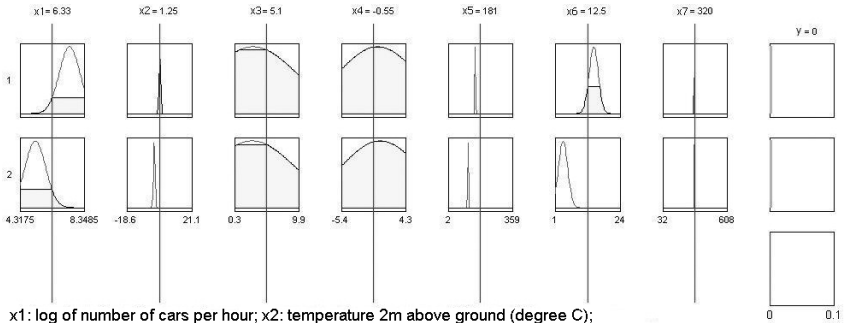
4.4 Results Comparison

We used two other tools in order to compare the results generated by our HMM-fuzzy model. All the tools were executed in 10-fold CV.

As ANNs are commonly used in similar applications, we try to minimize the MSE of ANN as far as possible. We have empirically chosen the architecture of ANN that has the smallest MSE in training data. The ANN had 7 nodes in the

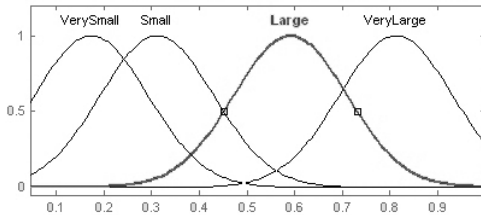


(a) Two groups in the dataspace after using HMM-bucketing (plotted with respect to the first 3 attributes only)



x1: log of number of cars per hour; x2: temperature 2m above ground (degree C);
 x3: wind speed (m/s); x4: temperature difference between 25 and 2m above ground (degree C);
 x5: wind direction (degrees between 0 to 360); x6: hour of day;
 x7: day number from October 1, 2001.

(b) Two fuzzy rules for dividing the dataspace shown in Fig. 1(a)



(c) Membership function of the first attribute shown in Fig. 1(b)

Fig. 1. Effect of fuzzy rules in the dataspace

Table 1. Comparison of Performance of the Tools using 10-fold CV

<i>Technique</i>	<i>MSE</i>	<i>Number of fuzzy rules</i>
HMM-fuzzy Model	0.0097	2.9 ± 1.3703
Fuzzy model following subtractive clustering	0.0102	5.5 ± 1.5811
ANN	0.0216	—

input layer, 21 nodes in the hidden layer and 1 node in the output layer. We used the LM algorithm along with tan-sigmoid activation function. The epochs and training goal were chosen to be 500 and 0.001, respectively. The average training error occurred for the ANN was only 0.002.

We have also generated a forecasting model using the subtractive clustering-based fuzzy model reported in [17]. In this approach, we must provide the radius of the cluster before generating fuzzy rules from the available dataset. Based on the given parameter, i.e., the cluster radius, the model generates a number of clusters in an unsupervised way. We have empirically chosen the cluster radius to be 0.6. Each cluster obtained corresponds in generating a fuzzy rule, i.e., each fuzzy rule relates a region in the input space to an output class.

MSE of HMM-fuzzy model is compared with the MSE and the number of fuzzy rules with confidence level of 95% or over of two other techniques using 10-fold CV is presented in Tab. 1.

5 Discussion and Conclusion

This study has demonstrated that HMM-fuzzy approach technique followed by gradient descent method is effectively able to model an hourly air pollution forecasting system that can predict concentrations of airborne pollutants. This hybrid technique clearly outperforms the other popular tools, such as ANN and fuzzy rule extraction (see Tab. 1). Moreover, the HMM-fuzzy approach generates a significantly fewer number of rules than the technique described in [17].

This system has reduced complexity and simultaneously improved forecasting accuracy. This is most likely because HMM's accuracy in identifying similarities within air pollutant data sequences (i.e. traffic volume and meteorological variables on a road) that consequently provides improved partitions in the input space. Besides, while partitioning the input space using HMM, the similarities among the feature attributes are identified by HMM in terms of fluctuations in magnitude. The end result is an improved set of fuzzy rules that can predict the concentration of PM_{10} particles.

To determine the efficiency of the HMM-fuzzy model proposed in this paper, it is vital to compare it with other well-performed fuzzy rule finding methods, for instance [17]. From the comparison, it is evident that other techniques consider the individual input features to be independent of each other and, this may generate extra rules making the overall system complex. The increased number of rules without taking into consideration the interdependencies among the input

variables does not always lead to a more effective model. Experimental results presented in this paper support this observation.

This paper demonstrates that the hybrid HMM-fuzzy model has the potential to achieve high levels of performance when it designs an hourly air pollution forecasting system that can effectively track and forecast concentrations of airborne pollutant variables. We recommend that higher sample sizes and various meteorological variables should be used in continual research along these lines. The results will forecast air pollution, extrapolate its effects and can help decide on a proper course of action to combat the problem.

References

1. Kolehmainen, M., Martikainen, H., Hiltunen, T., Ruuskanen, J.: Forecasting Air Quality Parameters Using Hybrid Neural Network Modelling. *Environmental Monitoring and Assessment* 65(1–2), 277–286 (2000)
2. Baklanov, A., Rasmussen, A., Fay, B., Berge, E., Finardi, S.: Potential and Shortcomings of Numerical Weather Prediction Models in Providing Meteorological Data for Urban Air Pollution Forecasting. *Water, Air, & Soil Pollution: Focus* 2(5–6), 43–60 (2002)
3. Neagu, C.D., Avouris, N., Kalapanidas, E., Palade, V.: Neural and Neuro-Fuzzy Integration in a Knowledge-Based System for Air Quality Prediction. *Applied Intelligence* 17(2), 141–169 (2002)
4. Zannetti, P.: *Air Pollution Modeling – Theories, Computational Methods and Available Software*. Computational Mechanics Publications, Southampton (1990)
5. Engelbrecht, A.: *Computational Intelligence: An Introduction*. J. Wiley & Sons, Hoboken (2002)
6. Poole, D., Macworth, A., Goebel, R.: *Computational Intelligence, A Logical Approach*. Oxford University Press, New York (1998)
7. Haykin, S.: *Neural Networks: A Comprehensive Foundation*. Prentice-Hall, New Jersey (1994)
8. Zimmermann, H.J.: *Fuzzy Set Theory – And Its Applications*, 2nd revised edn. Kluwer Academic Publishers, Dordrecht (1991)
9. Rabiner, L.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. *Proceedings of the IEEE* 77(2), 257–286 (1989)
10. Kaufmann, A.: *Introduction to the Theory of Fuzzy Sets*. Academic Press, New York (1975)
11. Comrie, A.: Comparing Neural Networks and Regression Models for Ozone Forecasting. *Journal of Air and Waste Management Association* 47, 653–663 (1997)
12. Gardner, M., Dorling, S.: Neural Network Modelling and Prediction of Hourly NO_x and NO_2 Concentrations in Urban Air in London. *Atmospheric Environment* 33(5), 709–719 (1999)
13. Gardner, M., Dorling, S.: Artificial Neural Networks (The Multilayer Perceptron) – A Review of Applications in the Atmospheric Sciences. *Atmospheric Environment* 32(14–15), 2627–2636 (1998)
14. Hassan, M., Nath, B., Kirley, M.: A HMM based Fuzzy Model for Time Series Prediction. In: *Proceedings of IEEE International Conference on Fuzzy Systems (FUZZ-IEEE 2006)*, Vancouver, BC, Canada, pp. 9966–9974 (2006)

15. Männle, M.: Identifying Rule-Based TSK Fuzzy Models. In: Proceedings of the European Congress on Intelligent Techniques and Soft Computing (EUFIT 1999), Aachen, Germany, ELITE Foundation, pp. 286–299 (1999)
16. Jang, J.: ANFIS: Adaptive-Network-Based Fuzzy Inference System. *IEEE Transaction on Systems, Man and Cybernetics* 23(3), 665–685 (1993)
17. Chiu, S.: Extracting Fuzzy Rules from Data for Function Approximation and Pattern Classification. In: Dubois, D., Prade, H., Yager, R. (eds.) *Fuzzy Information Engineering: A guided Tour of Applications*. John Wiley & Sons, Chichester (1997)

Relational Pattern Mining Based on Equivalent Classes of Properties Extracted from Samples*

Nobuhiro Inuzuka, Jun-ichi Motoyama,
Shinpei Urazawa, and Tomofumi Nakano

Nagoya Institute of Technology, Gokiso-cho Showa, Nagoya 466-8555, Japan
inuzuka@nitech.ac.jp, ha8bu3@phaser.elcom.nitech.ac.jp,
shin1008@phaser.elcom.nitech.ac.jp, nakano.tomofumi@nitech.ac.jp

Abstract. This paper extends the bottom-up relational miner MAPIX [9]. It takes a relational database consists of multiple relational tables including a target relation, and enumerates patterns with which a large part of instances in the target relation match. The patterns are given as logical formulae. Although a well-known system WARMR generates and tests possible patterns, it has limitation in its efficiency. MAPIX took a bottom-up approach and gained efficiency at the cost of variety of patterns. It searches and propositionalizes features appeared in instances. Patterns produced is only simple combinations of attributed. The proposed algorithm EQUIVPIX (an equivalent-class-based miner using property items extracted from examples) keeps the merits of bottom-up approach, i.e. time-efficiency and prohibition of duplicated patterns, and it widens pattern variation. EQUIVPIX introduces equivalent classes on properties extracted and also two combination operators of them.

1 Introduction

Relational pattern mining is discussed in the framework of multi-relational mining and it is suitable to use the technique of inductive logic programming (ILP). WARMR [2,3,4] is a representative algorithm of this context.

WARMR generates candidate patterns (queries) in top-down way from simple to complex in level-wise. Then it cuts down unnecessary patterns using a saved infrequent query set. The set has a similar function to the principle used in Apriori [1]. In spite of the cut-down procedure it has limitation, because of the exponentially growing space of hypothesis with respect to the length of patterns and the number of relations. MAPIX acquired much efficiency at the sacrifice of the variety of patterns. It only finds patterns as combination of attributes, which are dynamically constructed as a set of first-order literals from given target instances. It is bottom-up in the sense that attributes are not given in advance but are constructed from given instances. It first constructs first-order features, called property items, appeared in target instances. Then it applies Apriori-like

* Partially supported by Takahashi Industrial and Economic Research Foundation.

procedure for the property items. It succeeded to prohibit duplication of patterns in the sense of logical equivalence.

The bottom-up construction restricts the range of patterns in ones appeared in instances. This paper studies to construct a large variety of patterns by combining attributes. The method becomes truly first-order. The variety approaches to full enumeration in keeping the efficiency.

2 Preparations

Familiarity on logic programming is assumed. We use Datalog, a Prolog without functors, to represent data and patterns. A Datalog clause is a universally quantified formula of the form, $\forall(h \leftarrow b_1 \wedge \dots \wedge b_n)$. \forall is omitted when understood. For $c = h \leftarrow b_1 \wedge \dots \wedge b_n$, $\text{head}(c)$ denotes h and $\text{body}(c)$ denotes $b_1 \wedge \dots \wedge b_n$. When $n = 0$ a clause is called a fact. $\theta = \{v_1/t_1, \dots, v_n/t_n\}$ means a substitution and $P\theta$ for a formula P means replacing every variable v_i in P with a term t_i .

For our mining task a Datalog DB R is given. A predicate corresponds to a relation. A predicate p is extensional when every formula whose head uses p is a ground (no variable) fact in R , otherwise intensional. One of extensional relations is specified as a target (It corresponds to the concept target of WARMR). A fact of the target relation is called a $\text{target instantiation}$.

A query is existentially quantified conjunct form $\exists(b_1 \wedge \dots \wedge b_n)$. A query $\exists Q$ is said to succeed wrt R when $R \models \exists Q$.

Definitions bellow are brought from [3] with slight modification.

Definition 1 (pattern). A pattern P is a Datalog clause P such that $\exists(\text{target instantiation } \theta) \text{ possesses } P$. freq P is the number of $\text{target instantiation } \theta$ such that θ possesses P . $\text{supp}[P] = \text{freq } P / N$.

Definition 2 (frequent pattern). A pattern P is a pattern such that $\text{supp}[P] > \text{min}$. freq P is the number of $\text{target instantiation } \theta$ such that θ possesses P . $\text{supp}[P] = \text{freq } P / N$.

Consider a DB R_{fam} (Fig. 1), including relations, $\text{p}(x, y)$ meaning x is a parent of y , $\text{f}(x)$ for female x , $\text{m}(x)$ for male x , and $\text{gf}(x)$ meaning x is someone's grandfather. Let gf be a target and $\epsilon = \text{gf}(\text{taro})$ is a target instance. The following formula is a pattern.

$$P = \text{gf}(A) \leftarrow \text{m}(A) \wedge \text{p}(A, B) \wedge \text{m}(B)$$

$P(\epsilon)$ denotes a query, $P(\epsilon) = \exists((\text{m}(A) \wedge \text{p}(A, B) \wedge \text{m}(B))\theta) = \exists(\text{m}(\text{taro}) \wedge \text{p}(\text{taro}, B) \wedge \text{m}(B))$, where θ is the target instantiation of ϵ to P . $P(\epsilon)$ succeeds by assigning jiro to B then e possesses P . \square

In order to discuss for equivalence of patterns we use θ -subsumption.

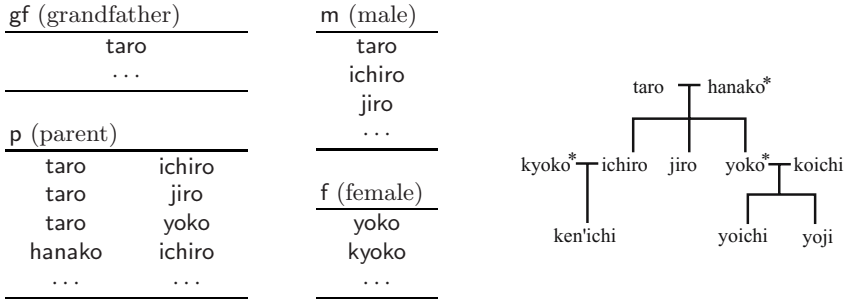


Fig. 1. A family example R_{fam} , including four relations. The gf is target.

Definition 3. $\exists C \quad \exists D \quad C \supseteq D \quad C \text{ subsumes } D$
 $D \text{ subsumption-equivalent } C \preceq D \quad C \preceq D \quad D \preceq C \quad C \sim D$

The \preceq coincides with logical implication when patterns have no recursion. Many ILP algorithms assumes modes for arguments of predicates to restrict patterns. Some arguments have a role as input and others as output. For example modes for the predicates mentioned are given as $p(+, -)$, $m(+)$, and $f(+)$, where $+/-$ means an input/output mode.

We distinct two classes of predicates obeying [6]. Predicates with at least one $\langle - \rangle$ -arg. are called *generating predicates*, e.g. $p(+, -)$, and have a role like a function generating a term from others. Predicates without $\langle - \rangle$ -arg. are called *describing predicates*, e.g. $m(+)$ and $f(+)$, and have a role describing a property of given terms. An instance of a path/check predicate in DB is called a path/check literal. We do not give mode for target.

3 Outline of MAPIX Algorithm

The outline of MAPIX algorithm is as follows:

1. It samples target instances from a target relation.
2. For each sampled instance it collects things (\dots) hold on DB.
3. By generalising the properties it generates first-order attributes (which correspond to items in association rule mining), called \dots .
4. It executes Apriori-like level-wise frequent pattern mining algorithm by regarding the satisfaction of a property item as possession of it.

For an instance $\epsilon = gf(taro)$ we may find a thing hold on it,

$$\{p(taro, ichiro), p(ichiro, ken), m(ken)\}.$$

It may be read that **taro** has a grandson and regard it a \dots of ϵ . By replacing terms by variables and giving a head we have a pattern,

$$gf(A) \leftarrow p(A, B) \wedge p(B, C) \wedge m(C). \tag{1}$$

Table 1. Properties and property items of $\epsilon = \text{gf}(\text{taro})$

$\text{pr1} = \{\text{p}(\text{taro}, \text{ichiro}), \text{m}(\text{ichiro})\}$	$\text{i1} = \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I)$
$\text{pr2} = \{\text{p}(\text{taro}, \text{ichiro}), \text{p}(\text{ichiro}, \text{ken}), \text{m}(\text{ken})\}$	$\text{i2} = \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{p}(I, K) \wedge \text{m}(K)$
$\text{pr3} = \{\text{p}(\text{taro}, \text{jiro}), \text{m}(\text{jiro})\}$	$\text{i3} = \text{gf}(T) \leftarrow \text{p}(T, J) \wedge \text{m}(J)$
$\text{pr4} = \{\text{p}(\text{taro}, \text{yoko}), \text{f}(\text{yoko})\}$	$\text{i4} = \text{gf}(T) \leftarrow \text{p}(T, Y) \wedge \text{f}(Y)$
$\text{pr5} = \{\text{p}(\text{taro}, \text{yoko}), \text{p}(\text{yoko}, \text{yoichi}), \text{m}(\text{yoichi})\}$	$\text{i5} = \text{gf}(T) \leftarrow \text{p}(T, Y) \wedge \text{p}(Y, X) \wedge \text{m}(X)$

The concepts of path and check literals reveal the structure. A path literal leads a term from a term, e.g. $\text{p}(\text{taro}, \text{ichiro})$ leads ichiro from taro . Terms lead from a term in a target instance make a chain, e.g. taro , ichiro , and ken , and it stops by a check literal, e.g. $\text{m}(\text{ken})$.

Path literals have a function referring an object (an attribute) of an instance, and a check literal describes its character (an attribute value). We assume all interesting features have this $\text{p}(\text{a}, \text{b})$ (a $\text{p}(\text{a}, \text{b})$ and a $\text{m}(\text{b})$)

. Similar ideas have appeared in first-order features in LINUS [7] and 1BC [5] and a search used in pathfinding [8].

When a target instance possesses a pattern yielded from a property as in (II), we can regard the pattern as an item of the instance. To find combinations of items that are frequently appeared in instances are a standard task in association rule mining. This was the idea of MAPIX.

4 Difficulties in MAPIX and the Idea to Them

Again consider the example and $\epsilon = \text{gf}(\text{taro})$. ϵ has properties $\text{pr1}, \dots, \text{pr5}$ as shown in Table I and then it has property items $\text{i1}, \dots, \text{i5}$. To see the structure a variable is given for positions occupied by a term.

MAPIX has difficulty that it can not treat patterns that cross more than one item. When MAPIX treats a combination of items it just concerns if an instance has the two items independently. For pr1 and pr2 , which are connected on ichiro , the itemset $\{\text{i1}, \text{i2}\}$ is treated as a pattern,

$$\text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(T, I') \wedge \text{p}(I', K) \wedge \text{m}(K).$$

This is possessed by ϵ but we may expect another straight pattern,

$$\text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K),$$

which is obtained when we unify I and I' according as I and I' are occupied by a term ichiro in the original properties. A strategy is suggested to obtain a pattern according to occurrence of terms.

We face another difficulty when we adopt this strategy and also are interested in producing a pattern only once in the sense of logical equivalence. We see $\text{i1} \sim \text{i3}$ and $\text{i2} \sim \text{i5}$. We can discard i3 because it relates to no other items but when

we throw away i5 we can not obtain the pattern by combining it and i4. The i2 can not be dropped either since it relates to i1. Keeping necessary items all causes large inefficiency, because the number of itemsets grows exponentially on the number of items and also because kept equivalent items produce many equivalent patterns.

Now we describe our idea. Let think of a pattern by combining the equivalent items i2 and i5 just as in an itemset of MAPIX.

$$gf(T) \leftarrow p(T, I) \wedge p(I, K) \wedge m(K) \wedge p(T, Y) \wedge p(Y, X) \wedge m(X). \quad (2)$$

We can see that this is equivalent to i2 and also to i5. It is convenient that a simple union of this pattern and i1,

$$gf(T) \leftarrow p(T, I) \wedge p(I, K) \wedge m(K) \wedge p(T, Y) \wedge p(Y, X) \wedge m(X) \wedge \underline{m(I)}.$$

yields an equivalent pattern produced by combining i1 and i2,

$$gf(T) \leftarrow p(T, I) \wedge p(I, K) \wedge m(K) \wedge m(I).$$

We can also find that combination of the pattern (2) and i4 yields,

$$gf(T) \leftarrow p(T, I) \wedge p(I, K) \wedge m(K) \wedge p(T, Y) \wedge p(Y, X) \wedge m(X) \wedge \underline{f(Y)}.$$

and it is equivalent to the pattern,

$$gf(T) \leftarrow p(T, Y) \wedge p(Y, X) \wedge m(X) \wedge f(Y).$$

Taking conjunction of equivalent items, such as in (2), as a single item, combining with other items produces patterns no more than once.

5 Properties, Property Items and Two Operators of Them

We introduce concepts including ones used in [9] with some extensions.

Definition 4 (property). $c \leftarrow R \xrightarrow{L} R$ property e

$$L \xrightarrow{c} L$$

$$L \xrightarrow{L} L$$

$$L \xrightarrow{e} e$$

Definition 5 (variablization). $\alpha \xrightarrow{\beta} \alpha$ variablization $\alpha \leftarrow \beta$

$$\alpha = \beta\theta \quad \theta = \{v_1/t_1, \dots, v_n/t_n\} \quad \beta \xrightarrow{fi} \alpha$$

$$t_1, \dots, t_n \quad \theta \quad ff$$

Definition 6. $L = \{l_1, \dots, l_m\}$
 $e \text{ var}(e \leftarrow L) = e \leftarrow l_1 \wedge \dots \wedge l_m$

When L is a property of e , $\text{var}(e \leftarrow L)$ is called a L -pattern (and in short) of e . Possessing P by e and a query $P(e)$ are said as in Definition 1.

The set $L = \{\text{p}(\text{taro, ichiro}), \text{p}(\text{ichiro, ken}), \text{m}(\text{ken})\}$ is a property of ϵ on $\text{m}(\text{ken})$. Then $\text{i1} = \text{var}(\epsilon \leftarrow L) = \text{gf}(A) \leftarrow \text{p}(A, B) \wedge \text{p}(B, C) \wedge \text{m}(C)$ is possessed by ϵ , i.e. $\mathbf{R}_{\text{fam}} \models \text{i1}(\epsilon)$. □

Here we give combination operators of properties to produce patterns.

Definition 7. $\mathbb{L} = \{L_1, \dots, L_n\}$
 $e \text{ i-pattern } \mathbb{L} = \{L_1, \dots, L_n\} \text{ independently combined pattern } P = \text{ind}(e \leftarrow \mathbb{L})$
 $(P) = (P_1)\rho \dots (P) = ((P_1) \wedge \dots \wedge (P_n))\rho$
 $(P_1)\rho = \dots = (P_n)\rho$

When $P = \text{ind}(e \leftarrow \mathbb{L})$ and $P_i = \text{var}(e \leftarrow L_i)$ for a set of properties $\mathbb{L} = \{L_1, \dots, L_n\}$ of e , it holds for any target instance $e' \mathbf{R} \models P(e')$ iff for all i ($1 \leq i \leq n$) $\mathbf{R} \models P_i(e')$.

Definition 8. $\mathbb{L} = \{L_1, \dots, L_n\}$
 $e \text{ s-pattern } \mathbb{L} = \{L_1, \dots, L_n\} \text{ structurally combined pattern } P = \text{str}(e \leftarrow \mathbb{L})$
 $P = \text{var}(e \leftarrow (L_1 \cup \dots \cup L_n))$

For $\mathbb{L}_1 = \{\text{pr1}, \text{pr2}\}$ of the example, its i-pattern is,

$$P_1 = \text{ind}(\epsilon \leftarrow \mathbb{L}_1) = \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(T, I') \wedge \text{p}(I', K') \wedge \text{m}(K').$$

The s-patterns of \mathbb{L}_1 and also of $\mathbb{L}_2 = \{\text{pr4}, \text{pr5}\}$ are as follows.

$$P_2 = \text{str}(\epsilon \leftarrow \mathbb{L}_1) = \text{var}(\epsilon \leftarrow \text{pr1} \cup \text{pr2}) = \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K).$$

$$P_3 = \text{str}(\epsilon \leftarrow \mathbb{L}_2) = \text{var}(\epsilon \leftarrow \text{pr4} \cup \text{pr5}) = \text{gf}(T) \leftarrow \text{p}(T, Y) \wedge \text{f}(Y) \wedge \text{p}(Y, X) \wedge \text{m}(X).$$

We realize that P_2 and P_3 can be combined like i-pattern, that is,

$$P_4 = \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K) \wedge \text{p}(T, Y) \wedge \text{f}(Y) \wedge \text{p}(Y, X) \wedge \text{m}(X),$$

which is another different pattern from others. □

We denote the pattern P_4 by $\text{ind}(\{P_2, P_3\})$. Here $\text{ind}()$ is extended for a set of patterns $\mathbb{P} = \{P_1, \dots, P_n\}$, that is, $\text{ind}(\mathbb{P})$ is a pattern P satisfying $\text{head}(P) = \text{head}(P_1)\rho$ and $\text{body}(P) = \text{body}(P_1)\rho \wedge \dots \wedge \text{body}(P_n)\rho$ for the mgu ρ unifying $\text{head}(P_1)\rho = \dots = \text{head}(P_n)\rho$.

Move to treat the equivalence among items. We have $\text{i2} = \text{var}(\epsilon \leftarrow \text{pr2}) \sim \text{i5} = \text{var}(\epsilon \leftarrow \text{pr5})$. In that case we can be aware of

$$P_5 = \mathbf{var}(\epsilon \leftarrow \text{pr2} \cup \text{pr5}) \sim \mathbf{var}(\epsilon \leftarrow \text{pr2}) \sim \mathbf{var}(\epsilon \leftarrow \text{pr5}).$$

The union $\text{pr2} \cup \text{pr5}$ can be taken as a compound property. Not only it is equivalent to pr2 and pr5 but it has the same role to make s-pattern.

$$\begin{aligned} \mathbf{str}(\epsilon \leftarrow \{\text{pr2} \cup \text{pr5}, \text{pr1}\}) &= \mathbf{var}(\epsilon \leftarrow (\text{pr2} \cup \text{pr5} \cup \text{pr1})) \\ &= \mathbf{gf}(T) \leftarrow \mathbf{p}(T, I) \wedge \mathbf{m}(I) \wedge \mathbf{p}(I, K) \wedge \mathbf{m}(K) \wedge \mathbf{p}(T, Y) \wedge \mathbf{p}(Y, X) \wedge \mathbf{m}(X) \\ &\sim \mathbf{str}(\epsilon \leftarrow \{\text{pr2}, \text{pr1}\}) = \mathbf{gf}(T) \leftarrow \mathbf{p}(T, I) \wedge \mathbf{m}(I) \wedge \mathbf{p}(I, K) \wedge \mathbf{m}(K) \end{aligned}$$

Similarly $\mathbf{str}(\epsilon \leftarrow \{\text{pr2} \cup \text{pr5}, \text{pr4}\}) \sim \mathbf{str}(\epsilon \leftarrow \{\text{pr5}, \text{pr4}\})$. We consider a class of equivalent properties as a compound property.

Definition 9. Ψ \approx Ψ f_i e

$$\text{pr}_i \approx \text{pr}_j \text{ iff } \mathbf{var}(e \leftarrow \text{pr}_i) \sim \mathbf{var}(e \leftarrow \text{pr}_j) \quad \text{pr}_i, \text{pr}_j \in \Psi.$$

$E \in \Psi / \approx = \bigcup_{\text{pr} \in E} \text{pr}$
 $\mathbf{str}(e \leftarrow E) = \mathbf{var}(e \leftarrow \bigcup_{\text{pr} \in E} \text{pr})$ compound property item
 caused by E

$i2 \sim i5$ and then $\text{pr2} \approx \text{pr5}$. Hence $E = [\text{pr2}]_{\approx} = [\text{pr5}]_{\approx} = \{\text{pr2}, \text{pr5}\}$ yields a compound property item,

$$\begin{aligned} \bigcup_{\text{pr} \in E} \text{pr} = \text{pr2} \cup \text{pr5} &= \{\mathbf{p}(\text{taro}, \text{ichiro}), \mathbf{p}(\text{ichiro}, \text{ken}), \mathbf{m}(\text{ken}), \\ &\quad \mathbf{p}(\text{taro}, \text{yoko}), \mathbf{p}(\text{yoko}, \text{yoichi}), \mathbf{m}(\text{yoichi})\}, \\ \mathbf{str}(\epsilon \leftarrow E) &= \mathbf{gf}(T) \leftarrow \mathbf{p}(T, I) \wedge \mathbf{p}(I, K) \wedge \mathbf{m}(K) \wedge \mathbf{p}(T, Y) \wedge \mathbf{p}(Y, X) \wedge \mathbf{m}(X). \quad \square \end{aligned}$$

6 The Mining Algorithm

We are ready to describe the algorithm. Unlike MAPIX an equivalent class of properties, i.e. $\mathcal{E} = \Psi / \approx$, is an item. A pattern is made from a subset of \mathcal{E} using $\mathbf{ind}()$ and $\mathbf{str}()$. It has a sense to use $\mathbf{str}()$ for classes only when they share terms not appeared in the target instance. Otherwise their s-pattern is equivalent to their i-pattern. So another components, \mathcal{B} , are used. $\mathbf{bind}_{i,j}$ designates to use $\mathbf{str}()$ for E_i and E_j in \mathcal{E} .

$$\mathcal{B} = \left\{ \mathbf{bind}_{i,j} \mid \left. \begin{array}{l} \exists \text{pr} \in E_i, \exists \text{pr}' \in E_j, \text{pr and pr' share a term not appeared} \\ \text{in the target instance} \end{array} \right\} \right.$$

The example has equivalent classes $\mathcal{E} = \{E_1, E_2, E_3\}$ where $E_1 = \{\text{pr1}, \text{pr3}\}$, $E_2 = \{\text{pr2}, \text{pr5}\}$, $E_3 = \{\text{pr4}\}$ and binders $\mathcal{B} = \{\mathbf{bind}_{1,2}, \mathbf{bind}_{2,3}\}$, because $\text{pr1} \in E_1$ and $\text{pr2} \in E_2$ share *ichiro* and $\text{pr5} \in E_2$ and $\text{pr4} \in E_3$ share *yoko*. (We omit $\mathbf{bind}_{2,1}$ and $\mathbf{bind}_{4,2}$.) \square

A subset $S \subseteq \mathcal{E} \cup \mathcal{B}$ makes sense when for every $\text{bind}_{i,j} \in S$ E_i and E_j are also in S . In this case we call S valid. For a valid $S \subseteq \mathcal{E} \cup \mathcal{B}$ a subset $S' \subseteq S$ is called a bind-maximal if S' is a maximal subset s.t. it includes a binder $\text{bind}_{i,j}$ for every pair of $E_i, E_j \in S'$.

Definition 10. S_1, \dots, S_m
 $S \subseteq \mathcal{E} \cup \mathcal{B}$ S $\text{pat}(e \leftarrow S)$

$$\text{pat}(e \leftarrow S) = \text{ind}(\text{str}(e \leftarrow (S_1 - \mathcal{B})), \dots, \text{str}(e \leftarrow (S_m - \mathcal{B}))).$$

Let us think of $\{E_1, E_2, E_3, \text{bind}_{1,2}\} \subseteq \mathcal{E} \cup \mathcal{B}$. Its bind-maximal subsets are $\{E_1, E_2, \text{bind}_{1,2}\}$ and $\{E_3\}$. Then this set represents

$$\begin{aligned} \text{pat}(e \leftarrow \{E_1, E_2, E_3, \text{bind}_{1,2}\}) &= \text{ind}(\text{str}(e \leftarrow \{E_1, E_2\}), \text{str}(e \leftarrow \{E_3\})) \\ &= \text{ind}(\text{str}(e \leftarrow \{\text{pr1} \cup \text{pr3}, \text{pr2} \cup \text{pr5}\}), \text{str}(e \leftarrow \{\text{pr4}\})) \\ &= \text{gf}(T) \leftarrow \{\text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K) \wedge \text{p}(T, J) \wedge \text{m}(J) \\ &\quad \wedge \text{p}(T, Y) \wedge \text{p}(Y, X) \wedge \text{m}(X)\} \wedge \{\text{p}(T, Y') \wedge \text{f}(Y')\} \\ &\sim \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K) \wedge \text{p}(T, Y') \wedge \text{f}(Y'). \end{aligned}$$

Another subset $\{E_1, E_2, E_3, \text{bind}_{1,2}, \text{bind}_{2,3}\} \subseteq \mathcal{E} \cup \mathcal{B}$ has bind-maximal subsets $\{E_1, E_2, \text{bind}_{1,2}\}$ and $\{E_2, E_3, \text{bind}_{2,3}\}$ and represents a pattern,

$$\begin{aligned} \text{pat}(e \leftarrow \{E_1, E_2, E_3, \text{bind}_{1,2}, \text{bind}_{2,3}\}) \\ \sim \text{gf}(T) \leftarrow \text{p}(T, I) \wedge \text{m}(I) \wedge \text{p}(I, K) \wedge \text{m}(K) \wedge \text{p}(T, Y') \wedge \text{f}(Y') \wedge \text{p}(Y', X') \wedge \text{m}(X'). \end{aligned}$$

All patterns are listed in $(\mathcal{E}, \mathcal{B})$ -form : $\{E_1\}, \{E_2\}, \{E_3\}, \{E_1, E_2\}, \{E_1, E_3\}, \{E_2, E_3\}, \{E_1, E_2, \text{bind}_{1,2}\}, \{E_2, E_3, \text{bind}_{2,3}\}, \{E_1, E_2, E_3\}, \{E_1, E_2, E_3, \text{bind}_{1,2}\}, \{E_1, E_2, E_3, \text{bind}_{2,3}\}, \{E_1, E_2, E_3, \text{bind}_{1,2}, \text{bind}_{2,3}\}$ □

Table 2 shows the algorithm, EQUIVPIX (an equivalent-class-based miner using property items extracted from examples), which enumerates all of subsets of $\mathcal{E} \cup \mathcal{B}$ that represent frequent patterns. The main routine simply obeys Apriori. CANDIDATE has differences from the original. Since EQUIVPIX uses binders it can not have a linear order on items. Therefore it has to check if a new candidate has not nominated yet. It introduces binders at the second level in Line 7. As MAPIX does EQUIVPIX also checks the subsumption in Line 8 to prohibit duplicate patterns.

Properties and binders have to be from a single instance in principle. This is partially solved. If a single instance includes all structure appeared in instances the instance is enough to produce all patterns. The idea is to integrate all instances to a single instance. It is done by making an isomorphism from an instance to another. Then all terms are replaced by terms mapped by the morphism. This overlapping yields a single large instance and it is reduced by the subsumption equivalence. Unfortunately this integration do not keep all of properties before the integration.

Table 2. An outline of the proposing algorithm EQUIVPIX

EQUIVPIX(\mathbf{R} , T , sup_{\min}):

input \mathbf{R} : a DB; T : target relation; sup_{\min} : minimum support threshold;

output Freq : all subsets of $\mathcal{E} \cup \mathcal{B}$ representing frequent patterns;

1. **Select** a target instance $e \in T$; (or build an instance by integrating all in T)
2. $\Psi :=$ the set of properties of e wrt \mathbf{R} ; $\mathcal{E} := \Psi / \approx$; $\mathcal{B} :=$ the set of binders for \mathcal{E} ;
3. **Remove** every $E \in \mathcal{E}$ s.t. $\text{str}(e \leftarrow E)$ is not frequent from \mathcal{E} ;
4. $k := 1$; $\mathcal{F}_1 := \{\{E\} \mid E \in \mathcal{E}\}$; $\text{Freq} := \mathcal{F}_1$;
5. **while** $\mathcal{F}_k \neq \emptyset$ **do**
6. $\mathcal{C}_{k+1} := \text{CANDIDATE}(\mathcal{F}_k)$; $\mathcal{F}_{k+1} := \{S \in \mathcal{C}_{k+1} \mid \text{sup}[\text{pat}(e \leftarrow S)] \geq \text{sup}_{\min}\}$;
7. $\text{Freq} := \text{Freq} \cup \mathcal{F}_{k+1}$; $k := k + 1$;
8. **Return** Freq ;

CANDIDATE(\mathcal{F}_k):

input \mathcal{F}_k : set of frequent itemsets of a level;

output \mathcal{C}_{k+1} : the set of candidate itemsets of the next level;

1. $\mathcal{C}_{k+1} := \emptyset$
 2. **For each** $S_1, S_2 \in \mathcal{F}_k$ s.t. $|S_1 \cap \mathcal{E} - S_2 \cap \mathcal{E}| = |S_2 \cap \mathcal{E} - S_1 \cap \mathcal{E}| = 1$ **do**
 3. **if** $S_1 \cup S_2 \notin \mathcal{C}_{k+1}$ **then** $\mathcal{C}_{k+1} := \mathcal{C}_{k+1} \cup \{S_1 \cup S_2\}$;
 4. **For each** $S \in \mathcal{C}_{k+1}$ **do**
 5. **For each** $E_i \in S$ **do**
 6. **if** $S - (\{E_i\} \cup \{\text{bind}_{i,j} \in \mathcal{B} \mid j \text{ is an index for any } E_j \in \mathcal{E}\}) \notin \mathcal{F}_k$
 7. **then delete** S from \mathcal{C}_{k+1} ;
 8. **if** $k = 1$ **then for each** $\{E_i, E_j\} \in \mathcal{C}_2$ **do**
 9. **if** $\text{bind}_{i,j} \in \mathcal{B}$ **then** $\mathcal{C}_2 := \mathcal{C}_2 \cup \{E_i, E_j, \text{bind}_{i,j}\}$;
 10. **if** $(\text{str}(e \leftarrow E_i) \preceq \text{str}(e \leftarrow E_j) \text{ or } \text{str}(e \leftarrow E_j) \preceq \text{str}(e \leftarrow E_i))$ **delete** S from \mathcal{C}_2 ;
 11. **Return** \mathcal{C}_{k+1} ;
-

Table 3. Results of experiment with dataset Bongard with $\text{sup}_{\min} = 5\%$

algorithms	runtime	(A)	(B)	(A) = the number of produced patterns. (B) = the number of produced patterns which is not equivalent with other patterns.
EQUIVPIX	237.7	625	625	
MAPIX	142.6	160	160	
WARMR	1098.5	5480	782	

7 An Experiment and Concluding Remarks

EQUIVPIX is implemented using SWI-Prolog on PC of Xeon 2.8GHz. We conveyed a simple experiment to compare with MAPIX and WARMR using a dataset Bongard. Its language bias is modified to test algorithms appropriately. As in Table 3 EQUIVPIX produced 625 patterns while WARMR, a complete enumerator, produced 782 patterns, a part of total 5480 outputs including duplication. Runtime was less than double of MAPIX.

EQUIVPIX includes ideas: (1) two combination operators for properties extracted from samples; (2) equivalence of properties to make compound properties, which keeps efficiency and prohibits duplicated patterns; and (3) the

heuristics to integrate instances. The number of patterns output by EQUIVPIX approached to WARMR compared to MAPIX.

References

1. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: VLDB, pp. 487–499 (1994)
2. Dehaspe, L., De Raedt, L.: Mining association rules with multiple relations. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 125–132. Springer, Heidelberg (1997)
3. Dehaspe, L.: Frequent pattern discovery in first-order logic, PhD thesis, Dept. Computer Science, Katholieke Universiteit Leuven (1998)
4. Dehaspe, L., Toivonen, H.: Discovery of Relational Association Rules. In: Džeroski, S., Lavrač, N. (eds.) Relational Data Mining, pp. 189–212. Springer, Heidelberg (2001)
5. Flach, P.A., Lachiche, N.: 1BC: A first-order Bayesian classifier. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 92–103. Springer, Heidelberg (1999)
6. Furusawa, M., Inuzuka, N., Seki, H., Itoh, H.: Induction of Logic Programs with More Than One Recursive Clause by Analysing Saturations. In: Džeroski, S., Lavrač, N. (eds.) ILP 1997. LNCS, vol. 1297, pp. 165–172. Springer, Heidelberg (1997)
7. Lavrač, N., Flach, P.A.: An extended transformation approach to inductive logic programming. *ACM Trans. Computational Logic* 2(4), 458–494 (2001)
8. Richards, B.L., Mooney, R.J.: Learning Relations by Pathfinding. In: AAAI 1992, pp. 50–52 (1992)
9. Motoyama, J., Urazawa, S., Nakano, T., Inuzuka, N.: A mining algorithm using property items extracted from sampled examples. In: Muggleton, S., Otero, R., Tamaddoni-Nezhad, A. (eds.) ILP 2006. LNCS (LNAI), vol. 4455, pp. 335–350. Springer, Heidelberg (2007)

Evaluating Standard Techniques for Implicit Diversity

Ulf Johansson^{1,*}, Tuve Löfström^{1,2}, and Lars Niklasson²

¹ University of Borås, Department of Business and Informatics, Borås, Sweden
{ulf.johansson, tuve.lofstrom}@hb.se

² University of Skövde, Department of Humanities and Informatics, Skövde, Sweden
lars.niklasson@his.se

Abstract. When performing predictive modeling, ensembles are often utilized in order to boost accuracy. The problem of how to maximize ensemble accuracy is, however, far from solved. In particular, the relationship between ensemble diversity and accuracy is, especially for classification, not completely understood. More specifically, the fact that ensemble diversity and base classifier accuracy are highly correlated, makes it necessary to balance these properties instead of just maximizing diversity. In this study, three standard techniques to obtain implicit diversity in neural network ensembles are evaluated using 14 UCI data sets. The experiments show that standard resampling; i.e. dividing the training data by instances, produces more diverse models, but at the expense of base classifier accuracy, thus resulting in less accurate ensembles. Building ensembles using neural networks with heterogeneous architectures improves test set accuracies, but without actually increasing the diversity. The results regarding resampling using features are inconclusive, the ensembles become more diverse, but the level of test set accuracies is unchanged. For the setups evaluated, ensemble training accuracy and base classifier training accuracy are positively correlated with ensemble test accuracy, but the opposite holds for diversity; i.e. ensembles with low diversity are generally more accurate.

1 Introduction

The main objective of all predictive modeling is to create a model likely to have high accuracy on unseen data. A technique commonly used to maximize generalization accuracy is to utilize ensembles of models, i.e. somehow combining a number of individual models. The main reason for the success of ensemble approaches is the fact that combining several models will eliminate uncorrelated base classifier errors; see e.g. [1]. This, however, requires that the base classifiers commit their errors on different instances; i.e. *ensemble diversity* has an effect on ensemble accuracy. The problem of how to maximize ensemble accuracy is unfortunately far from solved, though. In particular, the relationship between ensemble diversity and accuracy is not completely understood, making it hard to efficiently utilize diversity for ensemble creation.

* Ulf Johansson and Tuve Löfström are equal contributors to this work.

2 Background and Related Work

Krogh and Vedelsby in [2] derived the result that ensemble error depends not only on the average accuracy of the base models, but also on their diversity (ambiguity). More formally, the ensemble error, E , is:

$$E = \bar{E} - \bar{A} \quad (1)$$

where \bar{E} is the average error of the base models and \bar{A} is the ensemble diversity, measured as the weighted average of the squared differences in the predictions of the base models and the ensemble. Since diversity is always positive, this decomposition proves that the ensemble will always have higher accuracy than the average accuracy obtained by the individual classifiers. The two terms are, however, normally highly correlated, making it necessary to balance them instead of just maximizing the diversity term.

Brown et al. in [3] introduced a taxonomy of methods for creating diversity. The main distinction made is between *explicit* methods, where some metric of diversity is directly optimized, and *implicit* methods, where the method is supposed to produce diversity without actually targeting it. All standard resampling techniques are by nature implicit since they randomly sample the training instances for each base classifier.

For artificial neural network (ANN) ensembles, the most obvious method to introduce implicit diversity is to randomize the starting weights. Starting from randomized weights is, of course, a standard procedure for most ANN training. Many methods strive for diversity by splitting the training data in order to train each base classifier using a slightly different training set. Such resampling techniques can divide the available data either by *features* or by *instances*. For ANN ensembles, it is also possible to use ANNs with different architectures in the ensemble. If the base classifiers are standard, fully-connected, feed-forward ANNs, the number of hidden layers and the number of units in each layer can be varied.

According to Brown et al., random initialization of weights is generally ineffective for producing diverse ANNs. The reason is that ANNs often converge to the same, or very similar optima, in spite of starting in different parts of the space; see e.g. [4]. Brown et al. also state that manipulation of ANN architectures most often turns out to be quite unsuccessful. Regarding resampling, finally, Brown et al. say that the view is that it is more effective to divide training data by feature than by instance. In addition, Duin and Tax in [5] found that using one type of classifier on different feature sets was far more effective than using different classifiers on one feature set. Still, it should be noted that Duin and Tax conclude that best performance is achieved by combining both different feature sets and different classifiers.

In [6], Kuncheva and Whitaker studied ten statistics measuring diversity among binary classifier outputs; i.e. correct or incorrect vote for the class label. In the experimentation, all diversity measures evaluated showed low or very low correlation with test set accuracy. In our previous study, we followed up the Kuncheva and Whitaker study, empirically evaluating the ten diversity measures using ANN ensembles and 11 publicly available data sets; see [7]. The main result was again that all diversity measures showed very low correlation with test set accuracy. With these results in mind, the situation is that although theory suggests that diversity is beneficial for ensemble accuracy, we currently do not know how to efficiently utilize diversity or even what measure to use.

The purpose of this paper is to empirically evaluate some standard techniques for introducing implicit diversity in ANN ensembles. More specifically, the study compares

resampling techniques and the use of different architectures for base classifier ANNs against a baseline setup. The baseline setup combines a number of ANNs with identical architectures that were trained individually, using all available training data. The most important criterion is of course generalization accuracy; i.e. accuracy on a test set, but it is also interesting to compare the levels of diversity produced by the different methods. In addition, the study examines how diversity and generalization accuracy co-vary, depending on the technique used to introduce the diversity.

3 Method

In the main experiment, three standard techniques for introducing implicit diversity are evaluated using ANN ensembles and 14 data sets from the UCI repository [8]. All possible combinations of the three standard techniques are evaluated, resulting in 12 different setups. For each setup, 16 ANNs are trained and then all possible ensembles consisting of exactly 11 ANNs from the pool are formed, resulting in 4368 different ensembles. For actual evaluation, standard 10-fold cross-validation is used; i.e. results reported for a specific setup and data set are mean test set accuracies obtained by the 4368 ensembles over the ten folds of the data set. The standard techniques used in the different setups are:

- **Bootstrapping:** Here each ANN is trained using individual training sets. Every training set (called bootstrap) has the same size as the original training set, and is created by repeated sampling (according to a uniform distribution) of training instances. Since sampling is done with replacement, some instances may appear several times, while others are omitted. On average, a bootstrap sample contains approximately 63% of the original training instances.
- **Resampling using features:** Each ANN is again trained using individual training sets. Here, however, each training set uses all available training instances, but a certain proportion of the features are removed. Which features to remove is randomly decided when creating the training set for individual ANNs; i.e. each ANNs is trained using a randomized feature set. In the experimentation, two different levels of feature reduction are evaluated; keeping 80% or 90% of available features.
- **Varied architectures:** In this study, only fully-connected feed-forward ANNs are used. When using varied architectures, eight ANNs have one hidden layer and the remaining eight have two hidden layers. The exact number of hidden units is based on data set characteristics, but is slightly randomized for each ANN. For this randomization, we used the same formulas as in the previous study, see [8]. When the architecture is not varied, each ANN has an identical architecture with one hidden layer. The initial weights are, of course, still randomized, though.

Table 1 below summarizes the 12 setups evaluated. The baseline setup is S1.

Table 1. Setups evaluated

Setup	Bootstrap	Features	Varied architectures	Setup	Bootstrap	Features	Varied architectures
S1	No	100%	No	S7	Yes	90%	No
S2	No	100%	Yes	S8	Yes	90%	Yes
S3	Yes	100%	No	S9	No	80%	No
S4	Yes	100%	Yes	S10	No	80%	Yes
S5	No	90%	No	S11	Yes	80%	No
S6	No	90%	Yes	S12	Yes	80%	Yes

The data sets are summarized in Table 2 below. *Ins.* is the total number of instances in the data set. *Con.* is the number of continuous input variables and *Cat.* is the number of categorical input variables.

Table 2. Data set characteristics

Data set	Ins.	Con.	Cat.	Data set	Ins.	Con.	Cat.
Bupa liver disorder	345	6	0	Iono	351	34	0
Cleveland heart	303	6	7	Labor	57	8	8
Crx	690	6	9	Sick	2800	7	22
Pima Indian diabetes	768	8	0	Thyroid	3163	7	18
German credit	1000	7	13	Tictactoe	958	0	9
Hepatitis	155	6	13	Wisconsin breast cancer	699	10	0
Horse colic	368	7	15	Votes	435	0	16

4 Results

Table 3 below shows the test set accuracies obtained by the different setups. Using a standard Friedman test ($\alpha=0.05$), the only statistically significant difference is that S10 (i.e. 80% features, varied architectures, and no bootstrap) performs better than S11 (80% features, homogenous architecture and bootstrap) and S12 (80% features, varied architecture and bootstrap). The most interesting observations are, however, that the varied architecture is almost always beneficial, while bootstrapping generally decreases the accuracy. Specifically, the three best setups all use varied architectures and no bootstrapping but different proportions of features. The overall results regarding resampling using features are inconclusive, so it is not obvious how the proportion of features used affects the performance.

Table 3. Accuracy

Bootstrap	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Features	100%	100%	100%	100%	90%	90%	90%	90%	80%	80%	80%	80%
Var. Architect.	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Bupa	.702	.716	.698	.716	.706	.709	.713	.699	.724	.727	.704	.718
Cleveland	.823	.831	.796	.825	.818	.825	.820	.827	.822	.829	.813	.823
Crx	.859	.860	.864	.868	.861	.862	.860	.851	.871	.860	.860	.860
Diabetes	.764	.773	.776	.774	.773	.770	.770	.772	.768	.774	.768	.769
German	.764	.765	.762	.756	.761	.765	.767	.760	.761	.766	.754	.760
Hepatitis	.828	.837	.839	.828	.808	.838	.816	.822	.826	.853	.842	.815
Horse colic	.834	.835	.817	.844	.813	.838	.833	.819	.810	.833	.821	.836
Iono	.922	.928	.919	.926	.941	.939	.931	.930	.935	.936	.935	.925
Labor	.893	.900	.912	.920	.924	.911	.912	.903	.905	.938	.907	.919
Sick	.967	.968	.966	.964	.966	.967	.966	.966	.966	.967	.966	.966
Thyroid	.982	.983	.983	.982	.982	.983	.981	.984	.982	.983	.983	.981
Tictactoe	.886	.871	.855	.826	.875	.866	.853	.843	.860	.834	.847	.823
WBC	.965	.964	.959	.962	.962	.963	.960	.968	.965	.966	.962	.963
Votes	.959	.962	.954	.960	.953	.952	.961	.960	.958	.961	.954	.954
Mean	.8676	.8710	.8643	.8680	.8674	.8705	.8674	.8645	.8680	.8733	.8653	.8653
Mean Rank	7.21	4.43	7.36	6.36	6.86	5.21	7.14	7.36	6.50	3.21	8.43	7.93

Table 4 below shows the mean test set accuracies for the base classifiers in each setup. Here it is very obvious that the use of bootstrapping lowers the accuracy of the base classifiers significantly. On the other hand, the use of varied architectures is always favorable. Finally, resampling using features, as expected, also lowers base classifier accuracy.

Table 4. Mean base classifier accuracy

Bootstrap	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Features	100%	100%	100%	100%	90%	90%	90%	90%	80%	80%	80%	80%
Var. Architect.	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Bupa	.687	.689	.674	.677	.664	.663	.652	.637	.662	.675	.646	.647
Cleveland	.784	.795	.751	.776	.778	.775	.754	.757	.775	.779	.744	.761
Crx	.850	.849	.832	.829	.833	.830	.825	.814	.840	.829	.818	.815
Diabetes	.760	.760	.755	.744	.750	.751	.738	.738	.756	.752	.737	.739
German	.726	.722	.704	.704	.719	.718	.700	.699	.717	.721	.698	.702
Hepatitis	.785	.795	.787	.799	.771	.802	.773	.780	.780	.810	.783	.787
Horse colic	.770	.780	.742	.773	.750	.774	.734	.744	.751	.766	.739	.747
Iono	.891	.895	.884	.881	.903	.900	.882	.886	.898	.894	.881	.887
Labor	.871	.888	.835	.850	.870	.865	.854	.843	.868	.883	.840	.851
Sick	.965	.964	.961	.960	.960	.962	.957	.959	.961	.962	.959	.958
Thyroid	.981	.981	.980	.979	.980	.980	.979	.979	.981	.981	.979	.978
Tictactoe	.813	.794	.761	.743	.789	.775	.750	.736	.778	.763	.745	.730
WBC	.954	.955	.946	.950	.954	.953	.947	.951	.952	.956	.946	.952
Votes	.941	.946	.931	.942	.928	.928	.930	.933	.929	.936	.918	.928
Mean	.8413	.8437	.8245	.8290	.8320	.8339	.8196	.8183	.8319	.8361	.8167	.8202
Mean Rank	2.71	1.93	7.57	6.79	5.57	5.14	9.71	9.64	5.57	3.79	10.57	9.00

Table 5 below shows the mean ensemble diversity for each setup. In this study, diversity is measured using the *disagreement* measure, which is the ratio between the number of instances on which one classifier is correct and the other incorrect, to the total number of instances. The disagreement between two base classifier is calculated using (2). If the output of each classifier D_i is represented as an N -dimensional binary vector y_i , where $y_{j,i}=1$ if D_i recognizes correctly instance z_j and 0 otherwise, the notation N^{ab} means the number of instances for which $y_{j,i}=a$ and $y_{j,k}=b$.

$$Dis_{i,k} = \frac{N^{01} + N^{10}}{N^{11} + N^{10} + N^{01} + N^{00}} \tag{2}$$

For this pairwise diversity measure, the averaged value over the diversity matrix, which is the one used in this study, is calculated using (3).

$$D_{av} = \frac{2}{L(L-1)} \sum_{i=1}^{L-1} \sum_{k=i+1}^L Dis_{i,k} \tag{3}$$

As seen in Table 5, using either a subset of features or bootstrapping clearly raises diversity. Varied architectures, on the other hand, does not seem to significantly affect the diversity.

Table 5. Disagreement

Bootstrap	No	No	Yes	Yes	No	No	Yes	Yes	No	No	Yes	Yes
Features	100%	100%	100%	100%	90%	90%	90%	90%	80%	80%	80%	80%
Var. Architect.	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes	No	Yes
Bupa	.146	.168	.241	.240	.252	.271	.320	.325	.243	.255	.321	.327
Cleveland	.194	.165	.248	.218	.210	.216	.255	.255	.215	.219	.267	.244
Crx	.079	.088	.130	.137	.126	.134	.147	.156	.114	.137	.161	.167
Diabetes	.101	.112	.170	.176	.143	.151	.196	.204	.135	.157	.200	.197
German	.221	.226	.273	.273	.234	.238	.283	.283	.238	.239	.287	.281
Hepatitis	.194	.180	.202	.183	.218	.180	.221	.204	.207	.171	.218	.186
Horse colic	.218	.208	.268	.237	.254	.234	.292	.267	.255	.249	.288	.278
Iono	.101	.103	.123	.129	.100	.111	.130	.127	.105	.113	.132	.123
Labor	.105	.090	.182	.179	.129	.134	.171	.172	.130	.119	.195	.173
Sick	.017	.023	.026	.028	.024	.026	.031	.033	.023	.026	.030	.033
Thyroid	.006	.012	.011	.015	.010	.014	.015	.017	.009	.013	.013	.017
Tictactoe	.218	.236	.284	.290	.253	.271	.302	.312	.262	.266	.305	.308
WBC	.039	.036	.051	.046	.041	.040	.053	.051	.042	.039	.059	.043
Votes	.054	.046	.074	.054	.082	.075	.084	.073	.083	.071	.098	.077
Mean	.1211	.1210	.1630	.1574	.1483	.1497	.1786	.1770	.1471	.1480	.1839	.1752
Mean Rank	11.07	10.93	5.93	6.14	8.29	7.57	3.00	3.29	8.14	7.86	2.21	3.57

Ultimately we would like to have a method for selecting or searching for a specific ensemble based on training or validation performance. With this in mind, we decided to look into how ensemble test accuracy varies with ensemble training accuracy, base classifier mean training accuracy and ensemble training diversity. More specifically, the 4368 ensembles were first sorted based on these three measures and then divided into ten groups. Finally, the average test set accuracy for each group was calculated.

When considering single data sets, the picture is sometimes quite clear; see the left plot in Figure 1 below. Here, both high ensemble training accuracy and high base classifier accuracy are beneficial for test set accuracy. It should be noted, however, that high diversity is, on the other hand, detrimental. For other data sets, the picture is, however, completely different; see the right plot in Figure 1. Here, it is actually favorable to pick ensembles with lower training accuracy, or consisting of less accurate base classifiers.

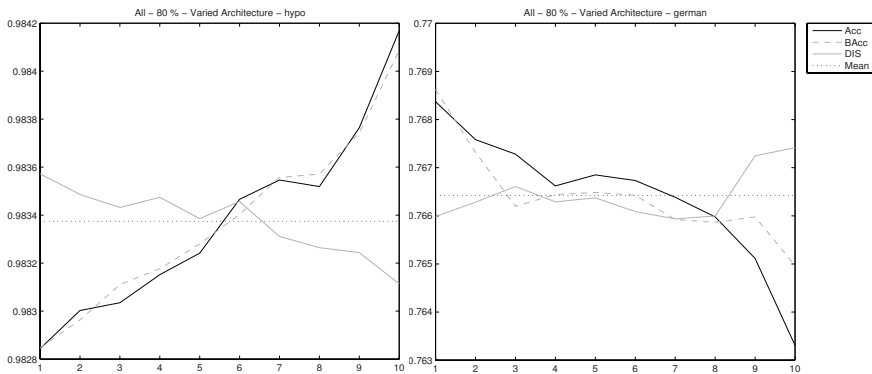


Fig. 1. Test set accuracy vs. ensemble training accuracy, base classifier mean training accuracy and ensemble training diversity. Thyroid and German data sets, 80% features, varied architecture and no bootstrap.

It is of course slightly awkward to average accuracies over several data sets without adjusting for the different accuracy levels, but Figure 2 below shows the overall pictures for setups S10 and S2.

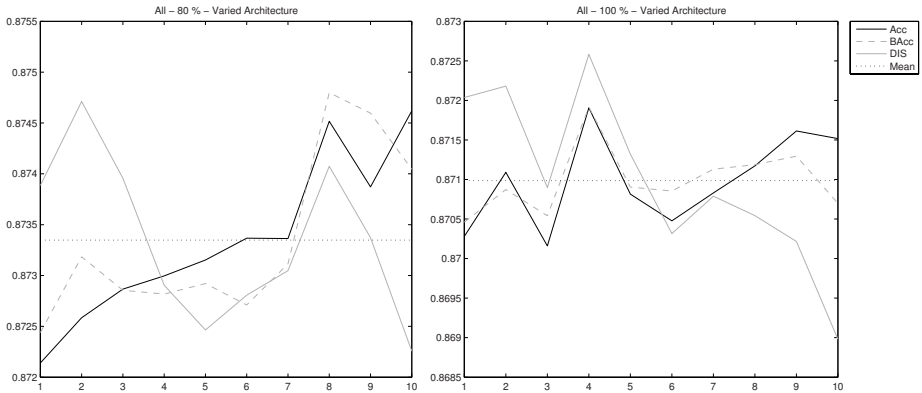


Fig. 2. Test set accuracy vs. ensemble training accuracy, base classifier mean training accuracy and ensemble training diversity. 80% and 100% features, varied architecture and no bootstrap. Averaged over all data sets.

Although it should be no surprise, it is interesting to note that it is usually advantageous to select ensembles with high training accuracy or consisting of highly accurate base classifiers. The most important observation is, however, that it is generally better to select an ensemble with low diversity.

5 Conclusions

In this paper, several measures to introduce implicit diversity in neural network ensembles were evaluated. The main conclusion is that although several setups outperformed the baseline setup, not all methods for producing implicit diversity are successful. In this study, bootstrapping increased diversity but also lowered base classifier accuracy, leading to an overall decrease in ensemble accuracy. Using heterogeneous ensembles, on the other hand, produced more accurate ensembles but without increasing diversity. Resampling using features, finally, lowers base classifier accuracy and increases diversity, but how this affects ensemble accuracy is not clear. The experiments also show that although implicit diversity is often beneficial, it is rarely wise to select a highly diverse ensemble. As a matter of fact, in this study where ANN ensembles are used, it is for most setups better to pick one of the least diverse ensembles.

References

- [1] Dietterich, T.G.: Machine learning research: four current directions. *The AI Magazine* 18, 97–136 (1997)
- [2] Krogh, A., Vedelsby, J.: Neural network ensembles, cross validation, and active learning. In: *Advances in Neural Information Processing Systems*, vol. 2, pp. 650–659. Morgan Kaufmann, San Mateo (1995)

- [3] Brown, G., Wyatt, J., Harris, R., Yao, X.: Diversity Creation Methods: A survey and Categorisation. *Journal of Information Fusion* 6(1), 5–20 (2005)
- [4] Sharkey, N., Neary, J., Sharkey, A.: Searching Weight Space for Backpropagation Solution Types. In: *Current trends in Connectionism: Proceedings of the 1995 Swedish Conference on Connectionism*, pp. 103–120 (1995)
- [5] Duin, R.P.W., Tax, D.M.J.: Experiments with classifier combining rules. In: Kittler, J., Roli, F. (eds.) *MCS 2000. LNCS*, vol. 1857, pp. 30–44. Springer, Heidelberg (2000)
- [6] Kuncheva, L.I., Whitaker, C.J.: Measures of Diversity in Classifier Ensembles and Their Relationship with the Ensemble Accuracy. *Machine Learning* 51, 181–207 (2003)
- [7] Johansson, U., Löfström, T., Niklasson, L.: The Importance of Diversity in Neural Network Ensembles - An Empirical Investigation. In: *The International Joint Conference on Neural Networks*, pp. 661–666. IEEE Press, Orlando (2007)
- [8] Blake, C.L., Merz, C.J., Repository, U.C.I.: of machine learning databases, University of California, Department of Information and Computer Science (1998)

A Simple Characterization on Serially Constructible Episodes

Takashi Katoh¹ and Kouichi Hirata^{2,*}

¹ Graduate School of Computer Science and Systems Engineering

² Department of Artificial Intelligence

Kyushu Institute of Technology

Kawazu 680-4, Iizuka 820-8502, Japan

Tel.: +81-948-29-7622; Fax: +81-948-29-7601

{f673024t, hirata}@ai.kyutech.ac.jp

Abstract. In this paper, we introduce a *parallel-free* episode that always has an arc between vertices with the same label and a *serially constructible* episode that is embedded into every parallel-free episode containing all of the serial episodes occurring in it. Then, we show that an episode is parallel-free if and only if it is serially constructible.

1 Introduction

It is one of the important tasks for data mining to discover frequent patterns from time-related data. For such a task, Mannila [4] have introduced the *episode* to discover frequent in an . Here, the episode is formulated as an of which label is an event type and of which edges specify the temporal precedent-subsequent relation in an event sequence, which proposes a richer representation of temporal relationship than a in (, [5]).

Then, Mannila [4] have designed an algorithm to construct episodes from a as a set of events and a as a sequence of events. Note that their algorithm is general but inefficient. In order to avoid such inefficiency, Katoh have introduced the specific forms of episodes, that is, [2], [3] and [1], and designed efficient algorithms to extract them. Such efficiency follows from the construction of episodes from just information for occurrences of serial episodes in an event sequence, which is obtained by scanning an event sequence just once. On the other hand, there has remained an open problem of

Consider the event sequence W consisting of a pair (e, t) of an event type e and the occurrence time t of e described as Figure 1 (upper left), where all of the serial episodes occurring in W are described as Figure 2 (upper right). Also

* The author is partially supported by Grand-in-Aid for Scientific Research 17200011 and 19300046 from MEXT, Japan.

consider the episodes D_1 , D_2 and D_3 described in Figure 1 (lower) as acyclic labeled digraphs. Note first that all of D_i are embedded into W .

Since D_2 is embedded into W with distinguishing an event type a in D_2 , D_2 is serially constructible from just information for occurrences of serial episodes in D_2 . On the other hand, while D_3 is embedded into W with distinguishing every event type in D_3 , D_3 is serially constructible from just information for occurrences of serial episodes in D_3 , because all of the serial episodes occurring in D_3 coincide with ones in a serial episode in W underlined in Figure 1 (upper right).

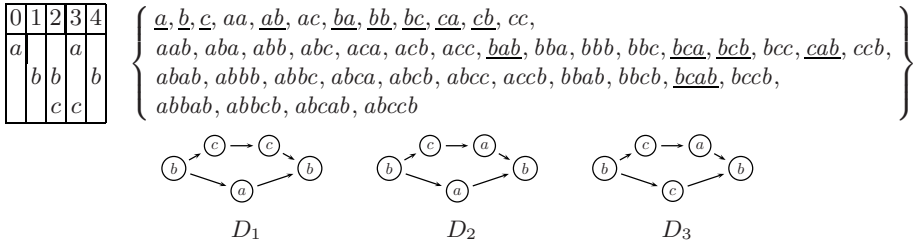


Fig. 1. An event sequence W (upper left), all of the serial episodes occurring in W (upper right), and episodes D_1 , D_2 and D_3 (lower)

In order to solve the problem of

with capturing the above situations, we formulate both an episode and an event sequence as an acyclic labeled digraph (, for short) of which label is an event type. We say that an ATL-digraph D is serially constructible if D always has an arc between vertices with the same label. Also we say that an ATL-digraph D is serially constructible if D is embedded into every parallel-free ATL-digraph containing all of the serial episodes occurring in D . Hence, we show that (as)

2 Episodes as Acyclic Transitive Labeled Digraphs

Let \mathcal{E} be a set of . Then, a pair (e, t) is called an , where $e \in \mathcal{E}$ and t is a natural number which is the () of the event. For a set $E \subseteq \mathcal{E}$ of event types, we denote $\{(e, t) \mid e \in \mathcal{E}\}$ by (E, t) . An S on \mathcal{E} is a triple $\langle S, T_s, T_e \rangle$, where $S = \langle (E_1, t_1), \dots, (E_n, t_n) \rangle$ is an ordered sequence of events such that $E_i \subseteq \mathcal{E}$ ($1 \leq i \leq n$) and $T_s \leq t_1 < \dots < t_n \leq T_e$. A S in $S = \langle S, T_s, T_e \rangle$ is an event sequence $W = (w, t_s, t_e)$ such that $t_s < T_e$, $T_s < t_e$ and w consists of all of the events (e, t) in S for $t_s \leq t < t_e$.

While Mannila [4] have formulated an episode as an , we formulate it as an acyclic labeled digraph as follows.

A (or a) $D = (V, A)$ consists of a finite, nonempty set V of and a (possibly empty) set A of ordered pairs of distinct vertices.

We sometimes denote V , A and $|V|$ by $V(D)$, $A(D)$ and $|D|$, respectively. A digraph (\emptyset, \emptyset) is called **empty** and denoted by \emptyset . An element of A is called an **arc**. For an arc $(u, v) \in A$, u is said to be **tail** of (u, v) and v is **head** of (u, v) . For a digraph D and a vertex $v \in V$, the **in-degree** of v in D , denoted by $d_{in}^D(v)$, is the number of vertices adjacent from v in D and the **out-degree** of v in D , denoted by $d_{out}^D(v)$, is the number of vertices adjacent to v in D . Then, we define $z_{in}^D(D) = \{v \in V \mid d_{in}^D(v) = 0\}$ and $z_{out}^D(D) = \{v \in V \mid d_{out}^D(v) = 0\}$. Hereafter, we refer digraphs (V, A) , (V_1, A_1) and (V_2, A_2) to D , D_1 and D_2 , respectively.

We denote a digraph $(V_1 \cup V_2, A_1 \cup A_2)$ by $D_1 \cup D_2$. For $W \subseteq V$, we denote a digraph $(V - W, A - \{(u, v) \in A \mid v \in W \text{ or } u \in W\})$ by $D - W$. Also we denote $z_{in}^D(D, W) = \{v \in V \mid (v, w) \in A, w \in W\}$ and $z_{out}^D(D, W) = \{v \in V \mid (w, v) \in A, w \in W\}$. Furthermore, D_1 is a **subdigraph** of D_2 if $V_1 \subseteq V_2$ and $A_1 \subseteq A_2$. For $S (\neq \emptyset) \subseteq V$, the **spanning subdigraph** $D[S]$, denoted by $\langle S \rangle_D$, is the maximal subgraph of D of which vertices is S , that is, $\langle S \rangle_D = (S, \{(u, v) \in A \mid u, v \in S\})$.

A **walk** in D is an alternating sequence $w = v_0 a_1 v_1 \cdots a_n v_n$ of vertices and arcs, beginning and ending with vertices, such that $a_i = (v_{i-1}, v_i)$ for $1 \leq i \leq n$, and refer to w as a v_0 - v_n walk. For vertices u and v in V , u is **reachable** to v (in D) if there exists a u - v walk in D . A digraph D is **strongly connected** if there exists no v - v walk in D . A digraph D is **transitive** if, for $u, v, w \in V$, it holds that $(u, w) \in A$ whenever it holds that $(u, v) \in A$ and $(v, w) \in A$. For a set L of labels, a digraph D is **labeled** (by L) if every vertex $v \in V$ has a label $l(v) \in L$. We call an acyclic transitive labeled digraph an **ATL-digraph**. For an ATL-digraph $D = (V, A)$, D^- denotes an acyclic labeled digraph with a minimal number of arcs satisfying that $D = (D^-)^*$, where $(D^-)^* = (V, \{(u, v) \in V \times V \mid u \text{ is accessible to } v \text{ in } D^-\})$. Note that D^- is uniquely determined for every ATL-digraph D .

Two ATL-digraphs D_1 and D_2 are **isomorphic**, denoted by $D_1 \cong D_2$, if there exists a bijection φ from V_1 to V_2 such that $(u, v) \in A_1$ if and only if $(\varphi(u), \varphi(v)) \in A_2$, and $l(v) = l(\varphi(v))$ for every $v \in V_1$. An ATL-digraph D_1 is **isomorphic to a subdigraph** of an ATL-digraph D_2 , denoted by $D_1 \sqsubseteq D_2$, if there exists an injection from V_1 to V_2 such that $(\varphi(u), \varphi(v)) \in A_2$ whenever $(u, v) \in A_1$, and $l(v) = l(\varphi(v))$ for every $v \in V_1$.

In this paper, we formulate an **event sequence** as an ATL-digraph of which label is an event type. Also we formulate a **serially constructible episode** $a_1 \cdots a_n$ [4] as an ATL-digraph $S = (\{v_1, \dots, v_n\}, \{(v_i, v_j) \mid 1 \leq i < j \leq n\})$ such that $l(v_i) = a_i$. We denote the set of all serial episodes embedded into D by $\mathcal{S}(D)$. We deal with an event sequence \mathcal{S} as an ATL-digraph $d(\mathcal{S}) = (V, A)$ satisfying the following conditions.

1. For every event $(e, t) \in \mathcal{S}$, there exists a vertex $v_{e,t} \in V$ such that $l(v_{e,t}) = e$.
2. For every pair $((e, t), (e', t')) \in \mathcal{S} \times \mathcal{S}$ of events, $(v_{e,t}, v_{e',t'}) \in A$ iff $t < t'$.

It is obvious that, for an event sequence \mathcal{S} , $d(\mathcal{S})$ is determined uniquely.

3 Parallel-Free and Serially Constructible Episodes

In this section, we newly introduce a **parallel-free episode** and a **serially constructible episode** as ATL-digraphs. Then, we show the main result of this paper that an episode as an ATL-digraph is parallel-free if and only if it is serially constructible.

Definition 1 (Katoh & Hirata [11]). Let W and D be ATL-digraphs $W = (V_1, A_1)$ and $D = (V_2, A_2)$, respectively. Then, we say that D is parallel-free in W if for every pair $(u, v) \in V_2 \times V_2$ such that $u \neq v$ and $l(u) = l(v)$, it holds that either $(u, v) \in A_1$ or $(v, u) \in A_1$. Also we say that D is parallel-free if D is parallel-free in D itself.

For example, every serial episode is parallel-free. Also $d(\mathcal{S})$ for an event sequence \mathcal{S} is parallel-free. Furthermore, if an ATL-digraph D is parallel-free, then D is parallel-free in an ATL-digraph W such that $D \sqsubseteq W$.

Definition 2 (Katoh & Hirata [11]). An ATL-digraph D is serially constructible if it holds that $D \sqsubseteq W$ for every parallel-free ATL-digraph W such that $(D) \subseteq (W)$.

Definition 2 requires that, for ATL-digraphs D and W , every serial episode in W is corresponding to exactly one serial episode in D . Hence, by regarding D as an episode and W as a window, Definition 2 claims that a window W contains the information of occurrences of serial episodes in D without duplication.

□ Every serial episode is serially constructible. On the other hand, let D and W be ATL-digraphs such that D^- and W^- are described as Figure 2. Then, it holds that W is parallel-free and $(D) = (W) = \{a, b, c, \dots, \dots, \dots, \dots, \dots\}$. However, there exists no injection from $V(D)$ to $V(W)$, so $D \not\sqsubseteq W$. Hence, D is not serially constructible.

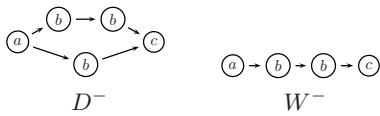


Fig. 2. D^- and W^- in Example 1

For three digraphs $D_i = (V_i, A_i)$ ($i = 1, 2, 3$) (that are possibly empty) such that $V_i \cap V_j = \emptyset$ ($1 \leq i < j \leq 3$) and two sets $B_1 \subseteq \{(u, v) \mid u \in V_1, v \in V_2\}$ and $B_2 \subseteq \{(u, v) \mid u \in V_2, v \in V_3\}$ of arcs, $D_1 \oplus_{B_1} D_2 \oplus_{B_2} D_3$ denotes a digraph (V, A) such that $V = V_1 \cup V_2 \cup V_3$ and $A = A_1 \cup A_2 \cup A_3 \cup B_1 \cup B_2$.

Definition 3. Let $D = (V, A)$ be an ATL-digraph.

1. We say that D has a serially constructible decomposition if there exist ATL-digraphs $D_i = (V_i, A_i)$ ($i = 1, 2, 3$) and sets B_i ($i = 1, 2$) of arcs such that $D^- = D_1^- \oplus_{B_1} D_2^- \oplus_{B_2} D_3^-$. In this case, we denote D by $[D_1, D_2, D_3]$.
2. We say that D has a serially constructible decomposition if there exist ATL-digraphs $D_i = (V_i, A_i)$ ($i = 1, 2, 3, 4$) and sets B_i ($i = 1, 2, 3, 4$) of arcs such that $D^- = (D_1^- \oplus_{B_1} D_2^- \oplus_{B_2} D_4^-) \cup (D_1^- \oplus_{B_3} D_3^- \oplus_{B_4} D_4^-)$ and $D_2 \cup D_3$ is parallel-free in D . In this case, we denote D by $[D_1, \langle D_2, D_3 \rangle, D_4]$.

Lemma 1. $W \quad D \quad E$
 $W \quad [D_1, D_2, D_3] \quad [E_1, E_2, E_3] \quad D_1 \cong E_1$
 $D_3 \cong E_3 \quad D_2 \cup E_2 \quad W$
 $F \quad W$
 $[F_1, \langle F_D, F_E \rangle, F_3]$ □

$$\begin{aligned}
 [F_1, F_D, F_3] &\cong [D_1, D_2, D_3] & [F_1, F_E, E_3] &\cong [E_1, E_2, E_3] \\
 F_1 \cong D_1 (\cong E_1) & & F_3 \cong D_3 (\cong E_3) & \\
 F_D \cong D_2 & & F_E \cong E_2 & \\
 v \in V(F_1) & & v \in V(D_1) & \quad v \in V(E_1) \\
 v \in V(F_3) & & v \in V(D_3) & \quad v \in V(E_3)
 \end{aligned}$$

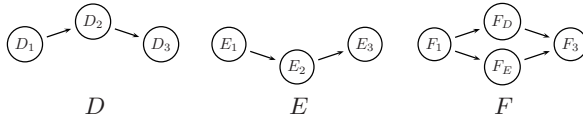


Fig. 3. Intuitive figures of D , E and F in Lemma 1 □

We show the statement by induction on $|D_1|$ and $|D_3|$.

If $|D_1| = |D_3| = 0$, that is, D_1 and D_3 are empty, then E_1 and E_3 are empty since $D_1 \cong E_1$ and $D_3 \cong E_3$. Then, for empty digraphs F_1 and F_3 , the condition 1 holds. Since $D_2 \cup E_2$ is parallel-free, it holds that $[\emptyset, \langle D_2, E_2 \rangle, \emptyset]$ is parallel-free.

Suppose that the statement holds for $|D_1| = |E_1| < n$ and $|D_3| = |E_3| < m$, and consider the case that $|D_3| = |E_3| = m$. Let φ be a bijection on $D_1 \cong E_1$ and $D_3 \cong E_3$. Also let v_1 be a vertex in $fi(D_3)$ and v_2 be $\varphi(v_1) \in E_3$. It is obvious that $v_2 \in fi(E_3)$. Let $D'_1 = D_1$, $D'_2 = D_2$, $D'_3 = D_3 - \{v_1\}$ and $D'_4 = \langle \{v_1\} \rangle_{D_3}$. Then, we can write D as Figure 4. Here, for a set $A_{i,j} = \{(u, v) \in A(D) \mid u \in D_i, v \in D_j\}$ ($1 \leq i < j \leq 3$) of arcs in D , every set $A'_{i,j}$ of arcs from $V(D'_i)$ to $V(D'_j)$ in D ($1 \leq i < j \leq 4$) satisfies the following statements.

$$\begin{aligned}
 A'_{1,4} &= A_{1,3} \cap \{(v, v_1) \in A(D) \mid v \in V(D_1)\}, \quad A'_{1,3} = A_{1,3} - A'_{1,4}, \quad A'_{1,2} = A_{1,2}, \\
 A'_{2,4} &= A_{2,3} \cap \{(v, v_1) \in A(D) \mid v \in V(D_2)\}, \quad A'_{2,3} = A_{2,3} - A'_{2,4}, \\
 A'_{3,4} &= \{(v, v_1) \in A(D) \mid v \in V(D_3)\}.
 \end{aligned}$$

Also let $E'_1 = E_1$, $E'_2 = E_2$, $E'_3 = E_3 - \{v_2\}$ and $E'_4 = \langle \{v_2\} \rangle_{E_3}$. Then, we can write E as Figure 4. Here, for a set $B_{i,j} = \{(u, v) \in A(E) \mid u \in E_i, v \in E_j\}$ ($1 \leq i < j \leq 3$) of arcs in E , every set $B'_{i,j}$ of arcs from $V(E'_i)$ to $V(E'_j)$ in E ($1 \leq i < j \leq 4$) satisfies the following statements.

$$\begin{aligned}
 B'_{1,4} &= B_{1,3} \cap \{(v, v_2) \in A(E) \mid v \in V(E_1)\}, \quad B'_{1,3} = B_{1,3} - B'_{1,4}, \quad B'_{1,2} = B_{1,2}, \\
 B'_{2,4} &= B_{2,3} \cap \{(v, v_2) \in A(E) \mid v \in V(E_2)\}, \quad B'_{2,3} = B_{2,3} - B'_{2,4}, \\
 B'_{3,4} &= \{(v, v_2) \in A(E) \mid v \in V(E_3)\}.
 \end{aligned}$$

Let $D' = D - \{v_1\}$ and $E' = E - \{v_2\}$. Then, it holds that $D' = [D'_1, D'_2, D'_3]$ and $E' = [E'_1, E'_2, E'_3]$. Since $|D'_3| = |E'_3| < m$ and by induction hypothesis, there exist ATL-digraphs F'_1 and F'_3 such that $F' = [F'_1, \langle F'_D, F'_E \rangle, F'_3]$ is parallel-free

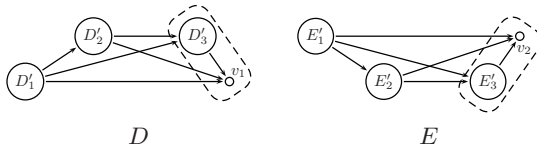


Fig. 4. Intuitive figures of D and E , where the dashed boxes denote D_3 and E_3

and embedded into D , and the statements from 1 to 5 replaced F_D, F_E, F_i, D_i and E_i with F'_D, F'_E, F'_i, D'_i and E'_i hold.

It is sufficient to show how to construct F_3 satisfying the statements. Note that v_1 and v_2 satisfy (a) $v_1 = v_2$, (b) $(v_1, v_2) \in A(W)$ or (c) $(v_2, v_1) \in A(W)$. We denote $A'_{1,4} \cup A'_{2,4} \cup A'_{3,4}$ and $B'_{1,4} \cup B'_{2,4} \cup B'_{3,4}$ by $A'_{*,4}$ and $B'_{*,4}$, respectively.

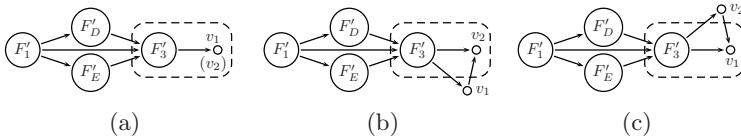


Fig. 5. Intuitive figures of F , where the dashed box is F_3 and we omit the arcs from F'_1, F'_D and F'_E to v_1 and v_2

(a) In the case that $v_1 = v_2$, construct the following ATL-digraph F_3 .

$$F_3 = (V(F'_3) \cup \{v_1\}, A(F'_3) \cup \{(v, v_1) \in A(W) \mid v \in V(F'_3)\}).$$

See Figure 5 (a). We denote an ATL-digraph by adding arcs $A'_{*,4} \cup B'_{*,4}$ to $[F'_1, \langle F'_D, F'_E \rangle, F_3]$ by F . Since F' is parallel-free, so is F . Since $[F'_1, F'_D, F_3] \cong D'$ and $[F'_1, F'_E, F_3] \cong E'$, it holds that $[F'_1, F'_D, F_3] \cong D$ and $[F'_1, F'_E, F_3] \cong E$, which implies the condition 1. By induction hypothesis, the conditions 2, 3 and 4 also hold. For every $v \in V(F_3)$, if $v \neq v_1$, then either $v \in V(D'_3)$ or $v \in V(E'_3)$. Since $D'_3 = D_3 - \{v_1\}$ and $E'_3 = E_3 - \{v_1\}$, it holds that either $v \in V(D_3)$ or $v \in V(E_3)$. If $v = v_1$, then it holds that $v_1 \in V(D_3)$, since $v_1 \in fi(D_3)$. Hence, for every $v \in F_3$, either $v \in V(D_3)$ or $v \in V(E_3)$, which implies the condition 5.

(b) Consider the case that $(v_1, v_2) \in A(W)$. Since $F'_3 \cong D'_3 \cong E'_3$ and $D_3 \cong E_3$, $(v, v_1) \in A(D_3)$ if and only if $(\varphi(v), v_2) \in A(E_3)$, where $v_2 = \varphi(v_1)$. Since either $v \in V(D'_3)$ or $v \in V(E'_3)$ for every $v \in V(F'_3)$, there exists a vertex $v \in V(F'_3)$ such that either $(v, v_1) \in A(D_3)$ or $(v, v_2) \in A(E_3)$. For the former case, if $(v, v_1) \in A(D_3)$, then there exists an arc $(v, v_2) \in A(W)$, since W is transitive and $(v_1, v_2) \in A(W)$. Hence, there exists a vertex $v \in V(F'_3)$ such that $(v, v_2) \in A(W)$, so construct the following ATL-digraph F_3 , see Figure 5 (b).

$$F_3 = (V(F'_3) \cup \{v_2\}, A(F'_3) \cup \{(v, v_2) \in A(W) \mid v \in V(F'_3)\}).$$

We denote an ATL-digraph by adding arcs $A'_{*,4} \cup B'_{*,4}$ to $[F'_1, \langle F'_D, F'_E \rangle, F_3]$ by F . Then, we can check that F satisfies the conditions as similar as the case (a).

(c) For the case that $(v_2, v_1) \in A(W)$, there exists a vertex $v \in V(F_3)$ such that $(v, v_1) \in A(W)$, so construct the following ATL-digraph F_3 , see Figure 5(c).

$$F_3 = (V(F'_3) \cup \{v_1\}, A(F'_3) \cup \{(v, v_1) \in A(W) \mid v \in V(F'_3)\}).$$

We denote an ATL-digraph by adding arcs $A'_{*,4} \cup B'_{*,4}$ to $[F'_1, \langle F'_D, F'_E \rangle, F_3]$ by F . Then, we can check that F satisfies the conditions as similar as the case (a).

We can give the similar proof of the case that $|D_1| = |E_1| = n$. □

Theorem 1.

For a parallel-free ATL-digraph F , we show the statement by induction on $|F|$. If $|F| \leq 1$, then F is a serial episode, so the statement holds.

Suppose that the statement holds for $|F| < n$ and consider the case that $|F| = n$ and F is not a serial episode. Then, there exist vertices u and v in F such that (1) $(u, v) \notin A(F)$ and (2) $(v, u) \notin A(F)$. Let $\dots(F, v) = \{v\} \cup \dots(F, \{v\}) \cup \dots(F, \{v\})$. Since F is transitive, $u \in \dots(F, v)$ implies that either u is accessible to v in F or v is accessible to u in F . Then, for this v , we construct the following ATL-digraphs F_1, F_2, F_3 and F_4 :

$$\begin{aligned} F_2 &= \langle V(F) - \dots(F, v) \rangle_F, \\ F_1 &= \langle \dots(F, \{v\}) \cap \dots(F_2, \{v\}) \rangle_F, \\ F_4 &= \langle \dots(F, \{v\}) \cap \dots(F_2, \{v\}) \rangle_F, \\ F_3 &= \langle V(F) - (V(F_1) \cup V(F_2) \cup V(F_4)) \rangle_F. \end{aligned}$$

For example, consider an ATL-digraph F such that F^- is described in Figure 6 (left). Suppose that $v \in V(F)$ in Figure 6 (left) satisfies the above condition. Then, we obtain F_i^- ($i = 1, 2, 3, 4$) as the dashed boxes in Figure 6 (right).

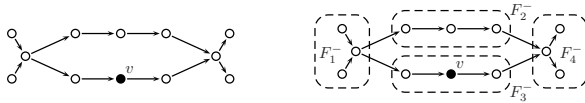


Fig. 6. ATL-digraphs F^- (left) and F_i^- ($i = 1, 2, 3, 4$) (right)

Then, from F , we can construct the ATL-digraphs $X = [F_1, F_2, F_4]$ and $Y = [F_1, F_3, F_4]$. By the construction of F_i , it is obvious that $v \in V(F_3)$ and $u \in V(F_2)$, so it holds that $|X| < n$ and $|Y| < n$. Also it holds that $\dots(X) \subseteq \dots(F)$ and $\dots(Y) \subseteq \dots(F)$. Since F is parallel-free, $F_2 \cup F_3$ is also parallel-free.

Let W be a parallel-free ATL-digraph such that $\dots(F) \subseteq \dots(W)$. Then, it holds that $\dots(X) \subseteq \dots(W)$ and $\dots(Y) \subseteq \dots(W)$. Since X and Y are serially constructible and by induction hypothesis, it holds that $X \sqsubseteq W$ and $Y \sqsubseteq W$.

By regarding X and Y as D and E in Lemma 1, there exists an ATL-digraph $Z = [F_1, \langle F_D, F_E \rangle, F_4]$ that is parallel-free in W and embedded into W . Since $F \cong Z$, it holds that $F \sqsubseteq W$. Hence, F is serially constructible. □

Theorem 2.

It is sufficient to show that, for every ATL-digraph $D = (V, A)$ having a pair $(u, v) \in V \times V$ such that $u \neq v$, $l(u) = l(v)$ and $(u, v), (v, u) \notin A$, there exists an ATL-digraph W such that $(D) \subseteq (W)$ but $D \not\subseteq W$.

Let A_u and $A_{u \rightarrow v}$ be the following sets of arcs.

$$A_u = \{(w, u) \in A \mid w \in (D, \{u\})\} \cup \{(u, w) \in A \mid w \in (D, \{u\})\},$$

$$A_{u \rightarrow v} = \{(w, v) \mid w \in (D, \{u\})\} \cup \{(v, w) \mid w \in (D, \{u\})\}.$$

Also let W be an ATL-digraph $(V - \{u\}, (A - A_u) \cup A_{u \rightarrow v})$. Then, $D - \{u\} (= (V - \{u\}, A - A_u))$ is a subgraph of W .

Let $S_k^n(u) = v_1 \cdots v_{k-1} u v_{k+1} \cdots v_n$ and $S_k^n(v) = v_1 \cdots v_{k-1} v v_{k+1} \cdots v_n$ be serial episodes containing u and v at k with length n ($n \geq 1, 1 \leq k \leq n$). Then, for every $S_k^n(u)$ embedded into D , there exists an $S_k^n(v)$ embedded into W . Since $l(u) = l(v)$, it holds that $S_k^n(u) \cong S_k^n(v)$. Since $D - \{u\}$ is a subgraph of W , every serial episode not containing u and embedded into D is embedded into W . Then, it holds that $(D) \subseteq (W)$. However, since $|W| = |D| - 1$, there exists no injection from $V(D)$ to $V(W)$, which implies that $D \not\subseteq W$. □

4 Conclusion

In this paper, we have shown that

$(\text{Serially Constructible Episodes}) \equiv (\text{Parallel-Free Episodes})$. This equivalence result gives a simple characterization on serially constructible episodes, and then concludes that a parallel-free episode is one of the theoretical limitations on efficiently constructing episodes.

It is a future work to design an efficient algorithm to extract parallel-free episodes from an event sequence. Also it is an important future work to extend a parallel-free and serially constructible episodes to [\[5\]](#).

References

1. Katoh, T., Hirata, K.: Mining frequent elliptic episodes from event sequences. In: Proc. 5th LLLL, pp. 46–52 (2007)
2. Katoh, T., Hirata, K., Harao, M.: Mining sectorial episodes from event sequences. In: Todorovski, L., Lavrač, N., Jantke, K.P. (eds.) DS 2006. LNCS (LNAI), vol. 4265, pp. 137–145. Springer, Heidelberg (2006)
3. Katoh, T., Hirata, K., Harao, M.: Mining frequent diamond episodes from event sequences. In: Torra, V., Narukawa, Y., Yoshida, Y. (eds.) MDAI 2007. LNCS (LNAI), vol. 4617, pp. 477–488. Springer, Heidelberg (2007)
4. Mannila, H., Toivonen, H., Verkamo, A.I.: Discovery of frequent episodes in event sequences. *Data Mining and Knowledge Discovery* 1, 259–289 (1997)
5. Pei, J., Han, J., Mortazavi-Asi, B., Wang, J., Pinto, H., Chen, Q., Dayal, U., Hsu, M.-C.: Mining sequential patterns by pattern-growth: The PrefixSpan approach. *IEEE Trans. Knowledge and Data Engineering* 16, 1–17 (2004)

Bootstrap Based Pattern Selection for Support Vector Regression

Dongil Kim and Sungzoon Cho*

Seoul National University, 599 Gwanangno, Gwanak-gu, Seoul, 151-744, Korea
{dikim01, zoon}@snu.ac.kr

Abstract. Support Vector Machine (SVM) results in a good generalization performance by employing the Structural Risk Minimization (SRM) principle. However, one drawback is $O(n^3)$ training time complexity. In this paper, we propose a pattern selection method designed specifically for Support Vector Regression (SVR). In SVR training, only a few patterns called support vectors are used to construct the regression model while other patterns are not used at all. The proposed method tries to select patterns which are likely to become support vectors. With multiple bootstrap samples, we estimate the likelihood of each pattern to become a support vector. The proposed method automatically determines the appropriate number of patterns selected by estimating the expected number of support vectors. Through the experiments involving twenty datasets, the proposed method resulted in the best accuracy among the competing methods.

1 Introduction

Support Vector Machines (SVM), developed by Vapnik based on the Structural Risk Minimization (SRM) principle [1], has performed with a great generalization accuracy [2]. Support Vector Regression (SVR), a regression version of SVM, was developed to estimate regression functions [3]. SVM is capable of solving non-linear problems, but, has relatively high training time complexity $O(n^3)$ and training memory span $O(n^2)$ where n is the number of training patterns.

Such algorithms as Chunking, SMO, SVM^{light} and SOR have been proposed to reduce the training time with time complexity $T \cdot O(nq + q)$ where T is the number of iterations and q is the size of working set [4]. However, their training time complexities are still strongly related to the number of training patterns. Another direction of research focused on selecting important patterns from the training dataset directly with minimum accuracy loss. SVM-KM [5], NPPS [6], Cross-Training based method [7] and Linear SVM based method [8] have been proposed. However, the binary class labels of training data are used to implement those approaches, which makes those approaches suitable for the classification problems, but not for the regression problems.

Recently, pattern selection methods designed especially for SVR such as HSVM [9] and Sun's method [10] have been proposed. However, both HSVM and Sun's

* Corresponding author.

method have a couple of critical parameters which are to be empirically determined. A stochastic pattern selection method was proposed which estimates the margin space of ε -insensitive tube (ε -tube) [11] and was successfully applied to a response modeling problem [12]. However, those three methods tend to degrade accuracy when they train high dimensional datasets. Moreover the number of patterns selected was manipulated with parameters or thresholds to be determined by users without any guidelines.

In this paper, we propose a bootstrap based pattern selection method. For better performances, we have focused on selecting support vectors. SVR can construct the same regression model with only support vectors. The proposed method tries to identify patterns which are likely to become support vectors. Since support vectors are always located outside the ε -tube under the ε -loss function. by multiple bootstrap samples, the number of times a pattern located outside the ε -tube are calculated, which is used as the likelihood of a pattern to become a support vector. On the other hand, the number of patterns selected is the most critical parameter. The proposed method does not leave the number of pattern selected as a parameter, but determines by itself. We estimate the expected number of support vectors, which is used as the appropriate number of patterns selected. In our experiments, twenty datasets were used while HSVM, Sun’s method and random sampling were adopted as benchmark methods. We compared their results in terms of the training time and Root Mean Squared Error (RMSE).

The remaining of this paper is organized as follows. In Section 2, we provide the main idea of the proposed method and state the algorithm. In Section 3, we present details of datasets and parameters for experiments as well as the result. In Section 4, we summarize the results and conclude the paper with a remark on limitations.

2 Pattern Selection for Support Vector Regression

SVR can train the same regression model even if input patterns were only support vectors. Hence, in some sense, support vectors are an ideal subset to be selected. However, before training, there is no way to identify support vectors from all training patterns. In this paper, we propose a bootstrap approach which estimates a likelihood of each pattern to become a support vector.

We construct k bootstrap samples $D_j = \{(\mathbf{x}_i^j, y_i^j)\}_{i=1}^l$ ($j = 1, \dots, k$) of size l ($l \ll n$) from the original dataset $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$. We then train an SVR with each bootstrap sample D_j and obtain k SVR regression functions f_j ($j = 1, \dots, k$). Every pattern \mathbf{x}_i in the original dataset D is evaluated by each regression function f_j , whether it is located inside or outside the ε -tube. If pattern \mathbf{x}_i is located outside the ε -tube of f_j , the pattern is marked, i.e. $m_{ij} = 1$. We then calculate $L_i = \sum_{j=1}^k m_{ij}$, the number of total markings of the pattern \mathbf{x}_i , which is used as the estimated likelihood of \mathbf{x}_i to become a support vector. At the same time, we calculated the expected number of support vectors $S = \frac{1}{k} \sum_{j=1}^k s_j$, (where $s_j = \sum_{i=1}^n m_{ij}$) by averaging the number of patterns

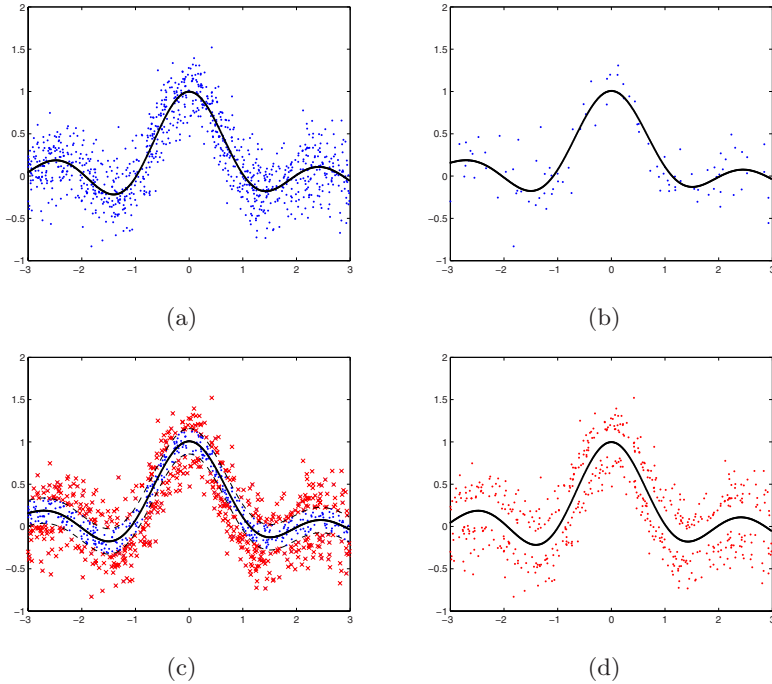


Fig. 1. (a) Original dataset and an SVR trained on it, (b) a bootstrap sample and an SVR trained on it, (c) original dataset patterns outside the ε -tube marked as red and the others as blue and (d) selected patterns and the resulting SVR trained using them

marked by f_j . After calculating L_i and S , we select S patterns deterministically with largest L_i . Or, we select S patterns stochastically based on the probability $\frac{L_i}{\sum_{i=1}^n L_i}$. Finally, an SVR is trained again with the selected patterns. Fig. 1 shows an example of the proposed method with a toy dataset while Fig. 2 presents the algorithm.

The number of patterns selected is a key factor of the proposed method. In the proposed method, a pattern located farther from the regression function is more likely to be selected. If the number of patterns selected is much smaller than the necessary number, only noisy patterns can be selected. On the other hand, if the number of patterns selected is much larger than the necessary number, pattern selection falls into a meaningless effort. By introducing S , we get a guideline of the number of patterns selected.

3 Experimental Results

Our experiments were conducted on twenty datasets including four artificial datasets and sixteen real world benchmark datasets. Real world benchmark

ALGORITHM

1. Initialize the number of bootstrap samples, k
Initialize the number of patterns in each bootstrap sample, l
2. Make k bootstrap samples of size l , D_j for $j = 1, \dots, k$, from the original dataset D by random sampling without replacement
3. Train SVR f_j with D_j, \forall_j
4. Evaluate the original dataset D by f_j, \forall_j
5. $m_{ij} = 1$, if a pattern \mathbf{x}_i is found outside the ε -tubes of f_j (otherwise $m_{ij} = 0$)
6. Calculate $L_i = \sum_{j=1}^k m_{ij}$
7. Calculate $S = \frac{1}{k} \sum_{j=1}^k s_j$, where $s_j = \sum_{i=1}^n m_{ij}$
8. Select S patterns deterministically with largest L_i ,
or select S patterns stochastically without replacement according to $\frac{L_i}{\sum_{i=1}^n L_i}$.
9. Train final SVR with S selected patterns

Fig. 2. The algorithm of the proposed method

datasets including time series datasets were gathered from Delve datasets¹, Time Series Data Library (TSDL)², Statlib³ and Korean Stock Market⁴. All datasets are summarized in Table. II

Artificial dataset 1 was originally introduced from [13] based on a mathematical function, $y = \frac{\sin \pi x}{\pi x} + \xi$ where $x \in [-3, 3]$ and $\xi \sim N(0, 0.5^2)$. Artificial dataset 2 introduced from [11] was generated based on $y = 2 \cos(15x) + (\xi_1 + \xi_2)$ where $x \sim \text{Beta}(1.5, 1)$, $\xi_1 \sim N(0, 0.5^2)$ and $\xi_2 \sim N(0, \sin^2(2(x+1)^2))$. Artificial dataset 3 is newly generated from a mathematical function, $y = \sin(2x) + \xi$ where $x \in [0, 5]$ and $\xi \sim N(0, 0.5^2)$. Add10 dataset is another artificial dataset gathered from the Delve datasets. We used only five relevant input features excluding five noise terms. Time series datasets were re-formulated as regression problems by using the previous 10 values to estimate the following one value, which is a typical way to solve time series problems. The Foreign Exchange dataset was re-formulated as a regression problem to estimate British/US exchange rate by using other 6 nations' exchange rates while the wind dataset was re-formulated to estimate the wind speed of Dublin station using observed other 11 stations' wind speed. For evaluating performances, the original dataset was randomly split into training and test data. The hyper-parameters of SVR were determined by cross-validation with $C \times \varepsilon = \{0.1, 0.5, 1, 3, 5, 7, 10, 20, 50, 100\} \times \{0.01, 0.05, 0.07, 0.1, 0.15, 0.3, 0.5, 0.7, 0.9, 1\}$. RBF kernel was used as a kernel function and the kernel parameter σ was fixed to 1.0 for all datasets. All datasets were normalized.

HSVM, Sun's method and random sampling were implemented to be compared. For HSVM, the partitioning parameter was set to 10 and similarity threshold was set to be the average value of similarities of all patterns. For

¹ Delve Datasets: <http://www.cs.toronto.edu/~delve/data/datasets.html/>

² TSDL: <http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/>

³ Statlib: <http://lib.stat.cmu.edu/datasets/>

⁴ Korean Stock Market: <http://www.kse.or.kr/>

Table 1. Summary of the datasets used in the experiments

Numb.	Name	# Train	# Test	# Attribute	Origin	Feature
1	Artificial Dataset 1	1000	1000	1	[13]	Artificial
2	Artificial Dataset 2	1000	1000	1	[11]	Artificial
2	Artificial Dataset 3	1000	1000	1	Generated	Artificial
4	Add10	2000	2000	5	Delve Datasets	Artificial
5	Santa Fe A	2000	2000	10	Santa Fe Competition	Time series
6	Santa Fe D	2000	2000	10	Santa Fe Competition	Time series
7	Santa Fe E	1500	500	10	Santa Fe Competition	Time series
8	Sun Spot	1500	500	10	TSDL	Time series
9	Melbourne Temperature	2000	1000	10	TSDL	Time series
10	Gold	700	300	10	TSDL	Time series
11	Daily IBM	2000	1300	10	TSDL	Time series
12	KOSPI 200	2000	1000	10	Korean Stock Market	Time series
13	S&P 500	2000	1000	10	TSDL	Time series
14	Foreign Exchange	2000	2000	7	TSDL	Non-time series
15	Wind	2000	2000	11	Statlib	Non-time series
16	Abalone	2000	2000	10	Delve Datasets	Non-time series
17	Bank	2000	2000	8	Delve Datasets	Non-time series
18	Census House	2000	2000	8	Delve Datasets	Non-time series
19	Computer Activity	2000	2000	12	Delve Datasets	Non-time series
20	Pumadyn Family	2000	2000	8	Delve Datasets	Non-time series

Sun’s method, the k of k -NN was fixed to 5 and the number of patterns selected was set to be similar to S from the proposed method. The parameters of the proposed method, k and l , were fixed to 10 and 10% of n , respectively. We evaluated the performances of each method by RMSE and training time (sec.). All experimental results were averaged over 30 repetitions.

Fig. 3 shows the experimental results of artificial datasets including three artificial datasets and Add10 dataset from the Delve datasets. The pairs of RMSE and training time in seconds are plotted corresponding to each method. The closer a result is plotted to the origin, the better the method performs. The solid line indicates the results of the random sampling from 10% to 100% of n . Marked squares and marked circles are experimental results of the proposed method with the deterministic selection (DET) and the stochastic selection (STO), respectively. The results of random samples were polynomially decreased as the number of patterns selected goes smaller. As Fig. 3 shows, the propose method shows competitive results. The proposed method resulted better performances than benchmark methods in terms of pairs of RMSE and training time. Random sampling resulted the best for artificial dataset 1. In this case, artificial dataset 1 was so easy that random sampling can handle it sufficiently.

Table 2 shows the experimental results of sixteen real world datasets in terms of RMSE. ‘# Dataset’ represents the index number of datasets given in Table 1 while ‘R30’, ‘R50’ and ‘R70’ represent the random sampling with 30%, 50% and 70% of original patterns, respectively. The deterministic selection of the proposed method resulted the best accuracy among the benchmark methods for fifteen datasets. Table 3 shows the observed training time. Each cell of Table 3 represents the percentage of training time of a method compared to original training time. HSVM and the stochastic selection of the proposed method ranked 1st for eight datasets and for five datasets, respectively. The deterministic

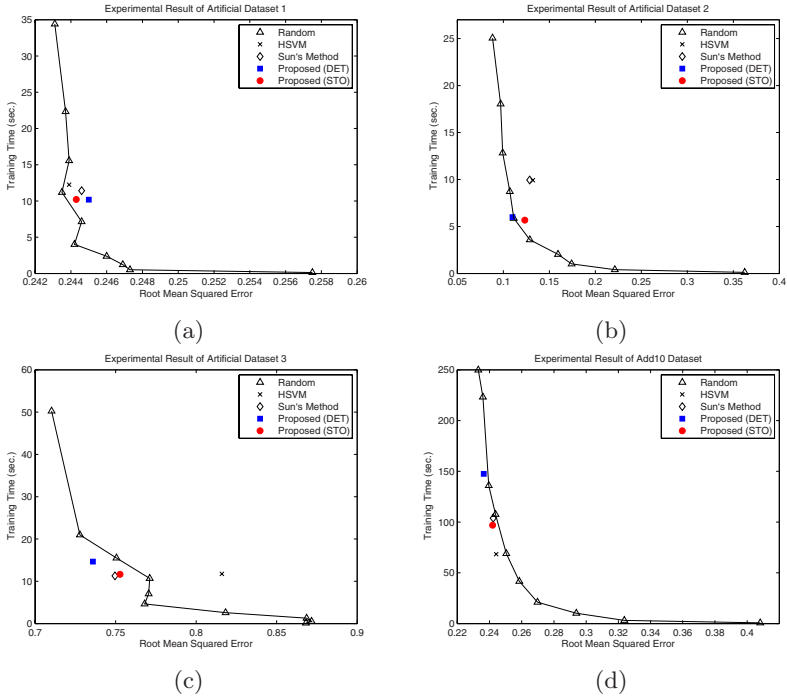


Fig. 3. Experimental results of artificial datasets

Table 2. Experimental results of sixteen real world datasets (RMSE)

# Dataset	R30	R50	R70	HSVM	Sun's Method	Proposed (DET)	Proposed (STO)
5	0.0458	0.0368	0.0321	0.0368	0.0352	0.0313	0.0318
6	0.3457	0.3058	0.2817	0.2861	0.2839	0.2693	0.2827
7	0.5595	0.5457	0.5379	0.5352	0.5439	0.5297	0.5316
8	0.6575	0.6325	0.6092	0.6002	0.5906	0.5840	0.5932
9	0.7359	0.7226	0.7175	0.7226	0.7201	0.7112	0.7115
10	0.2007	0.1865	0.1785	0.1630	0.1756	0.1668	0.1691
11	0.0707	0.0687	0.0665	0.0693	0.0694	0.0681	0.0690
12	0.0887	0.0862	0.0829	0.0843	0.0832	0.0826	0.0839
13	0.4456	0.4414	0.4237	0.4017	0.4709	0.3958	0.4284
14	0.0991	0.0821	0.0755	0.0814	0.0735	0.0721	0.0745
15	0.3735	0.3302	0.2977	0.3060	0.2807	0.2704	0.2821
16	0.7011	0.6819	0.6718	0.6841	0.6802	0.6647	0.6667
17	0.9299	0.9054	0.8909	0.9069	0.8922	0.8748	0.8789
18	0.7871	0.7616	0.7419	0.7608	0.7410	0.7312	0.7387
19	0.5720	0.5227	0.5001	0.5329	0.4829	0.4745	0.4852
20	0.8065	0.7672	0.7415	0.7623	0.7590	0.7164	0.7230

selection of the proposed method used much training time than HSVM, but, still used only around 15~50% of original training time.

Table 3. Experimental results of sixteen real world datasets. Each cell indicates the percentage of training time of a method compared to original training time.

# Dataset	HSVM	Sun's Method	Proposed (DET)	Proposed (STO)	# Dataset	HSVM	Sun's Method	Proposed (DET)	Proposed (STO)
5	69.83	79.64	45.40	44.52	13	23.57	31.94	2.91	3.10
6	26.56	38.58	40.29	33.67	14	35.25	38.85	46.07	40.51
7	23.49	26.79	24.81	27.68	15	22.76	47.67	52.26	48.10
8	24.85	20.14	17.43	14.58	16	33.46	20.83	17.46	15.38
9	15.48	16.54	14.97	14.41	17	21.59	27.70	37.47	34.40
10	29.95	98.01	16.15	17.60	18	20.61	27.65	30.47	27.11
11	23.64	18.19	20.69	16.74	19	17.76	27.93	36.29	28.57
12	28.69	27.62	41.78	35.55	20	22.22	27.77	42.65	40.10

Table 4. Summary of the experimental results

		HSVM	Sun's Method	Proposed (DET)	Proposed (STO)
	Frequency of ranking 1 st	2	0	18	0
	Averaged Std.	0.0100	0.0085	0.0055	0.0091
	Averaged Sensitivity (%)	-	-	87.78 (73.44~96.96)	73.44 (54.12~84.45)
Comparison to train all patterns	RMSE Increased (%)	8.63	7.36	2.58	5.16
	Training Time Used (%)	28.58	49.3	30.35	26.75

All the experimental results are summarized in Table. 4. Compared to benchmark methods, the deterministic selection of the proposed method, which showed the smallest standard deviation, ranked 1st for eighteen datasets over twenty datasets in terms of RMSE. Averaging over all experimental results, the proposed method with the deterministic selection can train SVR using 30.35% of original training time with only 2.58% of increased error. The proposed method with the stochastic selection increases RMSE about 2.5% than the proposed method with the deterministic selection, but it was still competitive. The sensitivity analysis shows that the proposed method selected on average 87.78% deterministically and 73.44% stochastically of the actual support vectors.

4 Conclusion

This paper provides a new pattern selection method to reduce training time of SVR. Only those patterns that were likely to become support vectors were selected and used for training. The proposed method automatically determined the number of patterns selected, which was a key factor of obtaining good results. Twenty datasets including sixteen real world datasets were analyzed. The results showed that the generalization performance of the proposed method was better than other benchmark methods. It performed well for diverse datasets.

Another strong point of the proposed method is that it has fewer critical parameters than benchmark methods. Several parameters such as similarity threshold and the number of patterns selected affect the accuracy but are ambiguous to users. Not only is there no guideline for those parameters, but also a parameter set determined best to a certain dataset is rarely best to other datasets. However,

the proposed method only needs to set k and l which do not affect the accuracy directly. In this paper, the proposed method showed good performances even though we used only one fixed parameter set.

There are limitations of the current work. The result of the proposed method can be largely affected by the number of support vectors. This method may select too many patterns for some applications that have a lot of support vectors.

Acknowledgement

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korea government (R01-2005-000-103900-0), the Brain Korea 21 program in 2007, and partially supported by Engineering Research Institute of SNU.

References

1. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
2. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge (2000)
3. Drucker, H., Burges, C.J.C., Kaufman, L., Smola, A., Vapnik, V.: Support Vector Regression Machines. In: Mozer, M.C., Jordan, M.I., Petsche, T. (eds.) Advances in Neural Information Processing System, vol. 9. MIT Press, Cambridge (1997)
4. Platt, J.C.: Fast Training of Support Vector Machines Using Sequential Minimal Optimization. In: Advanced in Kernel Methods; Support Vector Machines, pp. 185–208. MIT Press, Cambridge (1999)
5. Almeida, M.B., Braga, A., Braga, J.P.: SVM-KM: Speeding SVMs Learning with a Priori Cluster Selection and k -Means. In: Proc. of the 6th Brazilian Symposium on Neural Networks, pp. 162–167 (2000)
6. Shin, H., Cho, S.: Neighborhood Property based Pattern Selection for SVM. Neural Computation 19(3), 816–855 (2007)
7. Bakir, G.H., Bottou, L., Weston, J.: Breaking SVM Complexity with Cross-Training. In: Advances in Neural Information Processing Systems, vol. 17, pp. 81–88 (2005)
8. Joachims, T.: Training Linear SVMs in Linear Time. In: Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 217–226 (2006)
9. Wang, W., Xu, Z.: A Heuristic Training for Support Vector Regression. Neurocomputing 61, 259–275 (2004)
10. Sun, J., Cho, S.: Pattern Selection for Support Vector Regression based on Sparsity and Variability. In: 2006 IEEE International Joint Conference on Neural Networks (IJCNN), pp. 559–602 (2006)
11. Kim, D., Cho, S.: ϵ -tube based Pattern Selection for Support Vector Machines. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 215–224. Springer, Heidelberg (2006)
12. Kim, D., Lee, H., Cho, S.: Response Modeling with Support Vector Regression. Expert Systems with Applications 34(2), 1102–1108 (2008)
13. Chalimourda, A., Schölkopf, B., Smola, A.: Experimentally Optimal ν in Support Vector Regression for Different Noise Models and Parameter Settings. Neural Networks 17, 127–141 (2004)

Tracking Topic Evolution in On-Line Postings: 2006 IBM Innovation Jam Data

Mei Kobayashi and Raylene Yung

IBM Tokyo Research Laboratory,
1623-14 Shimotsuruma, Yamato-shi, Kanagawa-ken 242-8502 Japan
mei@jp.ibm.com, rayleney@gmail.com

Abstract. Participants in on-line discussion forums and decision makers are interested in understanding real-time communications between large numbers of parties on the internet and intranet. As a first step towards addressing this challenge, we developed a prototype to quickly identify and track topics in large, dynamic data sets based on assignment of documents to time slices, fast approximation of cluster centroids to identify discussion topics, and inter-slice correspondence mappings of topics. To verify our method, we conducted implementation studies with data from *Innovation Jam 2006*, an on-line brainstorming session, in which participants around the globe posted more than 37,000 opinions. Results from our prototype are consistent with the text in the postings, and would have required considerable effort to discover manually.

Keywords: discussion mining, topic detection and tracking.

1 Introduction

The past decade has witnessed the proliferation of computers and an explosive growth in electronically stored data. Contributions by the data mining community have been helpful for managing data archives, however, some traditional approaches are inadequate, because the volume of output from analyzing massive archives is too large for human understanding. Temporal analysis of dynamic databases has added another dimension to an already difficult technical problem. Nevertheless, impressive progress has made in understanding dynamic text documents sets. Examples of work in this area include analysis of news articles [1], Web search logs, e-mails, blogs [2], customer call center data, and US patent data. Recently, researchers have successfully identified and tracked the evolution of topics in massive archives of text documents – over 100 years of journal archives [3]; 9 months of personal e-mail, 17 years of [4], and over 200 years of presidential state-of-the-union addresses [7]. The works are detailed and accurate, but require extensive computations.

The focus of our work is to develop computationally fast methods to perform analysis of the content of very large and dynamically evolving sets of

¹ National Inst. of Standards & Technology Text REtrieval Competition: trec.nist.gov

² *blogpulse*TM is a trademark of Nielsen BuzzMetrics, Inc.: www.blogpulse.com

unformatted text data, such as on-line discussion records, internet postings, and e-mail. Detailed analysis is forsaken for speed since prospective participants need a quick means for browsing topics and catching up on on-going discussions.

Two works that are very closely related to ours analyzed on-line discussion data from a different perspective. Spangler et al. [6] developed a system to help human analysts in interactive text mining using *vector space models* (VSMs) and a k-means algorithm to generate the classes for postings from three on-line discussion sessions, each of which produced thousands of postings by thousands of participants around the world over a 72-hour period: [6] (2003), [7] (2004), and [8] (2005). The goal of our work is not just to aid, but to *reduce* human labor as much as possible.

Murakami et al. [4] developed a system to analyze a dataset from Innovation Jam (2006) by creating network graphs of relationships between postings and the *relationships* of individual postings. Graphs of postings (nodes) and their relationships (edges, lines) are based on data from the "relationships" feature. Phrases that express opinions about previous postings (e.g., "I like..." or, "I don't like...") are extracted as a first step for evaluating the importance of a posting. Five types of opinions are identified using pattern dictionaries:

"I like...", "I don't like...", "I love...", "I hate...", and "I am interested in...". Opinions that are a *direct* response to a posting are given greater importance weight than opinions that are a *response* to a posting. Users can quickly spot a posting that is central to discussions since the size of a node is proportionate to its importance value.

We developed a prototype to quickly and automatically identify topics and track their flow during on-line discussions, i.e., unedited text postings by thousands of different people. The system identifies topics through computationally inexpensive and fast approximation of cluster centroids. To track the evolution of topics, documents are assigned to overlapping slices of manageable size, based on time stamp information. Documents on a topic of interest that lie in an overlap are used to find similar documents in the subsequent slice (Section 2). We tested our method through implementation studies using our prototype and real-world data from I-Jam, a world-wide on-line discussion session (Section 3). The final section proposes some directions for future research (Section 4).

2 Topic Identification and Tracking

This section introduces algorithms for identifying and tracking topics. We present an overview of steps performed by our prototype before details of the algorithms.

To enable fast processing, our system divides a large set of documents into overlapping time slices, each of which contains several thousand documents (Fig. 1). Next, it applies a text parser [5] and automatically constructs a VSM with *term frequency-inverse document frequency* (tf-idf) weighting for each time slice since topics and word usage change over time. We use the cosine of the angle defined by vectors to determine document similarity. When the collection of documents is large, principal component analysis (PCA) will be used to address the curse of dimensionality problem. The VSM will be projected

onto a subspace spanned by the first few hundred principal components of the document-keyword matrix. PCA-based dimensional reduction also resolves some difficulties caused by noise (e.g., misspellings, acronyms, slang), synonymy and polysemy and reduces the need for constructing domain-specific dictionaries for unedited, unstructured text.

Our system applies a topic identification algorithm (Algorithm 1) to the first time slice to find sets of documents on similar topics. Documents in the overlap between the first and second time slices are identified by their document numbers, which are assigned in chronological order with respect to posting time. Documents that lie in the overlap help track a topic. Their corresponding document vectors in the second time slice are input as queries to find documents on similar topics that were posted during the latter half of the second time slice (Fig. 1). The system carries out an analogous procedure to identify and track topics in subsequent time slices.

Algorithm 1. *Identify documents on similar topics in a slice*

1. Assign documents in collection to overlapping time slices.
2. Construct a vector space model for each time slice.
3. Apply Algorithm 1 to the first time slice to identify topics.
4. Use Algorithm 2 and documents in the overlap between the first and second time slices to track the evolution of topics.
5. Follow an analogous procedure for subsequent time slices.

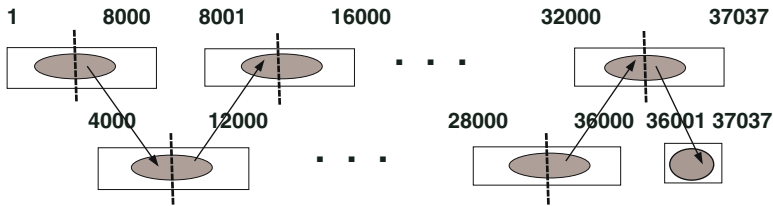


Fig. 1. Assign data to overlapping slices of 8000 posts each. Identify documents on similar topics in a slice (represented by ovals), then use documents in slice overlaps as seeds to find documents on similar topics in the next time slice.

Our prototype uses principal components as queries because the vectors point in directions with the most information about the contents of a database. Initial steps of a clustering algorithm identify sets of documents on a topic [2]. The algorithm must be a *fuzzy* clustering algorithm (i.e., it permits cluster overlaps), because topics that appear in postings may be inter-related. Since our goal is to find topics (not clusters), our system navigates around the difficulty of determining borders of clusters and the cohesiveness of clusters by using only documents that lie in the core of clusters to estimate the centroid. The centroid vector gives a keyword description of topics covered in the cluster. Empirical observations from implementations indicate that using 20 to 30 documents in the

cluster core is sufficient for computing a good approximation of the centroid for topic identification. Our system does not automatically use 30 top ranking documents to approximate a centroid, because some small clusters may have fewer than the default number. To check the cluster size, our prototype computes the plot of document relevancy (vertical axis) as a function of document relevancy ranking (horizontal axis). If the slope undergoes a large change before the 30th ranked document, the system only considers documents left of the change point to approximate the centroid (Fig. 2).

f_i

1. Make vector space model of documents.
2. Compute principal components of document vectors.
3. Perform dimensional reduction (if necessary).
4. Find documents on the same topic.
 - Find documents along principal components.
 - Use a high ranking document as query.
 - Retrieve documents with high relevancy ranking.
5. Find set of documents to approximate centroid of topic.
 - Sort documents (from highest to lowest relevancy).
 - Plot sorted documents as a function of relevancy.
 - Compute rate of change in relevancy.
 - Check for sudden changes before 30th document (Fig. 2).
6. Compute weights of keywords for centroid
 - Sum weights for all keywords in top 30 documents (or all documents for smaller clusters).
 - Divide by number of documents.
 - Order keywords (from largest to smallest weight).

To track the evolution of topics, divide the document set into overlapping time slices, and construct a new VSM for each time slice since keywords may become obsolete, and important new keywords may be introduced over time. Identify topics and relevant documents for the first time slice using Algorithm 1. To track a topic to the second time slice, use Algorithm 2. Identify documents that lie in the slice overlap using document identification numbers that have been assigned in chronological order, according to the posting time. Documents that lie in the latter half of a time slice are used to find similar documents and topics in next slice (Fig. 1). An analogous process for identifying and tracking topics is carried out for subsequent time slices.

1. Assign docs to overlapping time slices (Fig. 1).
2. Perform VSM of docs for each time slice.
3. Identify topics and relevant docs in first time slice (Alg. 1).
4. Track topic from the first to second time slice by identifying docs in slice overlap. Use them to find similar topics in second time slice (Fig. 1).

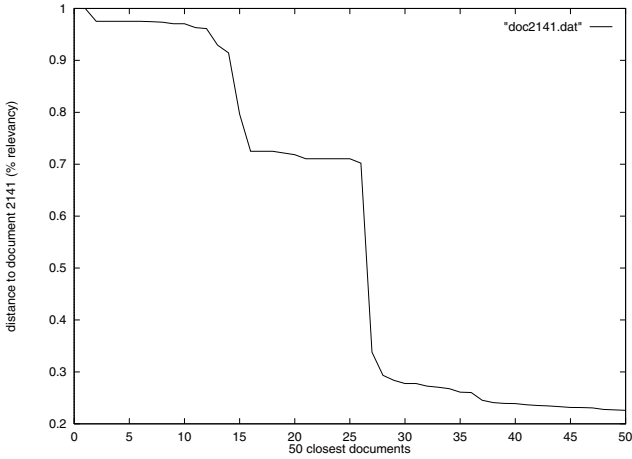


Fig. 2. Identify topics covered by a set of document vectors close to *document 2141* by computing an approximation for the centroid. Up to 30 documents close to document 2141 are used. If the relevancy ranking curve experiences a sharp drop before the 30th document, only documents to the left of the drop are used for approximating the centroid. In this example, the top 12 documents are used to approximate the centroid.

5. Find new topics in second time slice (as in Step 3)
6. Track topic in subsequent time slices using analogous procedure (Steps 4-5).

3 Implementation Studies

We conducted implementation studies with real-world data to identify and track topics in a collection of opinions (text) that were posted during the first phase of [IBM Innovation Jam](#) (I-Jam), an on-line brainstorming session organized by IBM Corporation³ [6]. Participants could post new trains of thought or reply to existing postings in any one of several categories, known as [topic areas](#). Although postings were more formal than chat rooms, they still contained material not commonly found in edited text (e.g., typos, grammatical errors, slang, emoticons). Participants were permitted but not required to read posts by others.

During the first Jam, participants were asked to rate ideas in postings, help analyze events, and communicate findings to other participants. This arrangement led to an unintentional bias favoring ideas introduced early on. Over successive Jams, text mining technologies were introduced to help analyze postings. However, the number and rate of postings has increased so much that a new generation of more powerful tools are needed to:

³ www.globalinnovationjam.com/get_started2006/

- find and identify topics under discussion;
- determine the discussion set to which each untagged posting⁴ belongs;
- analyze and format results in near real-time during a Jam so people can understand enough to begin participating at any intermediate time⁵;
- analyze postings in-depth after conclusion of a Jam for deeper understanding and insight about opinions and views of participants.

Data analysis software reduces labor and is relatively free of unintentional human bias. We believe that systems, such as ours, that can analyze large collections of text documents in (almost) real-time, with low associated hardware costs, will interest Jam organizers and decision-makers for large organizations.

3.1 2006 Innovation Jam Data Set

The 2006 I-Jam consisted of two distinct phases. The aim of the first was to create new ideas, then concrete proposals for design and implementation of prototypes. The ideas were reviewed off-line to determine a set of top proposals. The focus of the second phase was testing and refining the top proposals. We analyzed over 37,000 opinions posted during the first phase over a 78 hour period [4] on four pre-selected subjects:

- Going Places - transforming travel, transportation, recreation & entertainment;
- Finance & Commerce - the changing nature of global business and commerce;
- Staying Healthy - the science and business of well-being; and
- A Better Planet - balancing economic and environmental priorities.

Omission of 200-300 postings in languages other than English, left a final set of 37,037 documents. We divided the data set into overlapping time slices consisting of 8000 documents each. Each slice had 50% overlap with those directly preceding and following it so that all documents belong to one overlap, with the exception of those in the first half of the first time slice and the latter half of the last time slice. For larger data sets, more documents may be assigned to each slice with a different degree of slice overlap, depending on the intended use, properties of the data, hardware requirements, and financial constraints.

A shallow parser extracted keywords (nouns, verbs, adjectives) and conducted parts-of-speech tagging [5]. Standard and domain specific stopwords were deleted, as were words that appeared less than 10 times in a slice. If a larger number of documents are assigned to each slice, then the cutoff point for removing rare terms should be set higher. Principal components of the document vectors in each slice were computed using SVDLIBC, version 1.34⁶. Each slice contained 8000 documents and approximately 3000 keywords. Given extracted keywords and their frequency counts, the computations to determine the tf-idf VSM and all 8000 principal components took about 10 minutes with no special tuning

⁴ The posting has not been marked by the authors as a "reply-to" another posting.

⁵ Recent participants have commented on the difficulty of joining discussions once the Jam has started since there are too many back postings to read.

⁶ SVDLIBC: www.tedlab.mit.edu:16080/~dr/SVDLIBC/

for each time slice on an IBM ThinkPad T42, model 2373-J8J with 1.0 GB of main memory. We can expect much faster performance in a working system since only the first several hundred (not all) components will not be computed. Computations will be easily under a minute with tuning and increase in the main memory.

Table 1. Results from query using 2nd principal component (PC)

doc. #	dist to PC	title
2141	0.168724	Generating power for the plants
1932	0.267695	Saving energy
2251	0.249761	Generate solar to consume less fossil fuels
2542	0.153877	IBM can go GREEN
4083	0.145951	IBM becoming a role model
4597	0.136097	More solar energy
4607	0.134192	Maximizing existing resources ...

3.2 Experimental Results

To find topics in the first time slice, principal components were input as queries, and the most relevant documents were used as a queries to identify important topics. Typical results from experiments are given in Table 1 for the second principal component, which consists of major keywords:

and in the positive direction and and in the negative direction. The relevancy ranking curve for document distances to the second principal component experiences a sharp drop-off just after the tenth ranked document (Fig. 2), so we use it as the cut-off point for approximating the centroid (note the 30 document default). A number of interesting topics were seen to evolve over ten time slices, as shown by two representatives examples below: and

Topic 1: food, health & exercise. The weights of twelve predominant keywords had relatively large changes during discussions:

, and . Tracking three keywords () with relatively large changes over 10 time slices shows the flow of topics in the postings (Fig. 3). During the sixth time slice, curves for and have local maxima, while has a local minimum. However, by the eighth time slice, has its global maximum, while and are in decline. Samples of unedited quotations from the sixth and eighth time slices (marked with 6 or 8), with typos, acronyms, and facial icons, are consistent with these findings:

- (6) ... (7/25/06 11:41 PM) ... Finally someone who nows and takes care about health c We should enjoy our lives and work ...
- (6) ... (7/25/06 10:28 PM) ... we need to step up on providing and urging IBMers into a better and healthy lifestyle ... This might include exercises, games, healthy foods, etc. A healthy IBMer will definitely contributes better than an unhealthy one!!=)
- (6) ... (7/26/06 4:22 AM) Yes, we needs a Gym at IBM. This will keep us healthy and fresh ...
- (8) ... (7/27/06 12:25 AM) Why check after eating? Why not before eating?
- (8) ... (7/27/06 2:16 AM) Yes i agree, being active is just as important as eating the right food when it comes to staying healthy .but what really are the best foods for us to eat.

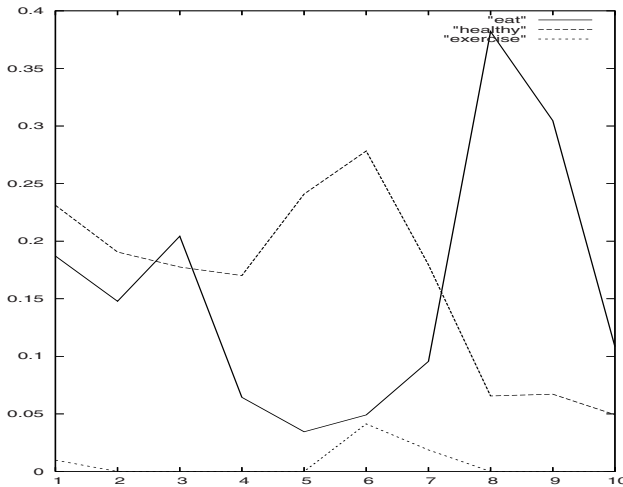


Fig. 3. Weight of keywords *eat*, *exercise*, and *healthy* (y-axis) in the centroid as a function of the time slice (x-axis)

Topic 2: Methods for payment & security. The weights of four keywords had relatively large changes over 10 time slices:

(Fig. 4). During the third time slice, payment and security dominate the discussion. By the fifth time slice, the discussion shifts to signature and methods for authentication. There is another shift in the discussion by the eighth time slice to security issues. Some samples of unedited postings from the third, fifth, and eighth time slices that support the findings using our system are:

- (3) ... (7/25/06 1:19 AM) ... many people owns multi credit cards of different banks, if the data of these transaction data can be shared between banks, bank and credit card owner will both benefit. of cause, there may be some privacy and security issues ...

(5) ... *fi* ... (7/25/06 8:58 PM) I think we need alternative signature when we use our credit card. If your arm broken and you can't sign, how can we use our credit card? How about using our finger print or eye identity? ...

(8) ... (7/26/06 10:07 PM) ... There are however issues that have to be considered specifically: 1. Who will administer the card? 2. Additional equipment may be required in order to use the card properly (or rather, will available equipment be able to read it?) 3. If the card is lost/stolen, the replacement process may not be so trivial ...

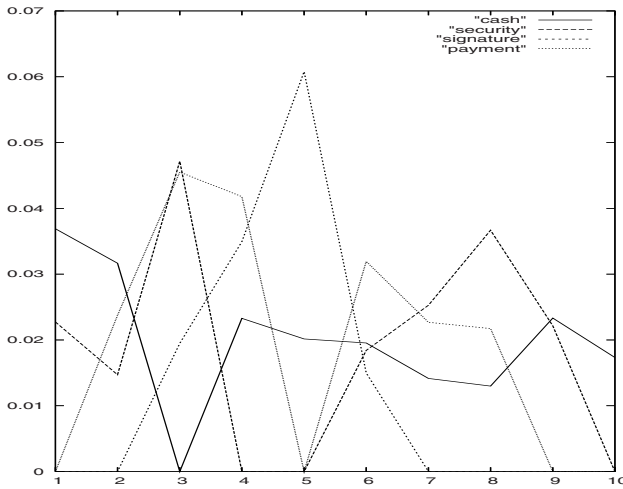


Fig. 4. Weight of keywords *money*, *payment*, *security*, and *signature* (y-axis) in the centroid as a function of the time slice (x-axis)

4 Conclusions and Directions for Future Work

We developed and implemented a prototype to automatically identify and track topics in a text data set consisting of time-stamped postings from on-line discussions. The prototype divides the dataset into overlapping time slices and automatically computes a vector space model and principal components for each time slice. Principal components are used as queries to find postings on important topics. Postings in the overlap of time slices are used to track topics from the earlier time slice to the subsequent time slice. Our prototype shows the evolution of topics in discussions by displaying changes in the contributions of keywords in relevant documents in each time slice. The results can help users understand the overall flow of a discussion without having to read individual postings.

In future work, our first task is to develop methods to speed up computations and automate tuning for arbitrary data sets, for example, determining the minimum number of principal components to find topics and the "right size" for

time slices. Small time slices enable fast computation during PCA. However, they may not reliably find slowly evolving topics or minor topics. Large time slices are better for finding major and minor topics, but incur a higher computational cost, especially if PCA-based dimensional reduction is needed.

A second area to explore is use of supplementary information such as the "reply-to" feature input by users in on-line postings and e-mail. The reliability of reply-to information depends on the reliability of participants who post and may be inaccurate. Merging information output by our prototype graph-based analysis systems is likely to be highly beneficial.

A third area of interest is improvement of the GUI. We conducted some preliminary studies with a real-time binary k-means-based method for displaying an overview of related topics in the database [3]. The bird's eye, macroscopic view is useful for a quick overview of results for users who do not have a clear idea of what to search and need to browse before zooming in on a topic of interest.

A fourth area to investigate is refinement of the mathematical model for documents. Currently, a very good shallow parser extracts terms for vector space modeling. We would like to incorporate advanced natural language processing technologies, such as key phrase extraction and identification of patterns that express sentiments (e.g., positive, negative, neutral opinions).

Acknowledgments. The authors thank the IBM Natural Language Processing team at the TJ Watson Lab for use of their English parser and A. Murakami at IBM Research, Tokyo for providing data and helpful technical discussions.

References

1. Blei, D., Lafferty, J.: Dynamic topic models. In: Proc. ICML, pp. 113–120. ACM, New York (2006)
2. Kobayashi, M., Aono, M.: Vector space models for search and cluster mining. In: Berry, M. (ed.) *Survey of Text Mining*, ch. 5, pp. 103–122. Springer, Heidelberg (2004)
3. Kobayashi, M., Kim, W.: Private communication on work in progress (2007)
4. Murakami, A., et al.: Innovation Jam: analysis of online discussions records using text mining technology. In: Proc. Int'l. Wkshp. on Intercultural Collaboration. LNCS. Springer, Heidelberg (to be published)
5. Neff, M., Byrd, R., Boguraev, B.: The Talent system. *Natural Language Eng.* 10, 307–326 (2004)
6. Spangler, W., Kreulen, J., Newswanger, J.: Machines in the conversation. *IBM Systems Journal* 45, 785–799 (2006)
7. Wang, X., McCallum, A.: Topics over time. In: Proc. KDD, pp. 424–433. ACM, New York (2006)

PAID: Packet Analysis for Anomaly Intrusion Detection

Kuo-Chen Lee, Jason Chang, and Ming-Syan Chen

National Taiwan University

{kcllee, jason}@arbor.ee.ntu.edu.tw, mschen@cc.ee.ntu.edu.tw

Abstract. Due to the growing threat of network attacks, detecting and measuring the network abuse are increasingly important. Network intrusion detection is the most frequently deployed approach. Detection frequently relies on only signature matching methods, and therefore suffers from lower accuracy and higher false alarm rates. This investigation presents a data-mining model (PAID) that constructs a packet header anomaly detection system with a Bayesian approach. The model accurately and automatically detects new malicious network attempts. On the DARPA evaluation data set, our method yields an accuracy of over 99.2% and a false positive rate of 0.03% for a DoS attack. Experimental results validate the feasibility of PAID to detect network intrusion.

Keywords: Data Mining, Intrusion Detection, Network Security, Machine Learning.

1 Introduction

Intrusion detection (ID) techniques are fundamental components of a security infrastructure adopted to detect and stop intruders. ID techniques are conventionally categorized as either misuse detection or anomaly detection. Misuse detection (also called signature-based detection) strives to detect well-known attacks by characterizing the rules. Misuse detection has a low false alarm rate but cannot identify any new attacks without pre-defined rules. In anomaly detection, each user has a profile within the system. Any deviation from the user profiles is regarded as an anomaly attack. Anomaly detection can detect new attacks but suffer from more false alarms than misuse detections.

Data mining methods extract rules from large datasets and use these rules to identify new instances in similar data. This investigation attempts to identify attacks by examining packet headers but not their contents. An anomaly detection model that learns the rules from data packet header fields at the network link is presented. The detection model is created by a data mining algorithm trained over a given set of training data. The proposed model does not examine application-layer protocols like HTTP or SMTP, and hence does not directly detect application-layer attacks, rather may detect attempts hidden in application layer connections. The proposed model can recognize not only well-known attacks but also new attacks.

This investigation presents a novel anomaly detection model (PAID) using packet header analysis. We also propose a mechanism to transform and aggregate continuous

features into buckets. The proposed mechanism is then applied with naive Bayes to perform intrusion detection.

This investigation concentrates mainly on network based intrusion detection, which monitors traffic at source and destination hosts. A network based intrusion detection system is an independent platform that is able to not only identify intrusions by examining network traffic but also monitor multiple hosts. The network-based system detects denial of service (DoS), probes, and remote-to-local (R2L) attacks. Host-based intrusion detection comprises an agent on a host that identifies intrusions by analyzing system calls, logs, file system modifications, and other host activities. The host-based system detects user-to-root (U2R) and R2L attacks.

To evaluate the performance of PAID, this work presents a comprehensive evaluation based on a large set of intrusion training and test data provided by DARPA intrusion detection evaluation dataset, which is a conventionally adopted dataset. Simulation results reveal that the proposed model is very effective, achieving over 99.2% accuracy with 0.03% false positive rate for DoS attacks.

The rest of this paper is organized as follows. Section 2 discusses related work in anomaly detection. Section 3 then presents an approach that adopts Bayesian methods to train the classifiers using packet headers. Section 4 presents the experimental results. Conclusions are finally drawn in Section 5.

2 Related Work

Network intrusion detection is typically signature based (i.e., misuse detection). Rules to identify any connection addressed to a nonexistent host or service are easy to write in SNORT [10]. However, this approach is vulnerable to novel attacks. In addition, it is also difficult to keep the rules or patterns up to date. An alternative approach is the anomaly detection, which classifies normal and suspicious traffics.

Various techniques have been proposed for modeling anomalous detection. Anomaly detection systems such as NIDES [6], SPADE [7], PHAD [11], ALAD [12] and SVM [15] model network traffic and generate an alarm when a deviation from the normal model is detected. These systems differ from one to another in the features extracted from available audit data and the particular algorithms they apply to derive the normal models. Most features are obtained from the packet headers. SPADE, ALAD and NIDES model the distribution of the source and destination IP addresses, port numbers, and the TCP connection state. PHAD adopts 34 attributes, which is much more than the other approaches. These attributes are obtained from the packet headers of the Ethernet, IP, TCP, UDP and ICMP packets. Lakhina et al. [2] proposed to use entropy as a measure of distributions of packet features to identify and classify anomaly network traffic volumes. Cardenas et al. [1] proposed a framework for IDS evaluation by viewing it as a multi-criteria optimization problem.

PHAD (Packet Header Anomaly Detector) [11] monitors the headers of Ethernet, IP, and the transport layer without any preconceptions. PHAD constructs profiles for 34 different fields from these headers by looking at attack-free traffic and then clusters them. A new value that does not fit into any of the clusters is considered as a new cluster, and the closest two clusters are merged.

Kruegel et al. describe a service specific intrusion detection system [3], which combines the type, length, and payload distribution of the request as the features to calculate an anomaly score. The 256 ASCII characters are grouped into six segments: 0, 1–3, 4–6, 7–11, 12–15 and 16–255. A uniform distribution model of these 6 segments is then computed for all requests to each service.

Bayes network is one of the most widely adopted graphical models for presenting uncertain data [4]. It is a directed acyclic graph (DAG) in which vertices represent events, and edges denote the relations between events. The numerical component quantifies the different links in the DAG according to distribution of the conditional probability of each node in the context of its parents. Bayes networks have been widely used to create models for anomaly intrusion detection. Puttini et al. [14] presented a behavior model using Bayes to obtain the model parameters. Goldman [13] proposed a model that simulates an intelligent attacker using Bayes networks to generate a plan of goal-directed actions.

3 Packet Analysis for Anomaly Intrusion Detection (Paid)

3.1 Packet Header Analysis

Network packet headers are streams of bytes. Network packet data contain parameters with values ranging among very large sets. A difficulty in anomaly detection is the choice of features.

In this paper, we propose a data mining system PAID to detect the network intrusions. Fig. 1 illustrates the process of analyzing network traffic using PAID. The data preprocessor in both training and testing procedures only filters packet headers information for further analysis. The following five attributes of packet header in PAID is monitored: duration, protocol, service, packet length, and connection flag. Duration is the elapsed time of the same connection. Protocol is network protocol, such as TCP, UDP, etc. Service is the port number on the destination, such as HTTP port number 80, telnet port number 23, etc. Packet length is total packet length. Connection flag is the flag of packets, such as normal or error. These attributes are obtained easily from real network traffic.

In the training procedure, when data fields are extracted, the system inspects every value to determine whether it is continuous. The continuous value is defined by data type. For instance, the packet length is an integer with the range from 0 to 4294967295, where PAID treats it as continuous value. The protocol is also an integer with the range from 0 to 300, and PAID treats it as discrete value. PAID will predefine types of features. For example, duration is continuous, protocol is discrete, service is discrete, packet length is continuous, and connection flag is discrete. The transformation procedure is performed if the value is continuous. The system subsequently attempts to place the transformed value into the pre-arranged bucket array that is managed dynamically. If no appropriate bucket is available for a new value, or the bucket array is full, then the system attempts to perform the bucket data aggregation to release additional bucket for later application. The feature extraction procedure is performed repeatedly until all training data are inserted into buckets. Training procedure will produce trained buckets after the entire training data are processed.

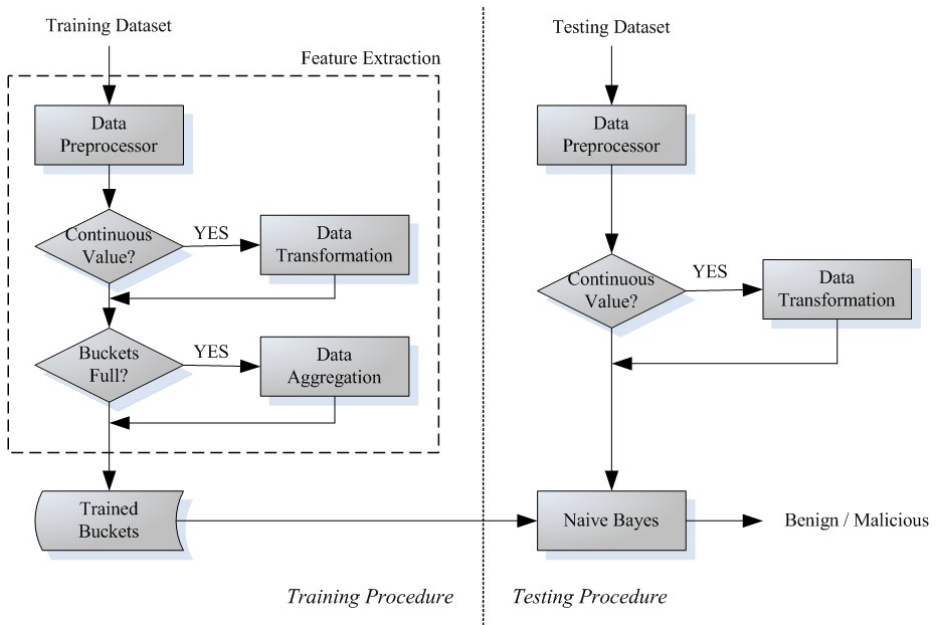


Fig. 1. Architecture of packet header analysis

In the testing procedure, the arriving data are filtered and transformed as described in the training procedure. The naive Bayes classifier is applied following the data transformation.

Bucket array is an input as a naive Bayes parameter, and the system computes the probability of the incoming packet headers. The output of the classifier is the highest probability class for a given set of packets. Since two classes, benign and malicious, are adopted, the output of our system labels test data as benign or malicious. The labeled test data can be inserted into trained buckets, and then the PAID can perform dynamic analysis of real-time network.

3.2 Data Transformation and Aggregation

To identify network attacks, information needs to be collected and aggregated from a large number of network packets. Since the attributes in network packets vary, this operation must be more sensitive to received packets. PAID provides methods for such transformation and aggregation with the details described as follows. The fields in the packet header are integer values with lengths from one to four bytes. The values of some fields could range from 0 to 4294967295, and storing every value of every packet needs much memory. Moreover, holding each continuous value of the testing dataset may result in difficulties when receiving new values not observed in the training dataset. To avoid the problems with continuous data, these continuous values have to be transformed into several data buckets. The number of data buckets is limited and predefined.

For continuous data, each value is placed into an individual bucket if the limit of buckets is not reached. If the limit is reached, we perform the aggregation procedure. First, the aggregation procedure separates all buckets into several groups according to the sign of volume of bucket. The volume of bucket is derived from subtract malicious from benign. Afterward, the procedure calculates the square measure of each group. We also derive the maximum square measure of bucket groups, `max_group`. Fig. 2(a) shows the first step of the aggregation procedure. After the first step is performed, the `max_group` will be the gray area in Fig. 2(a).

In some cases, the group size may be small but the volume of the bucket is very high. The bucket increases if these buckets in the group are combined. The group will become one bucket, and the `max_group` will always point to that big bucket. Finally, the system will stop. To prevent this drawback, if the bucket number in sign group is one (less than two), aggregation procedure will try to find the maximum bucket number in one group (longest group) and change `max_group` to this longest group. Fig. 2(b) shows that there are only a few buckets between point A and point B, but this group of buckets has very high volume. The second step will find out maximum size (longest group) of groups. For instance, in Fig. 2(b), the longest group is between point B and point C.

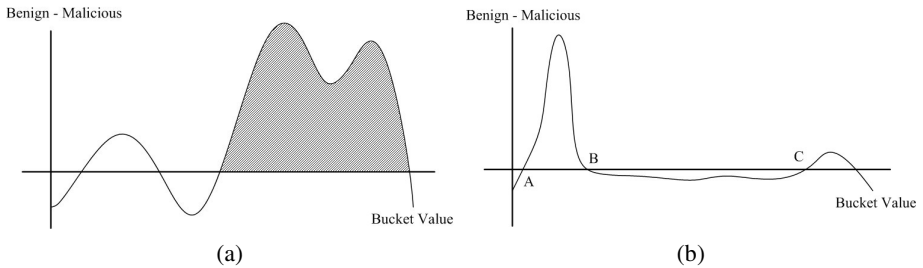


Fig. 2. (a) Aggregation procedure step one (b) Aggregation procedure step two

After `max_group` from step one or step two is obtained, the aggregation procedure tries to find out two buckets and combine them. The final step of aggregation procedure finds the maximum volume of two joint buckets. These two joint buckets will be combined, and a new bucket with the arriving value is appended. Fig. 3(a) shows the final step of aggregation procedure. The gray area in Fig. 2(a) is selected in first step. In the gray area of Fig. 2(a), the maximum volume of two joint buckets is left peak of the area. After final step is performed, the combined buckets are the gray area in Fig. 3(a), and we append a new bucket with the coming value.

For the discrete data, each value obtains a separate bucket, and transformation is thereby merely a mapping. When new value comes but does not match any existing bucket, PAID allocates a new bucket for the new value. If the number of discrete values exceeds the limit of buckets, we perform the above aggregation procedure. The model treats the discrete data as continuous data if the number of discrete values is larger than the limit of buckets.

Although transforming data into buckets causes some loss of information for continuous values, it increases the practicality of the model. This approach saves not only the memory but also the execution time. The training procedure with the whole DARPA dataset needs only 30 seconds with Intel Core 2 Duo 1.8GHz CPU in our implementation.

The PAID system may also avoid problems from predicting new values not existing in the testing dataset. When new values come in the data transformation of testing procedure, the data transformation finds the proper value according to the slope. Consider Fig. 3(b) as an example. It shows that there are no existing training data set between point A and point B. The data transformation finds proper value according to the slop around point A and B.

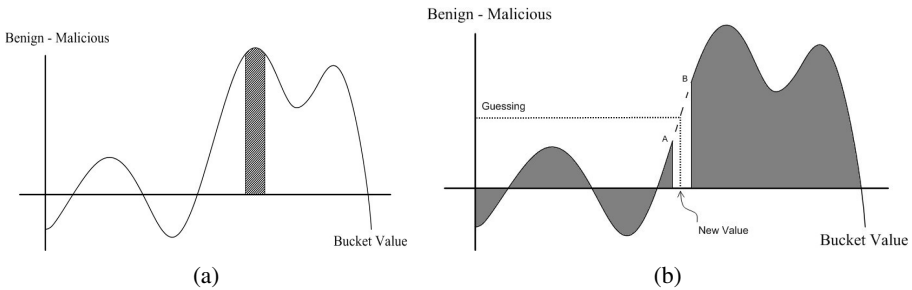


Fig. 3. (a) Final step of aggregation procedure (b) Guessing of brand new value

4 Evaluation

The data set used in our experiment originates from MIT Lincoln Laboratory, and has been developed for IDS evaluations by DARPA [9]. The network was operated like a real environment, being blasted with multiple attacks. In each connection, various quantitative and qualitative features were extracted and form a new data set. The processed data are available known as KDD cup 1999 dataset [8].

This experiment focuses on Network and DoS datasets. Fig. 4 shows the ROC (Receiver Operating Characteristic) curve of the relationship between the false positive and detection rates. In each of these ROC plots, the x-axis represents the false positive rate, derived as the percentage of normal connections classified as intrusions, and the y-axis denotes the detection rate, obtained as the percentage of intrusions detected. A data point in the upper left corner corresponds to the optimal performance, namely a low false positive rate with high detection rate. Fig. 4 plots the curves with the different bucket sizes as mentioned in Subsection 3.2.

The first experiment involved the network dataset containing DoS and Probing attacks. Fig. 4(a) reveals that we achieve over 94.8% accuracy with 0.08% false positive rate for DoS attacks with a bucket size of 40. Fig. 4(a) plots five lines, corresponding to bucket sizes 20, 30, 40, 50 and 60. The figure demonstrates that the detection model performs best with a bucket size of 40. The detection rate falls rapidly with decreasing bucket size. If we increase the buck size, the detection rate will also decrease.

Fig. 4(b) shows that PAID performed well in the DoS attack dataset. We achieve over 99.2% accuracy with a 0.03% false positive rate for DoS attacks. Additionally, a 99.35% accuracy resulted in a 0.08% false positive rate. Again, bucket size 40 had the best performance.

Fig. 4 indicates that the best bucket size for this experiment is approximately 40. Reducing the bucket size causes the aggregation procedure to be invoked frequently. Aggregating very many buckets causes significant loss of information in the features and lowers the detection rate while maintaining the same false positive rate. In contrast, raising the bucket size cause continuous values to be divided into more buckets, thus possibly separating connections belonging to the same class into different buckets. Therefore, it decreases the accuracy with the same false positive rate.

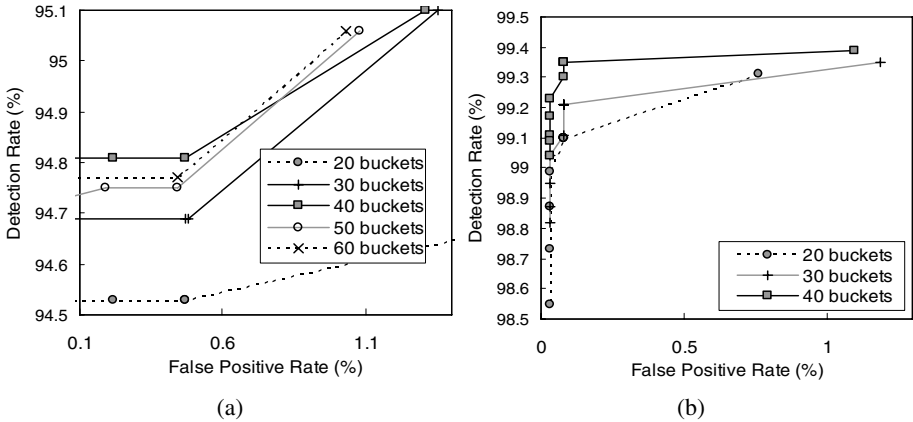


Fig. 4. ROC curves for (a) network dataset (b) DoS dataset

The experimental results are as expected because PAID employs statistical detection. The experimental results indicate that the R2L and U2R attacks do not have frequent patterns, unlike most of the DoS and Probing attacks. This is because the DoS and Probing attacks involve many connections to same hosts in a very short time, whereas the R2L and Probing attacks are potentially enveloped in the data portions of the packets. Therefore, each attack generally involves only a single connection.

5 Conclusion

In this paper, we present an anomaly detection model using packet header analysis. The proposed model, PAID, transforms and aggregates the continuous packet header fields, and applies them to a naive Bayes classifier. A series of evaluation tests is also performed on our approach using the KDD dataset, and the experimental results reveal that this model is effective at detecting network attacks. The KDD dataset achieves an accuracy of over 99.2% and a false positive rate of 0.03% for DoS attacks, and over 94.7% with 0.03% false positive rate for network attacks. Furthermore, PAID also achieves balanced performance for popular patterns and an averagely shorter detection

time for every attack, which can meet the time requirements for on-line intrusion detection. Compared with the conventional signature-based approach, the proposed statistical-based model of the naive Bayes classifier has a much higher accuracy and effectiveness than other tested schemes, particularly for DoS attacks. Since PAID only inspects packet headers only, it has a very low intrusion detection cost in large networks. Our model is an efficient and reliable mechanism for intrusion detection.

References

1. Cardenas, A., Baras, J.S.: A Framework for the Evaluation of Intrusion Detection Systems. In: IEEE Symposium on Security and Privacy (May 2006)
2. Lakhina, A., Crovella, M., Diot, C.: Mining anomalies using traffic feature distributions. In: SIGCOMM (2005)
3. Kruegel, C., Toth, T., Kirda, E.: Service Specific Anomaly Detection for Network Intrusion Detection. In: Symposium on Applied Computing (SAC), Span (March 2002)
4. Jensen, F.V.: Introduction to Bayesian networks. UCL Press (1996)
5. Gu, G., Fogla, P., Dagon, D., Lee, W., Skoric, B.: Towards an Information-Theoretic Framework for Analyzing Intrusion Detection Systems. In: Proc. European Symposium Research Computer Security (September 2006)
6. Javits, H.S., Valdes, A.: The NIDES statistical component: Description and justification. Technical report, SRI International, Computer Science Laboratory (1993)
7. Hoagland, J.: SPADE, Silicon Defense (2000), <http://www.silicondefense.com/software/spice>
8. KDD99 cup dataset (2006), <http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>
9. Massachusetts Institute of Technology Lincoln Laboratory, 1998 darpa intrusion detection evaluation dataset overview (2005), <http://www.ll.mit.edu/IST/ideval/>
10. Roesch, M.: Snort - lightweight intrusion detection for networks (2007), <http://www.snort.org>
11. Mahoney, M., Chan, P.K.: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In: Proc. ACM SIGKDD (2002)
12. Mahoney, M.: Network Traffic Anomaly Detection Based on Packet Bytes. In: Proc. ACM-SAC (2003)
13. Goldman, R.: A Stochastic Model for Intrusions. In: Symposium on Recent Advances in Intrusion Detection (RAID) (2002)
14. Puttini, R., Marrakchi, Z., Me, L.: Bayesian Classification Model for Real-Time Intrusion Detection. In: 22th International Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (2002)
15. Mukkamala, S., Janoski, G., Sung, A.H.: Intrusion Detection Using Neural Networks and Support Vector Machines. In: Proc. IEEE Int'l Joint Conf. on Neural Networks (2002)

A Comparison of Different Off-Centered Entropies to Deal with Class Imbalance for Decision Trees

Philippe Lenca¹, Stéphane Lallich²,
Thanh-Nghi Do³, and Nguyen-Khang Pham⁴

¹ Institut TELECOM, TELECOM Bretagne, Lab-STICC, Brest, France
philippe.lenca@telecom-bretagne.eu

² Université Lyon, Laboratoire ERIC, Lyon 2, Lyon, France
stephane.lallich@univ-lyon2.fr

³ INRIA Futurs/LRI, Université de Paris-Sud, Orsay, France
dtngchi@lri.fr

⁴ IRISA, Rennes, France
pnguyenk@irisa.fr

Abstract. In data mining, large differences in prior class probabilities known as the class imbalance problem have been reported to hinder the performance of classifiers such as decision trees. Dealing with imbalanced and cost-sensitive data has been recognized as one of the 10 most challenging problems in data mining research. In decision trees learning, many measures are based on the concept of Shannon's entropy. A major characteristic of the entropies is that they take their maximal value when the distribution of the modalities of the class variable is uniform. To deal with the class imbalance problem, we proposed an off-centered entropy which takes its maximum value for a distribution fixed by the user. This distribution can be the a priori distribution of the class variable modalities or a distribution taking into account the costs of misclassification. Others authors have proposed an asymmetric entropy. In this paper we present the concepts of the three entropies and compare their effectiveness on 20 imbalanced data sets. All our experiments are founded on the C4.5 decision trees algorithm, in which only the function of entropy is modified. The results are promising and show the interest of off-centered entropies to deal with the problem of class imbalance.

Keywords: Decision trees, Shannon entropy, Off-centered entropies, Imbalance class.

1 Class Imbalance Problem

In supervised learning, the data set is said to be imbalanced if the class prior probabilities are highly unequal. In the case of two-class problems, the larger class is called the majority class and the smaller the minority class. Real-life two-class problems have often minority class prior under 0.10 (e.g. fraud detection, medical diagnostic or credit scoring). In such a case the performances of data mining algorithms are lowered, especially the error rate corresponding to the

minority class, even though the minority class corresponds to positive cases and the cost of misclassifying the positive examples is higher than the cost of misclassifying the negative examples. This problem gave rise to many papers, from which one can cite papers from [1], [2] and [3]. Dealing with imbalanced and cost-sensitive data has been recognized as one of the 10 most challenging problems in data mining [4]. As summarized by the review papers of [5], [6] and [7] or by the very comprehensive papers of [8] and [9], solutions to the class imbalance problems were proposed both at the data and algorithmic level.

At the data level, these solutions change the class distribution. They include different forms of re-sampling, such that over-sampling [3] [10] or under-sampling [11], on a random or a directed way. A comparative study using C4.5 [12] decision tree show that under-sampling beat over-sampling [13]. At the algorithmic level, a first solution is to re-balance the error rate by weighting each type of error with the corresponding cost [14]. A study of the consistency of re-balancing costs, for misclassification costs and class imbalance, is presented in [15]. For a comparison of a cost sensitive approach and a sampling approach one can see for example [16]. In decision trees learning, other algorithmic solutions consist in adjusting the probabilistic estimates at the tree leaf or adjusting the decision thresholds. [17] propose to use a criterion of minimal cost, while [18] explore efficient pre-pruning strategies for the cost-sensitive decision tree algorithm to avoid overfitting. At both levels, [19] studied three issues (quality of probabilistic estimates, pruning, and effect of preprocessing the imbalanced data set), usually considered separately, concerning C4.5 decision trees and imbalanced data sets.

Our contribution belongs to the second category. We propose to replace the entropy used in tree induction algorithms by an off-centered entropy. That is to say that we work at the split level of decision trees learning taking into account an entropy criterion. The rest of the paper is organized as follows. In Section 2, we first review splitting criteria based on Shannon's entropy. We first recall basic considerations on Shannon's entropy and then briefly present our off-centered entropy and the asymmetric entropy. Then, we compare the entropies' performance on 20 imbalanced data sets in Section 3. Finally, Section 4 draws conclusions and suggests future work.

2 From Shannon's Entropy to Non-centered Entropies

In this section we first recall basic considerations on Shannon's entropy and then present the two families of non-centered entropies. For both of them we mainly present the boolean case and mention the results in the general case. Experiments presented in Section 3 are done in the boolean case.

2.1 Usual Measures Based on Shannon's Entropy

In supervised learning of induction tree on categorical variables, many learning algorithms use predictive association measures based on the entropy proposed by Shannon [20]. Let us consider a class variable Y having q modalities,

$p = (p_1, \dots, p_q)$ be the vector of frequencies of Y , and a categorial predictor X having k modalities. The joint relative frequency of the couple (x_i, y_j) is denoted $p_{ij}, i = 1, \dots, k; j = 1, \dots, q$. What is more, we denote by $h(Y) = -\sum_{j=1}^q p_{.j} \log_2 p_{.j}$ the a priori Shannon's entropy of Y and by $h(Y/X) = E(h(Y/X = x_i))$ the conditional expectation of the entropy of Y with respect to X .

Shannon's entropy, is a real positive function of $p = (p_1, \dots, p_q)$ to $[0..1]$, verifying notably interesting properties for machine learning purposes:

1. **Invariance by permutation of modalities:** $h(p)$ does not change when the modalities of Y are permuted;
2. **Maximality:** the value of $h(p)$ reaches its maximum $\log_2(q)$ when the distribution of Y is uniform, i.e. each modality of Y has a frequency of $1/q$;
3. **Minimality:** the value of $h(p)$ reaches its minimum 0 when the distribution of Y is sure, centered on one modality of Y , the others being of null frequency;
4. **Strict concavity:** the entropy $h(p)$ is a strictly concave function.

Amongst the measures based on Shannon's entropy, particularly studied in by [21] and [22], we especially wish to point out:

- the entropic gain [23], which values $h(Y) - h(Y/X)$;
- the u coefficient [24] is the relative gain of Shannon's entropy i.e. the entropic gain normalized on the a priori entropy of Y , and values $\frac{h(Y) - h(Y/X)}{h(Y)}$;
- the gain-ratio [12] which relates the entropic gain of X to the entropy of X , rather than to the a priori entropy of Y in order to discard the predictors having many modalities. It values $\frac{h(Y) - h(Y/X)}{h(X)}$;
- the Kvalseth coefficient [25], which normalizes the entropic gain by the mean of the entropies of X and Y . It then values $\frac{2(h(Y) - h(Y/X))}{h(X) + h(Y)}$.

The peculiarity of these coefficients is that Shannon's entropy of a distribution reaches its maximum when this distribution is uniform. Even though it is the entropic gain with respect to the a priori entropy of Y which is used in the numerator part of the previously mentioned coefficients, the entropies of Y and $Y/X = x_i$ used in this gain are evaluated on a scale for which the reference value (maximal entropy) corresponds to the uniform distribution of classes. The behavior of Shannon's entropy is illustrated in Fig. 1 in the boolean case.

It would seem more logical to evaluate directly the entropic gain through the use of a scale for which the reference value would correspond to the a priori distribution of classes. The above-mentioned characteristic of the coefficients based on the entropy is particularly questionable when the classes to be learned are highly imbalanced in the data, or when the classification costs differ largely.

2.2 Off-Centered Entropy

The construction of an off-centered entropy principle is sketched out in the case of a boolean class variable in [26] and [27]. In these previous works we proposed

a parameterized version of several statistical measures assessing the interest of association rules and constructed an off-centered entropy.

Let us consider a class variable Y made of $q = 2$ modalities. The frequencies distribution of Y for the values 0 and 1 is noted $(1 - p, p)$. We wish to define an off-centered entropy associated with $(1 - p, p)$, noted $\eta_\theta(p)$, which is maximal when $p = \theta$, θ being fixed by the user and not necessarily equal to 0.5 (in the case of a uniform distribution). In order to define the off-centered entropy, following the proposition described in [26], we propose that the $(1 - p, p)$ distribution should be transformed into a $(1 - \pi, \pi)$ distribution such that: π increases from 0 to 1/2 when p increases from 0 to θ , and π increases from 1/2 to 1 when p increases from θ to 1. By looking for an expression of π as $\pi = \frac{p-b}{a}$, on both intervals $0 \leq p \leq \theta$ and $\theta \leq p \leq 1$, we obtain: $\pi = \frac{p}{2\theta}$ if $0 \leq p \leq \theta$, $\pi = \frac{p+1-2\theta}{2(1-\theta)}$ if $\theta \leq p \leq 1$.

To be precise, the thus transformed frequencies should be denoted as $1 - \pi_\theta$ and π_θ . We will simply use $1 - \pi$ and π for clarity reasons. They do correspond to frequencies, since $0 \leq \pi \leq 1$. The off-centered entropy $\eta_\theta(p)$ is then defined as the entropy of $(1 - \pi, \pi)$: $\eta_\theta(p) = -\pi \log_2 \pi - (1 - \pi) \log_2(1 - \pi)$.

With respect to the distribution $(1 - p, p)$, clearly $\eta_\theta(p)$ is not an entropy strictly speaking. Its properties must be studied considering the fact that $\eta_\theta(p)$ is the entropy of the transformed distribution $(1 - \pi, \pi)$, $\eta_\theta(p) = h(\pi)$. The behavior of this entropy is illustrated in Fig. 1 for $\theta = 0.2$.

The off-centered entropy preserves various properties of the entropy, among those studied in particular by [28] in a data mining context. Those properties are easy to prove since $\eta_\theta(p)$ is defined as an entropy on π and thus possess such characteristics. It can be noticed that $\eta_\theta(p)$ is maximal for $p = \theta$ i.e. for $\pi = 0.5$. Invariance by permutation of modalities property is of course voluntarily abandoned. Proofs are given in detail in [29].

Following a similar way as in the boolean case we then extended the definition of the off-centered entropy to the case of a variable Y having q modalities, $q > 2$ [29,30]. The off-centered entropy for a variable with $q > 2$ modalities is the defined by $\eta_\theta(p) = h(\pi^*)$ where: $\pi_j^* = \frac{\pi_j}{\sum_{j=1}^q \pi_j}$ (in order to satisfy the normalization property), $0 \leq \pi_j \leq 1$, $\sum_{j=1}^q \pi_j = 1$ (π_j should be analogous to frequencies), $\pi_j = \frac{p_j}{q\theta_j}$ if $0 \leq p_j \leq \theta_j$, $\pi_j = \frac{q(p_j - \theta_j) + 1 - p_j}{q(1 - \theta_j)}$ if $\theta_j \leq p_j \leq 1$.

2.3 Off-Centered Generalized Entropies

Shannon’s entropy is not the only diversity or uncertainty function usable to build coefficients of predictive association. [31] already proposed a unified view of the three usual coefficients (the λ of Guttman, the u of Theil and the τ of Goodman and Kruskal), under the name of α coefficient. In a more general way we built the β coefficient, which are the analogue of the standardized gain when Shannon’s entropy is replaced by whichever concave function of uncertainty [32].

One of the particularities of the off-centering we here propose, compared to the approach proposed by [33] is that rather than defining a single off-centered entropy, it adapts to whichever kind of entropy. We thus propose a decentering

framework that one can apply to any measure of predictive association based on a gain of uncertainty [30].

2.4 Asymmetric Entropy

With an alternative goal, directly related to the construction of a predictive association measure, especially in the context of decision trees, [34] proposed a consistent and asymmetric entropy for a boolean class variable. This measure is asymmetric in the sense that one may choose the distribution for which it will reach its maximum; and consistent since it takes into account n , the size of the sampling scheme. They preserve the $\lim_{n \rightarrow \infty} h_{\theta}(p) = h(p)$ property but alter the $\lim_{p \rightarrow 0} h_{\theta}(p) = 0$ one in order to let the entropy reach its maximal value for a distribution chosen by the user ($\lim_{p \rightarrow 0} h_{\theta}(p) = 0$ maximal for $p = \theta$, where θ is fixed by the user). This implies revoking the $\lim_{p \rightarrow 1} h_{\theta}(p) = 0$ property. They propose: $h_{\theta}(p) = \frac{p(1-p)}{(1-2\theta)p + \theta^2}$. It can be noticed that for $\theta = 0.5$, this asymmetric entropy corresponds to the quadratic entropy of Gini. The behavior of this entropy is illustrated in Fig. 1 for $\theta = 0.2$.

In [33], the same authors extend their approach to the situation where the class variable has $q > 2$ modalities. What is more, since one may only make an estimation of the real distribution $(p_j)_{j=1, \dots, q}$ with an empirical distribution $(f_j)_{j=1, \dots, q}$, they wish that for same values of the empirical distribution, the value of the entropy should decrease as n rises (property 5, a new property called $\lim_{n \rightarrow \infty} h_{\theta}(p) = h(p)$). They thus are led to modify the third property ($\lim_{p \rightarrow 1} h_{\theta}(p) = 0$) in a new property 3' ($\lim_{p \rightarrow 1} h_{\theta}(p) = 0$): the entropy of a sure variable is only required to tend towards 0 as $n \rightarrow \infty$. In order to comply with these new properties, they suggest to estimate the theoretical frequencies p_j by their Laplace estimator, $\hat{p}_j = \frac{nf_j + 1}{n + q}$. They thus propose a consistent asymmetric entropy as: $h_{\theta}(p) = \sum_{j=1}^q \frac{\hat{p}_j(1-\hat{p}_j)}{(1-2\theta_j)\hat{p}_j + \theta_j^2}$.

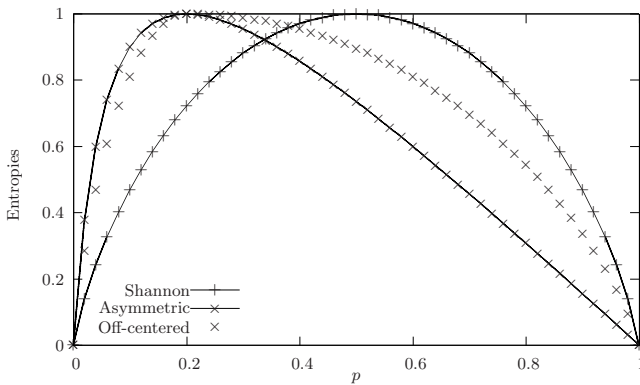


Fig. 1. Off-centered, asymmetric and Shannon's entropies

3 Experiments with More or Less Imbalanced Data Sets

In our experiments, we compare the behaviors of decision tree algorithms to classify imbalanced data sets using our proposed off-centered entropy OCE, the Shannons entropy SE and the asymmetric entropy AE. To achieve the evaluation we added OCE and AE to the decision tree algorithm C4.5 [12]. In these experiments, in each node the distribution for which OCE and AE are maximal is the a priori distribution of the class variable in the considered node.

The experimental setup used the 20 data sets described in Table 1 (column 1 indicates the data set name, the numbers of cases and of attributes), where the first twelve ones are from the UCI repository [35], the next six are from the Statlog repository [36], the following data set is from the DELVE repository (<http://www.cs.toronto.edu/~delve/>), while the last one is from [37].

In order to evaluate the performance of the considered entropies for classifying imbalanced data sets, we pre-processed multi-class (more than two classes, denoted by an asterisk) data sets as two-class problems. The columns 2 and 3 of Table 1 show how we convert multi-class to minority and majority classes. For example, with the OpticDigits data set, the digit 0 is mapped to the minority class (10%) and the remaining data are considered as the majority class (90%). For the 20-newsgroup collection, we pre-processed the data set by representing each document as a vector of words. With a feature selection method which uses mutual information, we get a binary data set of 500 dimensions (words).

The test protocols are presented in the column 4 of Table 1. Some data sets are already divided in training set (trn) and testing set (tst). If the training set and testing set are not available then we used cross-validation protocols to evaluate the performance, else k-fold cross validation is used. With a data set having less than 300 data points, the test protocol is leave-one-out cross-validation (loo). It involves using a single data point of the data set as the testing data and the remaining data points as the training data. This is repeated such that each data point in the data set is used once as testing data. With a data set having more than 300 data points, k-fold cross-validation is used to evaluate the performance. In k-fold cross-validation, the data set is partitioned into k folds. A single fold is retained as the validation set for testing, and the remaining k-1 folds are used as training data. The cross-validation process is then repeated k times. The k results are then averaged. The columns 5 to 9 of Table 1 present the results according to each entropy in terms of tree size, global error rate, error rate on the minority class and on the majority class (best results are in bold). The synthetic comparisons two by two are presented in Table 2.

For these first comparisons, we recall that the rule of prediction is the majority rule. The definition of another rule of prediction, better adapted to non-centered entropies, is one of the enhancements which we intend to accomplish.

We can conclude that the non-centered entropies, particularly the off-centered entropy, outperform the Shannon's entropy. These both entropies significantly improve the MinClass accuracy, without penalizing the MajClass accuracy, where MinClass (MajClass) accuracy is the proportion of true results in the minority (majority) class.

Table 1. Experiments on 20 imbalanced data sets

Base	Class Min.	Class Maj.	Valid.	Method	Tree size	Acc.	MinClass acc.	MajClass acc.
Opticdigits* 5620 64	10%(0)	90%(rest)	trn-tst	SE	27	99.39	96.63	99.69
				AE	21	99.83	100.00	99.81
				OCE	21	99.67	99.44	99.69
Tictactoe 958 9	35%(1)	65%(2)	10-fold	SE	69	93.33	87.50	96.49
				AE	89	93.65	89.52	95.82
				OCE	89	94.17	90.43	96.15
Wine* 178 13	27%(3)	73%(rest)	loo	SE	5	95.51	89.58	97.69
				AE	5	97.19	95.83	97.69
				OCE	5	97.19	95.83	97.69
Adult 48842 14	24%(1)	76%(2)	trn-tst	SE	123	86.25	60.85	94.11
				AE	171	85.67	60.02	93.61
				OCE	107	85.70	61.61	93.15
20-newsgrp* 20000 500	5%(1)	95%(rest)	3-fold	SE	9	98.59	73.31	99.95
				AE	13	98.65	74.49	99.95
				OCE	13	98.65	74.49	99.95
Breast Cancer 569 30	35%(M)	65%(B)	10-fold	SE	18	94.04	90.43	96.31
				AE	11	94.39	90.40	96.90
				OCE	13	93.33	90.45	95.20
Letters* 20000 16	4%(A)	96%(rest)	3-fold	SE	67	99.47	91.48	99.81
				AE	99	99.35	90.00	99.75
				OCE	105	99.44	92.59	99.73
Yeast* 1484 8	31%(CYT)	69%(rest)	10-fold	SE	52	71.76	47.95	82.66
				AE	65	71.82	48.82	82.26
				OCE	34	72.34	47.00	84.02
Connect-4* 67557 42	10%(draw)	90%(rest)	3-fold	SE	4141	83.25	57.02	91.72
				AE	4013	83.46	57.59	91.81
				OCE	4037	84.07	60.09	91.82
Glass* 214 9	33%(1)	67%(rest)	loo	SE	39	77.10	72.41	80.32
				AE	23	78.97	72.86	81.94
				OCE	21	86.45	78.57	90.28
Spambase 4601 57	40%(spam)	60%(rest)	10-fold	SE	250	93.00	90.94	94.31
				AE	269	93.22	91.52	94.28
				OCE	225	93.35	91.21	94.67
Ecoli* 336 7	15%(pp)	85%(rest)	10-fold	SE	11	94.55	74.68	98.19
				AE	14	94.24	76.50	97.43
				OCE	11	95.45	81.93	97.80
Pima 768 8	35%(1)	65%(2)	10-fold	SE	25	74.94	62.79	81.42
				AE	20	75.71	64.30	81.82
				OCE	20	75.19	63.15	81.62
German 1000 20	30%(1)	70%(2)	10-fold	SE	39	74.27	40.00	88.36
				AE	40	73.54	40.07	86.95
				OCE	43	74.48	44.40	86.45
Shuttle* 58000 9	20%(rest)	80%(1)	trn-tst	SE	27	99.99	99.93	100.00
				AE	19	99.80	99.90	100.00
				OCE	11	99.99	99.97	100.00
Segment* 2310 19	14%(1)	86%(rest)	10-fold	SE	7	99.22	95.78	99.79
				AE	18	99.31	95.91	99.85
				OCE	19	99.31	96.75	99.75
Satimage* 6435 36	24%(1)	90%(rest)	trn-tst	SE	99	97.35	94.36	98.25
				AE	103	98.00	96.10	98.57
				OCE	93	97.95	95.23	98.77
Vehicle* 846 18	24%(van)	76%(rest)	10-fold	SE	41	94.81	88.49	95.70
				AE	31	94.94	90.66	96.33
				OCE	32	95.18	88.95	97.10
Splice* 3190 60	25%(EI)	75%(rest)	10-fold	SE	72	96.37	92.74	97.62
				AE	62	96.40	93.23	97.50
				OCE	24	96.40	93.69	97.33
All-Aml 72 7129	35% (AML)	65%(ALL)	loo	SE	3	91.18	92.86	90.00
				AE	3	91.18	92.86	90.00
				OCE	3	91.18	92.86	90.00

Table 2. Comparison of Shannon entropy (SE), Off-centered entropy (OCE) and Asymmetric entropy (AE)

OCE vs. SE	Tree size	Acc.	MinClass acc.	MajClass acc.
Mean (OCE-SE)	-9.900	0.76%	1.94%	0.44%
Mean Std. dev. (OCE-SE)	6.318	0.47%	0.53%	0.53%
Student ratio	-1.567	1.621	3.673	0.830
p-value (Student)	Non sign.	Non sign.	0.0016	Non sign.
OCE wins	12	16	18	7
Exaequo	3	1	1	5
SE wins	5	3	1	8
p-value (sign test)	Non sign.	0.0044	0.0000	Non sign.
AE vs. SE	Tree size	Acc.	MinClass acc.	MajClass acc.
Mean (AE-SE)	-1.750	0.25%	1.04%	-0.01%
Mean Std. dev. (AE-SE)	7.500	0.14%	0.37%	0.14%
Student ratio	-0.233	1.746	2.808	-0.048
p-value (Student)	Non sign.	0.0970	0.0112	Non sign.
AE wins	8	14	15	8
Exaequo	2	1	1	4
SE wins	10	5	4	8
p-value (sign test)	Non sign.	Non sign.	0.0192	Non sign.
OCE vs. AE	Tree size	Acc.	MinClass acc.	MajClass acc.
Mean (OCE- AE)	-8.150	0.51%	0.90%	0.45%
Mean Std. dev. (OCE- AE)	4.563	0.38%	0.49%	0.44%
Student ratio	-1.786	1.330	1.846	1.014
p-value (Student)	0.0901	0.1991	0.0805	0.3234
OCE wins	8	11	11	8
Exaequo	6	5	3	4
AE wins	6	4	6	8
p-value (sign test)	Non sign.	Non sign.	Non sign.	Non sign.

Indeed, compared to Shannon’s entropy SE, the off-centered entropy OCE improves the MinClass accuracy 18 times out of 20, with 1 defeat and 1 equality, which corresponds to a p-value of 0.0000. The corresponding average gain in accuracy is close to 0.02 (p-value = 0.0016 according to a paired t-test). The accuracy of the MajClass is not significantly modified, but the global accuracy is improved 16 times out of 20, with 3 defeats and 1 equality (p-value = 0.0044), while the average corresponding gain is close to 0.008. Moreover, the trees provided by OCE have often a more reduced size, but this reduction is not significant.

The asymmetric entropy AE gives slightly less significant results when compared to Shannon’s entropy SE. It improves 15 times out of 20 the MinClass accuracy (p-value = 0.0192), with an average gain close to 0.01 (p-value = 0.0112). However, the improvement of the global accuracy is not significant: AE wins 14 times out of 20, with 1 equality and 5 defeats, while the increase of the global accuracy is only 0.002. In the same way, the performance for the MajClass accuracy is comparable (AE wins 8 times, SE wins 8 times, and 4 equalities). Furthermore, for the size of the tree, the performance is also comparable (AE wins 8 times, SE wins 10 times, and 2 equalities).

When comparing the two non-centered entropies OCE and AE, one can observe a slight but not significant superiority of the off-centered entropy OCE for each criterion. Particularly, a gain of 1 point on the MinClass error rate and 0.5 point on the total error rate must be noticed.

4 Conclusion and Future Works

In order to deal with imbalanced classes, we proposed an off-centered split function for learning induction trees. It has the characteristic to be maximum for the distribution a priori of the class in the node considered. We then compare, in the boolean case on 20 imbalanced data bases, the performances of our entropy with the entropy of Shannon and an asymmetric entropy. All our experiments are founded on C4.5 decision trees algorithm, in which only the entropy is modified. Compared to Shannon's entropy both non-centered entropies, significantly improve the minority class accuracy, without penalizing the majority one. Our off-centered entropy is slightly better than the asymmetric one, but this is not statistically significant. However one major advantage of our proposal is that it can be applied to any kind of entropy, for example to the quadratic entropy of Gini used in the CART algorithm [38]. We plan to improve the pruning scheme and the criterion to affect a class to a leaf. Indeed, these two criteria such as defined in C4.5, do not well support the recognition of the minority class. We then can hope for an improvement of our already good results. It could be also valuable to take into account a cost-sensitive matrix.

References

1. Japkowicz, N. (ed.): *Learning from Imbalanced Data Sets/AAAI* (2000)
2. Chawla, N., Japkowicz, N., Kolcz, A. (eds.): *Learning from Imbalanced Data Sets/ICML* (2003)
3. Chawla, N., Japkowicz, N., Kolcz, A. (eds.): *Special Issue on Class Imbalances. SIGKDD Explorations*, vol. 6 (2004)
4. Yang, Q., Wu, X.: 10 challenging problems in data mining research. *International Journal of Information Technology & Decision Making* 5(4), 597–604 (2006)
5. Japkowicz, N.: The class imbalance problem: Significance and strategies. In: *IC-AI*, pp. 111–117 (2000)
6. Japkowicz, N., Stephen, S.: The class imbalance problem: A systematic study. *Intelligent Data Analysis* 6(5), 429–450 (2002)
7. Visa, S., Ralescu, A.: Issues in mining imbalanced data sets - A review paper. In: *Midwest AICS Conf.*, pp. 67–73 (2005)
8. Weiss, G.M., Provost, F.: The effect of class distribution on classifier learning. *TR ML-TR 43*, Department of Computer Science, Rutgers University (2001)
9. Weiss, G.M., Provost, F.: Learning when training data are costly: The effect of class distribution on tree induction. *J. of Art. Int. Research* 19, 315–354 (2003)
10. Liu, A., Ghosh, J., Martin, C.: Generative oversampling for mining imbalanced datasets. In: *DMIN*, pp. 66–72 (2007)
11. Kubat, M., Matwin, S.: Addressing the curse of imbalanced data sets: One-sided sampling. In: *ICML*, pp. 179–186 (1997)
12. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1993)
13. Drummond, C., Holte, R.: C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In: *Learning from Imbalanced Data Sets/ICML* (2003)
14. Domingos, P.: Metacost: A general method for making classifiers cost sensitive. In: *KDD*, pp. 155–164 (1999)

15. Zhou, Z.H., Liu, X.Y.: On multi-class cost-sensitive learning. In: AAI, pp. 567–572 (2006)
16. Weiss, G.M., McCarthy, K., Zabar, B.: Cost-sensitive learning vs. sampling: Which is best for handling unbalanced classes with unequal error costs? In: DMIN, pp. 35–41 (2007)
17. Ling, C.X., Yang, Q., Wang, J., Zhang, S.: Decision trees with minimal costs. In: ICML (2004)
18. Du, J., Cai, Z., Ling, C.X.: Cost-sensitive decision trees with pre-pruning. In: Kobti, Z., Wu, D. (eds.) Canadian AI 2007. LNCS (LNAI), vol. 4509, pp. 171–179. Springer, Heidelberg (2007)
19. Chawla, N.: C4.5 and imbalanced datasets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: Learning from Imbalanced Data Sets/ICML (2003)
20. Shannon, C.E.: A mathematical theory of communication. *Bell System Technological Journal* (27), 379–423, 623–656 (1948)
21. Wehenkel, L.: On uncertainty measures used for decision tree induction. In: IPMU, pp. 413–418 (1996)
22. Loh, W.Y., Shih, Y.S.: Split selection methods for classification trees. *Statistica Sinica* 7, 815–840 (1997)
23. Quinlan, J.R.: Induction of decision trees. *Machine Learning* 1(1), 81–106 (1986)
24. Theil, H.: On the estimation of relationships involving qualitative variables. *American Journal of Sociology* (76), 103–154 (1970)
25. Kvalseth, T.O.: Entropy and correlation: some comments. *IEEE Trans. on Systems, Man and Cybernetics* 17(3), 517–519 (1987)
26. Lallich, S., Vaillant, B., Lenca, P.: Parametrised measures for the evaluation of association rule interestingness. In: ASMDA, pp. 220–229 (2005)
27. Lallich, S., Vaillant, B., Lenca, P.: A probabilistic framework towards the parameterization of association rule interestingness measures. *Methodology and Computing in Applied Probability* 9, 447–463 (2007)
28. Zighed, D.A., Rakotomalala, R.: *Graphes d’Induction – Apprentissage et Data Mining*. Hermes (2000)
29. Lallich, S., Vaillant, B., Lenca, P.: Construction d’une entropie décentrée pour l’apprentissage supervisé. In: QDC/EGC 2007, pp. 45–54 (2007)
30. Lallich, S., Lenca, P., Vaillant, B.: Construction of an off-centered entropy for supervised learning. In: ASMDA, p. 8 (2007)
31. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classifications, i. *JASA I*(49), 732–764 (1954)
32. Lallich, S.: *Mesure et validation en extraction des connaissances à partir des données*. In: Habilitation à Diriger des Recherches, Université Lyon 2, France (2002)
33. Zighed, D.A., Marcellin, S., Ritschard, G.: *Mesure d’entropie asymétrique et consistante*. In: EGC, pp. 81–86 (2007)
34. Marcellin, S., Zighed, D.A., Ritschard, G.: An asymmetric entropy measure for decision trees. In: IPMU, pp. 1292–1299 (2006)
35. Blake, C.L., Merz, C.J.: *UCI repository of machine learning databases* (1998)
36. Michie, D., Spiegelhalter, D.J., Taylor, C.C. (eds.): *Machine Learning, Neural and Statistical Classification*. Ellis Horwood (1994)
37. Jinyan, L., Huiqing, L.: *Kent ridge bio-medical data set repository*. Technical report (2002)
38. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: *Classification and Regression Trees*. Wadsworth International (1984)

FISViz: A Frequent Itemset Visualizer

Carson Kai-Sang Leung*, Pourang P. Irani, and Christopher L. Carmichael

The University of Manitoba, Winnipeg, MB, Canada
{kleung, irani, umcarmil}@cs.umanitoba.ca

Abstract. Since its introduction, frequent itemset mining has been the subject of numerous studies. However, most of them return frequent itemsets in the form of textual lists. The common cliché that “a picture is worth a thousand words” advocates that visual representation can enhance user understanding of the inherent relations in a collection of objects such as frequent itemsets. Many visualization systems have been developed to visualize raw data or mining results. However, most of these systems were not designed for visualizing frequent itemsets. In this paper, we propose a *frequent itemset visualizer (FISViz)*. FISViz provides many useful features so that users can effectively see and obtain implicit, previously unknown, and potentially useful information that is embedded in data of various real-life applications.

1 Introduction

Frequent itemset mining [1,10,11,12,13] plays an essential role in the mining of various patterns (e.g., association rules, correlation, sequences, episodes, maximal itemsets, closed itemsets) and is in demand for many real-life applications. Mined frequent itemsets can answer many questions (examples of which are shown in Fig. 1) that help users make important decisions. Hence, numerous frequent itemset mining algorithms have been proposed over the last decade. However, most of them return a collection of frequent itemsets in *textual form* (e.g., a very long unsorted list of frequent itemsets). As a result, users may not easily see the useful information that is embedded in the data.

To assist users in gaining insight into massive amounts of data or information, researchers have considered many visualization techniques [7,16]. Visualization systems like Spotfire [2], VisDB [8] and Polaris [17] have been developed for visualizing data but *not* the mining results. For systems that visualize the mining results, the focus has been mainly on results such as clusters [9,15], decision trees [3], social networks [4] or association rules [5,6]—instead of frequent itemsets. Showing a collection of frequent itemsets in *graphical form* can help users understand the nature of the information and show the relations embedded in the data.

Recently, researchers have shown interests in visualizing frequent itemsets. For instance, Munzner et al. [14] presented a visualization system called PowerSetViewer (PSV), which provides users with guaranteed visibility of frequent itemsets. However, PSV does not show the relationship between related itemsets (e.g., not easy to know that itemsets {apples, bananas} and {apples, bananas, cherries} are related—the former is a

* Corresponding author.

- Q1.** Store managers may want to find answers to the following questions:
- (a) What kinds of fruits (e.g., {apples, bananas}) are frequently purchased by customers?
 - (b) How frequently items are purchased *individually* (e.g., 70% of customers purchased apples)?
 - (c) How frequently items are purchased *together* (e.g., only 30% of customers purchased apples and bananas together)?
 - (d) What items are frequently purchased together with cherries (e.g., {apples, bananas, cherries, dates})?
 - (e) Which itemset has the highest cardinality (e.g., a basket containing 30 different kinds of fruits)?
 - (f) Which is the most frequently purchased 3-itemset (e.g., {apples, bananas, cherries})?
- Q2.** University administrators may want to know which popular elective courses (e.g., {Astronomy 101, Biology 102, Chemistry 100}) are taken by students?
- Q3.** Internet providers may want to figure out what Webpages are frequently browsed by Internet users in a single session?
- Q4.** Bookstore owners may want to know which books are also bought by customers who bought a particular data mining book?
-

Fig. 1. Sample questions answered by frequent itemset mining

subset of the latter). Yang [18] also developed a system that can visualize frequent itemsets. However, his system was primarily designed to visualize association rules, and it does not scale very well in assisting users to immediately see certain patterns among a very large number of items/itemsets.

Hence, some natural questions to ask are: Can we design a system that explicitly shows relationships among frequent itemsets? Can we help users find satisfactory answers to important questions that could lead to critical business decisions?

To this end, we present a visualizer to enhance the data mining process of the user by providing answers to some important business questions. The **key contribution** of our work is a novel interactive system, called *FISViz*, for *visualizing frequent itemsets*. This visualizer provides users with clear and explicit depictions about frequent itemsets that are embedded in the data of interest. Hence, *FISViz* enables users—at a glance—to infer patterns and answers to many questions (e.g., Q1-Q4 in Fig. 1); it also provides interactive features for constrained and interactive mining. Moreover, with *FISViz*, users can efficiently find closed itemsets and can easily formulate association rules from the displayed frequent itemsets.

This paper is organized as follows. Next section describes related work. In Section 3, we introduce our *FISViz* and describe its design as well as features. Section 4 shows evaluation results. Then, we briefly discuss, in Section 5, the scalability of *FISViz* with respect to large datasets. Finally, conclusions are presented in Section 6.

2 Related Work

Yang [18] designed a system mainly to visualize association rules—but can also be used to visualize frequent itemsets—in a two-dimensional space consisting of many vertical axes. In the system, all domain items are sorted according to their frequencies and are

evenly distributed along each vertical axis. A k -itemset is then represented by a curve that extends from one vertical axis to another connecting k such axes. The thickness of the curve indicates the frequency (or support) of such an itemset. However, such a representation suffers from the following problems: (i) The use of thickness only shows *relative* (but not *exact*) frequency of itemsets. Comparing the thickness of curves is not easy. (ii) Since items are sorted and *evenly* distributed along the axes, users only know some items are more frequent than the others, but cannot get a sense of how these items are related to each other in terms of their exact frequencies (e.g., whether item a is twice as frequent as, or just slightly more frequent than, item b). (iii) Although Yang's system is able to show both association rules and frequent itemsets, his system does not provide users with many interactive features, which are necessary if a large graph containing many items to be displayed.

PowerSetViewer (PSV) [14] is designed specifically for displaying frequent itemsets in the context of the powerset universe. With PSV, frequent itemsets are first grouped together based on cardinality (each represented by a different background color) in a two-dimensional grid; itemsets of the same cardinality are then mapped into grid squares. When the number of k -itemsets exceeds the number of allocated grid squares, PSV maps several frequent itemsets into one square. A square is highlighted if it contains at least one frequent itemset. This provides users with *guaranteed visibility* of itemsets. While PSV is truly designed for visualizing frequent itemsets, it also suffers from the following problems: (i) As a highlighted grid square may contain many frequent itemsets, it is not easy to find out which one or more itemsets (among all the itemsets represented by such a square) are frequent. (ii) PSV does not tell the *exact* frequencies of frequent itemsets. (iii) It is difficult to grasp the relationships between two related itemsets (e.g., $\{a, b\}$ is a subset of $\{a, b, c, d\}$).

3 FIsViz: Our Proposed System for Visualizing Frequent Itemsets

In this section, we show basic representation and demonstrate features of our proposed *frequent itemset visualizer (FIsViz)*.

3.1 Basic Representation of FIsViz

FIsViz shows frequent k -itemsets in a two-dimensional space. The x -axis shows the n domain items, which are arranged in non-ascending frequency order (by default) on the x -axis. The y -axis, which can be in *logarithmic-scale* or *normal-scale*, shows the frequencies of itemsets. A connecting edge between two items suggests that the two items appear together in the dataset. In this way, a non-singleton itemset (e.g., $\{\text{apples, bananas, cherries}\}$) is represented by a polyline (series of consecutive edges) ended with a left-pointing triangle. Each singleton itemset (e.g., $\{\text{apples}\}$) is represented by a circle. See Fig. 2(a) for a snapshot of the basic representation of FIsViz. Observations (Fig. 3) of this snapshot reveal the following *properties* associated with this basic representation of FIsViz:

1. FIsViz provides users with a quick intuitive overview about the frequency of each individual domain item (indicated by a circle) with frequency clearly indicated by its y -position.

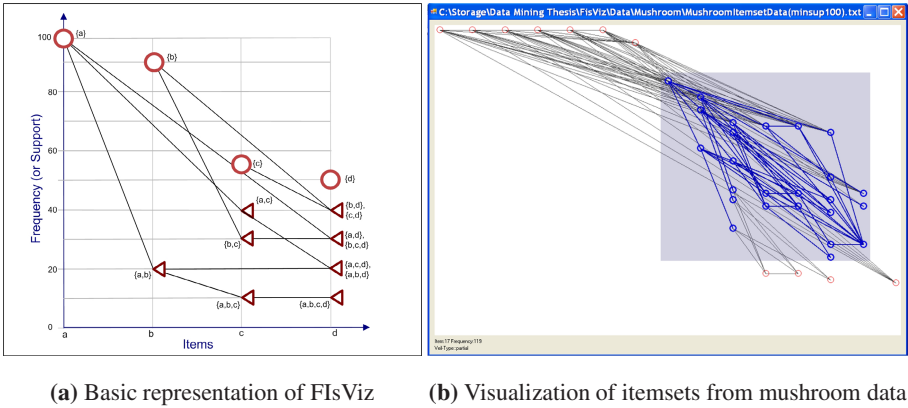


Fig. 2. Snapshots of our proposed FISViz

1. Items a and b frequently occur individually (with $sup(a)=100\%$ and $sup(b)=90\%$), but their combination $\{a, b\}$ does not occur frequently (with $sup(\{a, b\})=20\%$).
2. The leftmost item is a (which has the highest frequency) and the rightmost item is d (which has the lowest frequency). Moreover, $sup(a)=100\% \geq sup(b) \geq sup(c) \geq sup(d)=50\%$.
3. $sup(\{a, b\})=20\%$, $sup(\{a, b, c, d\})=10\%$ and $sup(\{c, d\})=40\%$. Knowing this information, users can easily obtain the support, confidence and lift of association rule $\{a, b\} \Rightarrow \{c, d\}$ using $sup(\{a, b, c, d\})$, $\frac{sup(\{a, b, c, d\})}{sup(\{a, b\})}$ and $\frac{sup(\{a, b, c, d\})}{sup(\{a, b\}) \times sup(\{c, d\})}$ respectively. Moreover, observing that $sup(\{a, b\}) = sup(\{a, b, d\})$, users can easily determine that $conf(\{a, b\} \Rightarrow \{d\})=100\%$.
4. When moving along the polyline representing $\{a, b, c, d\}$, itemset $\{a, b\}$ appears to the left of $\{a, b, c\}$ (as the former is a *prefix* of the latter). Similarly, $\{a, b, c, d\}$ appears to the right of $\{a, b, c\}$ (as the former is an *extension* of the latter). Moreover, $sup(\{a, b\})=20\% \geq sup(\{a, b, c\})=10\% \geq sup(\{a, b, c, d\})=10\%$.
5. All *subsets* of $\{a, c, d\}$ appear to the left and above $\{a, c, d\}$.
6. All *supersets* of $\{a, c\}$ appear to the right and below $\{a, c\}$.
7. $\{a, c\}$ is a *closed itemset*, but $\{b, c\}$ is *not* because $sup(\{b, c\}) = sup(\{b, c, d\})$.

Fig. 3. Observations on Fig. 2(a)

2. The most frequently occurring item (which with the highest frequency) appears on the left side and the least frequently occurring one appears on the right side.
3. Each k -itemset (where $k > 1$) is represented by a polyline, and its frequency is the frequency of the right-end item node of the polyline. The frequency is clearly indicated by the y -position of left-pointing triangle.
4. All prefixes of any k -itemset α appear on the left of α along the polyline that represents α , whereas all extensions of α appears on the right of α along such a polyline. Moreover, due to the Apriori property [11], it is guaranteed that the frequency of any prefix of $\alpha \geq$ the frequency of $\alpha \geq$ the frequency of any extension of α . When one moves along the polyline from right to left, the frequencies of prefixes of α are non-decreasing. Furthermore, users can see *how the frequency of α changes when*

(i) *truncating some items to form a prefix* or (ii) *appending some items to form an extension*.

5. All the nodes representing subsets of an itemset α appear to the left and above the node representing α . Knowing this property is useful because this reduces the search space (to only the left and above the node representing α) if one wants to search for all subsets of α .
6. Similarly, all the nodes representing supersets of an itemset α appear to the right and below the node representing α . Again, this property helps reduce the search space.
7. In addition to finding *frequent itemsets* (and their frequencies), users can also find *closed itemsets* (and their frequencies) effectively.

3.2 Features of FIsViz

Feature 1 (Query on frequency). With our FIsViz, users can easily find all *frequent items* and/or *frequent itemsets* (i.e., with frequencies exceeding the user-specified minimum frequency threshold *minsup*) by ignoring everything that lies below the “threshold line” $y=\text{minsup}$ (i.e., ignoring the lower portion of the graph). To a further extent, the representation of itemsets in FIsViz can lead to effective *interactive mining*. To elaborate, with FIsViz, users can easily see what (and how many) itemsets are above a certain frequency. Based on this information, users can freely adjust *minsup* (by moving the slider—which controls *minsup*—up and down along the y -axis) and find an appropriate value for *minsup*. See Fig. 2(b), which shows itemsets with frequencies $\geq \text{minsup}$. Moreover, FIsViz also provides two related features: (i) It allows users to interactively adjust *minsup* and automatically counts the number of itemsets that satisfy *minsup*. By doing so, users can easily find **top- N frequent itemsets**. (ii) It also allows users to pose a **range query on frequency** (by specifying both minimum and maximum frequency thresholds *minsup* and *maxsup*) and then shows all itemsets with frequencies falling within the range $[\text{minsup}, \text{maxsup}]$.

Feature 2 (Query on cardinality). In FIsViz, itemsets of different cardinalities are drawn in different color, and itemsets with higher cardinality are drawn over those with the lower cardinality. This helps users find *closed itemsets* and *maximal itemsets*. Moreover, FIsViz also allows users to pose a **range query on cardinality** so that only those frequent itemsets with cardinality k within the user-specified range $[k_{\min}, k_{\max}]$ are drawn.

Feature 3 (Query on itemsets). FIsViz also allows users to interactively select certain items of interest (e.g., promotional items in a store) and to pose the queries on itemsets. Examples of these queries include (i) “find all itemsets containing *some* of selected items”, (ii) “find all itemsets containing at least *all* of the selected items”, and (iii) “find all itemsets *not* containing any of the selected items”. See Fig. 2(b), in which selected itemsets are highlighted.

Feature 4 (Details-on-demand). Details-on-demand consists of techniques that provide more details whenever the user requests them. The key idea is that FIsViz gives users an overview of the entire dataset and then allows users to interactively select parts of the overview for which they request more details—by hovering the mouse over different parts of the display. Specifically, FIsViz supports details-on-demand in the

following ways: (i) When **the mouse hovers on an edge/polyline** connecting two nodes (say, representing items x and y), FISViz shows a list of itemsets containing both x and y . Selecting an itemset in the list instantly highlights the specific edge it is contained in, as well as both of its connecting nodes, so that users can see where the edge starts and ends. (ii) When the **mouse hovers over a node**, FISViz shows a list of all itemsets contained in all the edges starting or ending at this node. Selecting an itemset from the list instantly highlights the edge it is contained in. (iii) When the **mouse hovers over a pixel** in the display (even if it is not part of the graph), a small box appears showing the frequency and itemsets encoded by the mouse position. This is particularly useful when users need to see among the vast array of edges what a particular point in the display refers to.

Feature 5 (Formation of association rules). For many existing systems for visualizing association rules (which only shows the support and confidence of the rule $A \Rightarrow C$), it is not easy to obtain the frequencies of itemset A and of C . In contrast, our FISViz displays the information needed to infer and compute these rules. For instance, one can form a rule and then compute its *support* as well as *confidence* based on the frequencies of A and C . See Observation 3 in Fig. 3. Moreover, FISViz provides an additional benefit that users can compute other metrics such as *lift*.

Feature 6 (Ordering of domain items). By default, FISViz arranges the domain items (on the x -axis) in non-ascending frequency order. However, FISViz also provides users with an option to arrange items other orders. Having such an option is useful for *constrained mining*, in which users may want to arrange the items according to some constraints (e.g., put items of interest—say, promotional items—on the left and other items on the right of the screen). With this item ordering, the following property is preserved: *Frequencies of prefixes of the k -itemsets remain non-decreasing when moving from right to left.*

4 Evaluation Results

We conducted two sets of evaluation tests. In the first set, we tested functionality of our FISViz by showing how it can be applicable in various scenarios or real-life applications. In the second set, we tested performance of our FISViz.

4.1 Functionality Test

In the first set of evaluation tests, we compared our FISViz with existing systems like Yang's system [18] and PSV [14]. We considered many different real-life scenarios. For each scenario, we determined whether these systems can handle the scenarios. If so, we examined how these system display the mining results. The evaluation results show that our FISViz was effective in all these scenarios. A few samples of these scenarios are shown in Fig. 4.

4.2 Performance Test

In the performance test, we used (i) several IBM synthetic datasets [1] and (ii) some real-life databases (e.g., mushroom dataset) from UC Irvine Machine Learning

For **Q1(a)** in Fig. 1 frequently purchased fruits are itemsets with high frequency. With PSV, users may *not* be able to easily see the content of the itemsets because several itemsets may be mapped into a grid square. In contrast, our FIsViz shows all frequent itemsets by polylines, which are easily visible.

For **Q1(b)** and **Q1(c)**, Yang’s system shows frequencies of itemsets, but it does not give users the *exact* frequencies of itemsets because frequencies are represented by the thickness of curves. In PSV, the brightness of a grid square shows its density (i.e., the number of itemsets that were mapped into that square) but *not* its frequency. In contrast, users can easily obtain the frequencies of itemsets from our FIsViz.

For **Q1(d)**, PSV does *not* provide the linkage or relationship between related itemsets. In contrast, our FIsViz provides users with a feature of handling queries on itemsets containing one specific item (in this scenario, cherries).

For **Q1(e)**, PSV shows itemsets with highest cardinality on the bottom of the screen. Our FIsViz allows users to query on cardinality. Hence, itemsets with highest cardinality (i.e., poly-lines with the most number of nodes) can be displayed.

For **Q1(f)**, with FIsViz, users can first pose a query on cardinality to find only 3-itemsets, and then picks the itemset with the highest frequency.

Fig. 4. Sample scenarios and evaluation results for the functionality test

Depository. The results produced are consistent. Fig. 2(b) shows a screenshot of using the real-life mushroom dataset.

In the first experiment, we varied the size of databases. The results showed that the runtime (which includes CPU and I/Os) increased linearly with the number of transactions in the database.

In the second experiment, we varied the number of items in the domain. The results showed that the runtime increased when the number of domain items increased.

In the third experiment, we varied the user-defined frequency threshold. When the threshold increased, the number of itemsets that satisfy the threshold (i.e., itemsets to be displayed) decreased, which in turn leads to a decrease in runtime.

5 Discussion: Scalability of FIsViz

Recall that our FIsViz presents items on the x -axis. If each item is displayed by one pixel, then eventually the visualizer is limited by the number of items it can display within the user viewpoint. To overcome this limitation, we are developing the following approaches: (i) We apply *multi-resolution visualization*, with which we show the overall structure at one resolution and present details (upon the user request) at a different resolution. (ii) We span some of the displays beyond the viewpoint by carefully *embedding FIsViz with navigation facilities* (e.g., scrolling, panning) so that users can view items that are off-screen with minimum effort and without losing connectivity information from the lines in the display. (iii) We condense the large dataset by *creating taxonomies on domain items* based on their properties (e.g., item type) so that a large number of items can be coalesced onto a data point, which can then be opened for more details (or closed for fewer details) by users.

6 Conclusions

Most of frequent itemset mining studies return a collection of frequent itemsets in textual forms, which can be very long and difficult to comprehend. Since “a picture is worth a thousand words”, it is desirable to have visual systems. However, many existing visual systems were not designed to show frequent itemsets. To improve this situation, we proposed and developed a powerful *frequent itemset visualizer (FISViz)*, which provides users with explicit and easily-visible information among the frequent itemsets. Specifically, FISViz gives a quick intuitive overview of all the itemsets and their frequencies (e.g., visual clues show which individual items are most frequent and how the items or itemsets are distributed); it also provides in-depth details of interesting itemsets (e.g., itemsets of a certain frequency and/or cardinality) through human interaction like mouse hover. Evaluation results showed the effectiveness of FISViz in answering a board range of questions for real-life applications. These answers helps users in making appropriate business decisions.

Acknowledgement. This project is partially supported by NSERC (Canada) in the form of research grants.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proc. VLDB, pp. 487–499 (1994)
2. Ahlberg, C.: Spotfire: an information exploration environment. SIGMOD Record 25(4), 25–29 (1996)
3. Ankerst, M., Elsen, C., et al.: Visual classification: an interactive approach to decision tree construction. In: Proc. KDD, pp. 392–396 (1999)
4. Appan, P., Sundaram, H., Tseng, B.L.: Summarization and visualization of communication patterns in a large-scale social network. In: Ng, W.-K., Kitsuregawa, M., Li, J., Chang, K. (eds.) PAKDD 2006. LNCS (LNAI), vol. 3918, pp. 371–379. Springer, Heidelberg (2006)
5. Brunk, C., Kelly, J., Kohavi, R.: MineSet: an integrated system for data mining. In: Proc. KDD, pp. 135–138 (1997)
6. Han, J., Cercone, N.: AViz: A visualization system for discovering numeric association rules. In: Terano, T., Liu, H., Chen, A.L.P. (eds.) PAKDD 2000. LNCS (LNAI), vol. 1805, pp. 269–280. Springer, Heidelberg (2000)
7. Keim, D.A.: Information visualization and visual data mining. IEEE TVCG 8(1), 1–8 (2002)
8. Keim, D.A., Kriegel, H.-P.: Visualization techniques for mining large databases: a comparison. IEEE TKDE 8(6), 923–938 (1996)
9. Koren, Y., Harel, D.: A two-way visualization method for clustered data. In: Proc. KDD, pp. 589–594 (2003)
10. Lakshmanan, L.V.S., Leung, C.K.-S., Ng, R.T.: Efficient dynamic mining of constrained frequent sets. ACM TODS 28(4), 337–389 (2003)
11. Leung, C.K.-S., Khan, Q.I.: DSTree: A tree structure for the mining of frequent sets from data streams. In: Proc. IEEE ICDM, pp. 928–932 (2006)
12. Leung, C.K.-S., Khan, Q.I., Hoque, T.: CanTree: a tree structure for efficient incremental mining of frequent patterns. In: Proc. IEEE ICDM, pp. 274–281 (2005)

13. Leung, C.K.-S., Mateo, M.A.F., Brajczuk, D.A.: A tree-based approach for frequent pattern mining from uncertain data. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 653–661. Springer, Heidelberg (2008)
14. Munzner, T., Kong, Q., et al.: Visual mining of power sets with large alphabets. Technical report TR-2005-25, UBC, Canada (2005)
15. Pözlzbauer, G., Rauber, A., Dittenbach, M.: A vector field visualization technique for self-organizing maps. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 399–409. Springer, Heidelberg (2005)
16. Spence, R.: Information Visualization: Design for Interaction, 2nd edn. Prentice Hall, Harlow, UK (2007)
17. Stolte, C., Tang, D., Hanrahan, P.: Query, analysis, and visualization of hierarchically structured data using Polaris. In: Proc. KDD, pp. 112–122 (2002)
18. Yang, L.: Pruning and visualizing generalized association rules in parallel coordinates. IEEE TKDE 17(1), 60–70 (2005)

A Tree-Based Approach for Frequent Pattern Mining from Uncertain Data

Carson Kai-Sang Leung*, Mark Anthony F. Mateo, and Dale A. Brajczuk

The University of Manitoba, Winnipeg, MB, Canada
{kleung, mfmateo, umbrajcz}@cs.umanitoba.ca

Abstract. Many frequent pattern mining algorithms find patterns from traditional transaction databases, in which the content of each transaction—namely, items—is definitely known and precise. However, there are many real-life situations in which the content of transactions is uncertain. To deal with these situations, we propose a tree-based mining algorithm to efficiently find frequent patterns from uncertain data, where each item in the transactions is associated with an existential probability. Experimental results show the efficiency of our proposed algorithm.

1 Introduction

Over the past decade, there have been numerous studies [1,2,3,6,7,8,9,11,12,13,14,15] on mining frequent patterns from *precise* data such as databases of market basket transactions, web logs, and click streams. In these databases of precise data, users definitely know whether an item (or an event) is present in, or is absent from, a transaction in the databases. However, there are situations in which users are uncertain about the presence or absence of some items or events [4,5,10]. For example, a physician may highly suspect (but cannot guarantee) that a patient suffers from flu. The uncertainty of such suspicion can be expressed in terms of *existential probability*. So, in this uncertain database of patient records, each transaction t_i represents a visit to the physician's office. Each item within t_i represents a potential disease, and is associated with an existential probability expressing the likelihood of a patient having that disease in t_i . For instance, in t_i , the patient has an 80% likelihood of having the flu, and a 60% likelihood of having a cold regardless of having the flu or not. With this notion, each item in a transaction t_i in traditional databases containing precise data can be viewed as an item with a 100% likelihood of being present in t_i .

Since there are many real-life situations in which data are uncertain, *efficient algorithms for mining uncertain data* are in demand. To mine frequent patterns from *uncertain* data, Chui et al. [4] proposed an algorithm called *U-Apriori*. Although they also introduced a trimming strategy to reduce the number of candidates that need to be counted, their algorithm is Apriori-based (i.e., relies on the candidate generate-and-test paradigm). Hence, some natural questions to ask are: Can we avoid generating candidates at all? Since tree-based algorithms for handling precise data [8,13] are usually faster than their Apriori-based counterparts [19], is this also the case when handling

* Corresponding author.

uncertain data? In response to these questions, we did a feasibility study [10] on using a tree for mining uncertain data. The study showed that the tree can be used for uncertain data mining. Hence, in the current paper, we propose an efficient tree-based algorithm for mining uncertain data. The **key contributions** of our work are (i) the proposal of an effective tree structure—called a *UF-tree*—for capturing the content of transactions consisting of uncertain data, (ii) the development of an efficient algorithm—called *UF-growth*—for mining frequent patterns from the proposed tree, and (iii) two improvements to the proposed UF-growth algorithm for mining frequent patterns from the UF-tree. Experimental results in Section 5 show the effectiveness of our proposed algorithm in mining frequent patterns from uncertain data.

This paper is organized as follows. The next section gives related work and background. In Section 3 we introduce our UF-growth algorithm for mining frequent patterns from uncertain data. Improvements to this UF-growth algorithm are described in Section 4. Section 5 shows experimental results. Finally, conclusions are presented in Section 6.

2 Related Work and Background

Both the Apriori algorithm [1] and the FP-growth algorithm [8] were designed to handle *precise* data—but not *uncertain* data. A key difference between precise and uncertain data is that each transaction of the latter contains items and their *existential probabilities*. The existential probability $P(x, t_i)$ of an item x in a transaction t_i indicates the likelihood of x being present in t_i . Using the “*possible world*” interpretation of uncertain data [4,5], there are two possible worlds for an item x and a transaction t_i : (i) W_1 where $x \in t_i$ and (ii) W_2 where $x \notin t_i$. Although it is uncertain which of these two worlds be the true world, the probability of W_1 be the true world is $P(x, t_i)$ and that of W_2 is $1 - P(x, t_i)$. To a further extent, there are many items in each of many transactions in a transaction database *TDB*. Hence, the *expected support* of a pattern (or a set of items) X in *TDB* can be computed by summing the support of X in possible world W_j (while taking in account the probability of W_j to be the true world) over all possible worlds:

$$expSup(X) = \sum_j \left[sup(X) \text{ in } W_j \times \prod_{i=1}^{|TDB|} \left(\prod_{x \in t_i \text{ in } W_j} P(x, t_i) \times \prod_{y \notin t_i \text{ in } W_j} (1 - P(y, t_i)) \right) \right] \quad (1)$$

$$= \sum_{i=1}^{|TDB|} \left(\prod_{x \in X} P(x, t_i) \right). \quad (2)$$

With this setting, a pattern X is considered *frequent* if its expected support equals or exceeds the user-specified support threshold *minsup*.

To handle uncertain data, Chui et al. [4] proposed the **U-Apriori algorithm**, which is a modification of the Apriori algorithm. Specifically, instead of incrementing the support counts of candidate patterns by their *actual* support, U-Apriori increments the support counts of candidate patterns by their *expected* support (using Equation (2)). As indicated by Chui et al., U-Apriori suffers from the following problems: (i) Inherited from the Apriori algorithm, U-Apriori does not scale well when handling large amounts

of data because it also follows a levelwise generate-and-test framework. (ii) If the existential probabilities of most items within a pattern X are small, increments for each transaction can be insignificantly small. Consequently, many candidates would not be recognized as infrequent until most (if not all) transactions were processed.

3 Our Proposed UF-Growth Algorithm

In this section, we propose a tree-based algorithm, called **UF-growth**, for mining uncertain data. The algorithm consists of two main operations: (i) the construction of UF-trees and (ii) the mining of frequent patterns from UF-trees.

3.1 Construction of the UF-Tree

As with many tree-based mining algorithms, a key challenge here is how to represent and store data—in this case, uncertain data—in a tree? For precise data, each item in a database transaction TDB is implicitly associated with a definite certainty of its presence in the transaction. In contrast, for uncertain data, each item is explicitly associated with an *existential probability* ranging from a positive value close to 0 (indicating that the item has an insignificantly low chance to be present in TDB) to a value of 1 (indicating that the item is definitely present). Moreover, the existential probability of the item can vary from one transaction to another. Different items may have the same existential probability.

To effectively represent uncertain data, we propose a **UF-tree** which is a variant of the FP-tree. Each node in our UF-tree stores (i) an item, (ii) its expected support, and (iii) the number of occurrence of such expected support for such an item. Our proposed UF-growth algorithm constructs the UF-tree as follows. It scans the database once and accumulates the expected support of each item. Hence, it finds all frequent items (i.e., items having expected support $\geq \text{minsup}$). It sorts these frequent items in descending order of accumulated expected support. The algorithm then scans the database the second time and inserts each transaction into the UF-tree in a similar fashion as in the construction of an FP-tree except for the following:

- The new transaction is merged with a child (or descendant) node of the root of the UF-tree (at the highest support level) only if the same item *and the same expected support* exist in both the transaction and the child (or descendant) nodes.

With such a tree construction process, our UF-tree possesses a nice property that *the occurrence count of a node is at least the sum of occurrence counts of all its children nodes*. See Example 1 for an illustration on constructing a UF-tree.

Example 1. Consider the following transaction database TDB consisting of uncertain data:

Transactions	Contents
t_1	$\{a:0.9, d:0.72, e:\frac{23}{32}=0.71875, f:0.8\}$
t_2	$\{a:0.9, c:0.81, d:0.71875, e:0.72\}$
t_3	$\{b:\frac{7}{8}=0.875, c:\frac{6}{7}\approx 0.85714\}$
t_4	$\{a:0.9, d:0.72, e:0.71875\}$
t_5	$\{b:0.875, c:0.85714, d:0.05\}$
t_6	$\{b:0.875, f:0.1\}$

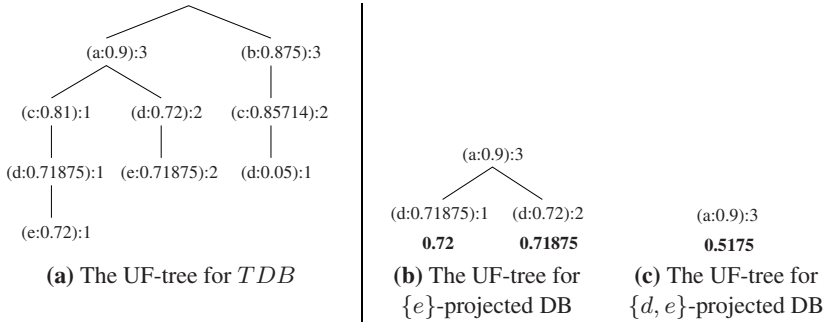


Fig. 1. The UF-trees

Here, each transaction contains items and their corresponding existential probability (e.g., the existential probability of item a in transaction t_1 is 0.9).

Let the user-specified support threshold $minsup$ be set to 1. The UF-tree can be constructed as follows. First, our UF-growth algorithm scans TDB once and accumulates the expected support of each item. Hence, it finds all frequent items and sorts them in descending order of (accumulated) expected support. Items a, b, c, d and e are frequent (i.e., expected support of each of these items $\geq minsup$), with their corresponding accumulated expected support of 2.7, 2.625, 2.52429, 2.20875 and 2.1575. Item f having accumulated expected support of $0.9 < minsup$ is removed because it is infrequent.

Then, UF-growth scans TDB the second time and inserts each transaction into the UF-tree. The algorithm first inserts the content of t_1 into the tree, and results in a tree branch $\langle (a:0.9):1, (d:0.72):1, (e:0.71875):1 \rangle$. It then inserts the content of t_2 into the UF-tree. Since the expected support of a in t_2 is the same as the expected support of a in an existing branch (i.e., the branch for t_1), this node can be shared. So, UF-growth increments the occurrence count for the tree node $(a:0.9)$ to 2, and adds the remainder of t_2 as a child of the node $(a:0.9):2$. As a result, we get the tree branch $\langle (a:0.9):2, (c:0.81):1, (d:0.71875):1, (e:0.72):1 \rangle$. Afterwards, UF-growth inserts the content of t_3 as a new branch $\langle (b:0.875):1, (c:0.85714):1 \rangle$ because the node $(b:0.875):1$ cannot be shared with the node $(a:0.9):2$. Remaining three transactions (t_4 to t_6) are then inserted into the UF-tree in a similar fashion. Consequently, at the end of the tree construction process, we get the UF-tree shown in Fig. 1(a) capturing the content of the above TDB of uncertain data. \square

3.2 Mining of Frequent Patterns from the UF-Tree

Once the UF-tree is constructed, our UF-growth algorithm recursively mines frequent patterns from this tree in a similar fashion as in the FP-growth algorithm except for the following:

- Our UF-growth uses UF-trees (instead of FP-trees) for mining.
- When forming a UF-tree for the projected database for a pattern X , we need to keep track of the expected support (in addition to the occurrence) of X .

- When computing the expected support of an extension of a pattern X (say, $X \cup \{y\}$), we need to multiply the expected support of y in a tree path by the expected support of X .

See Example 2 for an illustration on how the UF-growth algorithm finds frequent patterns from the UF-tree.

Example 2. Once the UF-tree for Example 1 is constructed, our proposed UF-growth algorithm recursively mines frequent patterns from this tree with $minsup=1$ as follows. It starts with item e (with $expSup(\{e\}) = 2.1575$). UF-growth extracts from two tree paths—namely, (i) $\langle\langle a:0.9 \rangle, \langle c:0.81 \rangle, \langle d:0.71875 \rangle\rangle$ occurs once with $(e:0.72)$ and (ii) $\langle\langle a:0.9 \rangle, \langle d:0.72 \rangle\rangle$ occurs twice with $(e:0.71875)$ —and forms the $\{e\}$ -projected DB. Then, $expSup(\{a, e\}) = (1 \times 0.72 \times 0.9) + (2 \times 0.71875 \times 0.9) = 1.94175$, and $expSup(\{d, e\}) = (1 \times 0.72 \times 0.71875) + (2 \times 0.71875 \times 0.72) = 1.5525$. So, both patterns $\{a, e\}$ and $\{d, e\}$ are frequent. However, $\{c, e\}$ is infrequent because $expSup(\{c, e\}) = 1 \times 0.72 \times 0.81 < minsup$. Thus, c is removed from the $\{e\}$ -projected DB. The UF-tree for this $\{e\}$ -projected DB is shown in Fig. 1(b).

Then, the UF-growth algorithm extracts from the UF-tree for the $\{e\}$ -projected DB to form the $\{d, e\}$ -projected DB, which consists of $\{a\}$ (which represents the frequent pattern $\{a, d, e\}$) with $expSup(\{a, d, e\}) = 3 \times 0.5175 \times 0.9 = 1.39725$, where $0.5175 = 0.71875 \times 0.72$ represents $expSup(\{d, e\})$ in this projected DB. The UF-tree for this $\{d, e\}$ -projected DB is shown in Fig. 1(c).

Next, UF-growth deals with items d, c and b (and finds all frequent supersets of $\{d\}$, $\{c\}$ and $\{b\}$) in a similar fashion. Consequently, by applying our proposed UF-growth algorithm to the UF-tree that captures the content of uncertain data in Example 1, we find frequent patterns $\{a\}$, $\{a, d\}$, $\{a, d, e\}$, $\{a, e\}$, $\{b\}$, $\{b, c\}$, $\{c\}$, $\{d\}$, $\{d, e\}$ and $\{e\}$. □

4 Improvements to Our Proposed UF-Growth Algorithm

The UF-tree above may appear to require a large amount of memory. Due to nature of uncertain data, the UF-tree is often larger than the FP-tree. This is because the FP-tree merges nodes sharing the same item whereas the UF-tree merges nodes sharing both the same item and the *same expected support*, where the expected support is in the domain of real numbers in the range of $(0,1]$ —which can be infinitely many. Hence, the chance of sharing a path in the FP-tree is higher than that in the UF-tree. However, it is important to note that, even in the worst case, the number of nodes in a UF-tree is the same as the sum of the number of items in all transactions in the original database of uncertain data. Moreover, thanks to advances in modern technology, we are able to make the same realistic assumption as in many studies [37,15] that *we have enough main memory space* in the sense that the trees can fit into the memory.

A natural question to ask is: Can we reduce the memory consumption? In this section, we discuss how we improve our proposed UF-growth algorithm.

Improvement 1. To reduce the memory consumption and to increase the chance of path sharing, we discretize and round the expected support of each tree node up to k decimal places (e.g., 2 decimal places). By so doing, we reduce the potentially infinite number of possible expected support values—in the domain of real numbers in

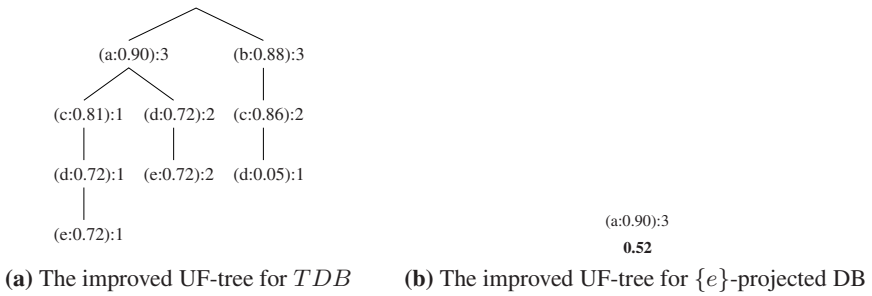


Fig. 2. The improved UF-trees (with Improvement 1)

the range of $(0,1]$ —to a maximum of 10^k possible values (e.g., at most 100 possible expected support values ranging from 0.01 to 1.00 inclusive when $k = 2$). Thus, sizes of the UF-trees for the original TDB and subsequent projected databases are reduced. Fig. 2 shows some of these smaller UF-trees when Improvement 1 is applied.

Improvement 2. Inspired by the idea of the co-occurrence frequent-itemset tree [6], we modify and improve the mining procedure in our proposed UF-growth algorithm so that UF-trees are built *only* for the first two levels (i.e., for the original TDB and for each singleton pattern). In other words, the improved UF-growth does *not* need to build subsequent UF-trees for any non-singleton patterns (e.g., *not* need to build a UF-tree for the $\{d, e\}$ -projected database). Specifically, the improved UF-growth systematically extracts relevant paths from the UF-tree built for each singleton, enumerates all subsets of each extracted tree path, summing the expected support of patterns extracted from these paths to find frequent patterns. See Example 3 for an illustration on how the improved UF-growth algorithm finds frequent patterns from the UF-tree.

Example 3. Similar to Example 2, the improved UF-growth builds a UF-tree for the original TDB (as illustrated in Example 1), finds frequent singleton patterns (e.g., $\{a\}$, $\{b\}$, $\{c\}$, $\{d\}$ and $\{e\}$), forms a projected DB and builds a UF-tree for each of these singletons starting with singleton $\{e\}$. From the $\{e\}$ -projected DB, the improved UF-growth does *not* build any subsequent trees such as $\{d, e\}$ -projected DB. Instead, the algorithm first extracts the tree path $\langle (a:0.9):3, (d:0.71875):1 \rangle$ that occurs once with $(e:0.72)$, enumerates all its subsets and obtains $\{a, e\}$, $\{a, d, e\}$ & $\{d, e\}$ (with their expected supports equal 0.648, 0.46575 & 0.5175 so far), and then decrements the occurrence counts of all nodes in this path. The algorithm then extracts the tree path $\langle (a:0.9):2, (d:0.72):2 \rangle$ that occurs twice with $(e:0.71875)$, enumerates all its subsets and obtains $\{a, e\}$, $\{a, d, e\}$ & $\{d, e\}$ (with their accumulative expected supports equal 1.94175, 1.39725 & 1.5525), and then decrements the occurrence counts of all nodes in this path. Afterwards, all the nodes have occurrence counts equal to 0. We find frequent patterns $\{a, e\}$, $\{a, d, e\}$ & $\{d, e\}$ and their expected supports, directly from the UF-tree representing the $\{e\}$ -projected DB and *without* forming any subsequent UF-trees for non-singletons. Our improved UF-growth applies this technique to other UF-trees for singletons and finds other frequent patterns in a similar fashion. As a result, it finds the same set of frequent patterns as in Example 2 but requires less memory space. \square

Note that Improvement 2 can be applied independently or in conjunction with Improvement 1 (i.e., rounding expected support values).

5 Experimental Results

We conducted the following experiments using various databases including the IBM synthetic datasets [1], real-life databases from the UC Irvine Machine Learning Depository, as well as datasets from the Frequent Itemset Mining Implementation (FIMI) Dataset Repository. The experimental results were consistent. Hence, for lack of space, we only show below the experimental results on the IBM datasets, which contain 100k records with an average transaction length of 10 items and a domain of 1,000 items. We assigned an existential probability from the range (0,1] to each item in each transaction. All experiments were run in a time-sharing environment on a 1 GHz machine. The reported results are based on the average of multiple runs. Runtime includes CPU and I/Os; it includes the time for both tree construction and frequent pattern mining steps. In the experiments, we mainly evaluated the efficiency of the proposed algorithm.

First, we tested the effect of *minsup*. Theoretically, (i) the runtime decreases when *minsup* increases and (ii) our UF-growth algorithm (which does *not* rely on the candidate generate-and-test paradigm) requires much less runtime than the U-Apriori algorithm [4] (which relies on the candidate generation process). Experimental results (as shown in Fig. 3(a)) confirmed that, when *minsup* increased, fewer patterns had expected support $\geq \textit{minsup}$, and thus shorter runtimes were required. Moreover, our tree-based mining algorithm (UF-growth) outperformed its Apriori-based counterpart (U-Apriori).

Second, we tested scalability of our proposed UF-growth algorithm. Theoretically, UF-growth should be scalable with respect to the number of transactions. Experimental results (as shown in Fig. 3(b)) confirmed that mining with our proposed algorithm had linear scalability.

Third, we tested the effect of the distribution of item existential probability. Theoretically, when items take on many different existential probability values, UF-trees (for the original *TDB*, projected databases for singletons as well as for non-singletons) become larger and times for both UF-tree construction and frequent pattern mining become longer. On the other hand, when items take on a few unique existential probability values, the runtime becomes shorter. This is confirmed by experimental results (as shown in Fig. 3(c)). Note that we can reduce the number of unique existential probability values by applying Improvement 1.

Fourth, we measured the number of nodes in UF-trees. Theoretically, our proposed UF-growth described in Section 3 builds UF-trees for the original *TDB* and projected databases for singletons as well as for non-singletons. The total number of nodes in the UF-tree representing the original *TDB* is no more than the total number of items in all transactions in *TDB*. The size of this tree, as well as other UF-trees, built for UF-growth with Improvement 1 is the same—and usually smaller than—that without improvement. Moreover, UF-growth with Improvement 2 builds *only* UF-trees representing the original *TDB* and projected databases for singletons; it does *not* build any UF-trees representing projected databases for non-singleton patterns. See Table 1 and Fig. 3(d). The graph shows the reduction in tree size when $k=2$ decimal places were

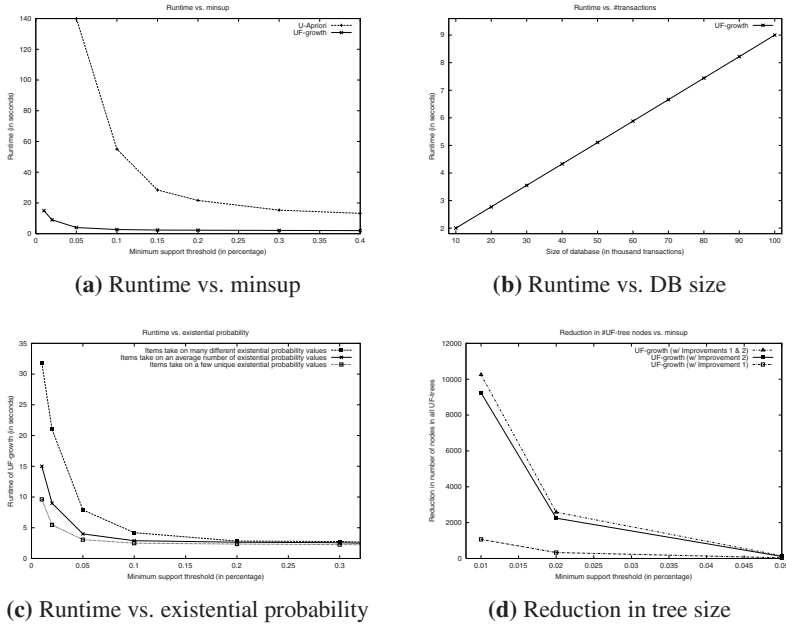


Fig. 3. Experimental results on our proposed UF-growth algorithm

Table 1. Comparison on the sizes of UF-trees for variants of the UF-growth algorithm

UF-trees for...	UF-growth	w/ Improvement 1	w/ Improvement 2	w/ Improvements 1 & 2
<i>TDB</i>	$\#nodes_{TDB}$	$\leq \#nodes_{TDB}$	$\#nodes_{TDB}$	$\leq \#nodes_{TDB}$
singletons	$\#nodes_{sing}$	$\leq \#nodes_{sing}$	$\#nodes_{sing}$	$\leq \#nodes_{sing}$
non-singletons	$\#nodes_{n.s}$	$\leq \#nodes_{n.s}$	0	0

used. More savings were observed when a lower k (e.g., $k=1$ decimal places) was used for Improvement 1.

6 Conclusions

Most existing algorithms mine frequent patterns from traditional transaction databases that contain precise data. However, there are many real-life situations in which one needs to deal with uncertain data. To handle these situations, we proposed (i) the *UF-tree* to effectively capture the content of transaction databases consisting of uncertain data (in which each item in every transaction is associated with an existential probability) and (ii) a tree-based mining algorithm called *UF-growth* to efficiently find frequent patterns from UF-trees. When compared with U-Apriori, our proposed UF-growth algorithm is superior in performance. In addition, we also presented two improvements (which can be applied independently or simultaneously) to UF-growth. The rounding

of expected support values and the elimination of UF-trees for projected databases for non-singleton patterns both contribute to the reduction of the amount of required memory and further speed-up of the mining process. Hence, with our tree-based approach, users can mine frequent patterns from uncertain data effectively.

Acknowledgement. This project is partially supported by NSERC (Canada) in the form of research grants.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules. In: Proc. VLDB, pp. 487–499 (1994)
2. Bonchi, F., Lucchese, C.: Pushing tougher constraints in frequent pattern mining. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 114–124. Springer, Heidelberg (2005)
3. Cheung, W., Zaïane, O.R.: Incremental mining of frequent patterns without candidate generation or support constraint. In: Proc. IDEAS, pp. 111–116 (2003)
4. 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)
5. Dai, X., Yiu, M.L., et al.: Probabilistic spatial queries on existentially uncertain data. In: Bauzer Medeiros, C., Egenhofer, M.J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 400–417. Springer, Heidelberg (2005)
6. El-Hajj, M., Zaïane, O.R.: COFI-tree mining: a new approach to pattern growth with reduced candidacy generation. In: Proc. FIMI (2003)
7. Giannella, C., Han, J., et al.: Mining frequent patterns in data streams at multiple time granularities. In: Data Mining: Next Generation Challenges and Future Directions, ch. 6. AAAI/MIT Press (2004)
8. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Proc. SIGMOD, pp. 1–12 (2000)
9. Lakshmanan, L.V.S., Leung, C.K.-S., Ng, R.T.: Efficient dynamic mining of constrained frequent sets. ACM TODS 28(4), 337–389 (2003)
10. Leung, C.K.-S., Carmichael, C.L., Hao, B.: Efficient mining of frequent patterns from uncertain data. In: Proc. IEEE ICDM Workshops, pp. 489–494 (2007)
11. Leung, C.K.-S., Irani, P.P., Carmichael, C.L.: FIsViz: A frequent itemset visualizer. In: Washio, T., Suzuki, E., Ting, K.M., Inokuchi, A. (eds.) PAKDD 2008. LNCS (LNAI), vol. 5012, pp. 644–652. Springer, Heidelberg (2008)
12. Leung, C.K.-S., Khan, Q.I.: DStree: a tree structure for the mining of frequent sets from data streams. In: Proc. IEEE ICDM, pp. 928–932 (2006)
13. Leung, C.K.-S., Lakshmanan, L.V.S., Ng, R.T.: Exploiting succinct constraints using FP-trees. SIGKDD Explorations 4(1), 40–49 (2002)
14. Ng, R.T., Lakshmanan, L.V.S., et al.: Exploratory mining and pruning optimizations of constrained associations Rules. In: Proc. SIGMOD, pp. 13–24 (1998)
15. Pei, J., Han, J., Mao, R.: CLOSET: an efficient algorithm for mining frequent closed itemsets. In: Proc. SIGMOD Workshop on DMKD, pp. 21–30 (2000)

Connectivity Based Stream Clustering Using Localised Density Exemplars

Sebastian Lühr and Mihai Lazarescu

Department of Computing, Curtin University of Technology,
Kent Street, Bentley 6102, Western Australia
{S.Luhr,M.Lazarescu}@curtin.edu.au

Abstract. Advances in data acquisition have allowed large data collections of millions of time varying records in the form of data streams. The challenge is to effectively process the stream data with limited resources while maintaining sufficient historical information to define the changes and patterns over time. This paper describes an evidence-based approach that uses representative points to incrementally process stream data by using a graph based method to cluster points based on connectivity and density. Critical cluster features are archived in repositories to allow the algorithm to cope with recurrent information and to provide a rich history of relevant cluster changes if analysis of past data is required. We demonstrate our work with both synthetic and real world data sets.

1 Introduction

Stream mining is an increasingly important area of research that aims to discover interesting information from continually evolving data sets whose size, combined with limitations in available memory and computational resources, typically constrains our ability to perform timely batch processing of the data. Instead, we desire means by which to incrementally maintain current and historical models of the data with which to perform queries. Stream data mining has been heavily investigated in the past five years with most efforts concentrated on the clustering aspect of the problem. Of the algorithms developed, however, only a small number can handle difficult clustering tasks without expert help, typically provided in the form of the number of partitions expected or the expected density of clusters. Moreover, none of these attempt to build a *rich* history to track the underlying changes in the clusters observed.

We present a sparse-graph based stream mining approach that employs representative cluster points to incrementally process incoming data. The graph based description is used because it allows us to model the spatio-temporal relationships in a data stream more accurately than is possible via summary statistics. A critical aspect of our research has been to avoid rediscovery of previously learned patterns by reusing useful cluster information. For this reason, a repository of knowledge is used to capture the history of the relevant changes occurring in the clusters over time. The use of the repository offers two major benefits.

First, the algorithm can handle recurrent changes in the clusters more effectively by storing a concise representation of persistent and consistent cluster features. These features assist in the classification of new data points belonging to historical cluster distributions within an evolving data stream. The retention of such features is important as they permit the algorithm to discard older data points in order to adhere to constraints in available memory and computational resources while continuing to store useful cluster features.

Second, the repository provides a concise knowledge collection that can be used to rebuild a cluster's overall shape and data distribution history. It is therefore possible to archive core cluster features for future off-line analysis when a recall of historical changes is desired.

2 Related Work

Several important stream mining algorithms have been introduced in recent years. One of the first data stream mining methods to consider the archival of cluster information was CluStream [1]. The algorithm uses microclusters to capture and record statistical summary information suitable for off-line analysis. CluStream is, however, best suited to situations in which clusters are spherical, reducing the algorithm's suitability to many real world data sets.

HPSStream, a modification of CluStream to enable clustering of high dimensional data, was proposed in [2]. The algorithm employs a data projection method to reduce the dimensionality of the data stream to a subset of dimensions that minimise the radius of cluster groupings. However, the underlying assumption remains that clusters in the projected space remain spherical in nature.

Most recently, a multi-density clustering technique that extends the DBSCAN [3] density-based clustering approach to stream mining was proposed in [4]. The algorithm, DenStream, extends DBSCAN by adapting the original density based connectivity search to a microcluster approach.

An incremental version of the DBSCAN was earlier proposed in [5]. As with DBSCAN, the algorithm obtains groupings based on the nearest neighbourhood connectivity of points within an a priori defined radius known as the ϵ -neighbourhood. Incremental DBSCAN is limited to keeping only the most recent data points in memory and is therefore likely to discard possibly reusable cluster information without consideration towards its value.

A well known algorithm, Chameleon [6] uses the hMEIS [7] multilevel graph partitioning algorithm to recursively divide a sparse graph into micro-clusters. These clusters are then iteratively merged based on user specified thresholds for measures of relative interconnectivity and closeness.

None of the algorithms mentioned provide a means to archive historical information. Those algorithms that facilitate archiving instead tend to store summary statistics with which general changes in clusters can be revisited.

3 Clustering Stream Data Via Representative Points

Our cluster representation involves the use of dynamically updated sparse graphs that, when used in conjunction with a repository of representative vertices, allows us to rebuild a cluster's history and to rapidly adapt to significant changes previously observed. The RepStream algorithm that we propose aims to capture such change in order to recall it at some future time should the change reoccur. RepStream is a single phase incremental algorithm that updates two sparse graphs of k -nearest neighbour connected vertices in order to identify clusters among data points. The first graph captures the connectivity relationship amongst the most recently seen data points and to select a set of representative vertices. The second graph is used to track the connectivity between the chosen representative vertices. The connectivity of the representative vertices on both graphs then forms the basis for the algorithm's clustering decision making.

The representatives we use offer two major advantages. First, since representative vertices typify a set of nearby data points, decisions made at this level improves performance by requiring only a subset of the data to be considered. Second, representative vertices are associated with a measure of usefulness which allows the algorithm to selectively retain highly valued representatives as historical descriptors of the cluster structures. This retention allows the algorithm to accurately classify new data points arriving within a region of the clustering space where the non-representative vertices have since been retired.

3.1 Preliminaries

Given a data stream P of time ordered points $P = \{p_1, \dots, p_{|P|}\}$, we wish to find groupings of points sharing similar properties. We define a cluster c to be a set of points $c = \{p_1 \dots p_{|c|}\}$ where each point p_i is a multidimensional vector $p_i = \{p_{i,1} \dots p_{i,D}\}$ of D dimensions. Let C be the set of clusters $C = \{c_1 \dots c_{|C|}\}$.

Let the set $G = \{g_1 \dots g_{|P|}\}$ be the ideal cluster assignments for points P such that the j^{th} element g_j correctly labels point p_j . We aim to assign labels to data points such that each point is correctly classified or any misclassification is minimised. The distance between point p_i and point p_j is given as $D(p_i, p_j)$.

Points are inserted into a directed k -nearest neighbour (K-NN) sparse graph $SG(V, E)$ of vertices $V = \{v_1, \dots, v_{|V|}\}$ and edges $E = \{e_1, \dots, e_{|E|}\}$ such that the i^{th} vertex v_i corresponds to point $p_i \in P$. Each edge is an ordered pair $\langle u, v \rangle$ of vertices such that $u, v \in V$. The sparse graph representation is used as it provides a rich representation of relationships that is otherwise not available by only labelling data points.

Updates to the sparse graph requires knowledge of each vertex's nearest neighbours. Let $NN(v_i)$ be a function that provides an ascending distance ordered array of the nearest neighbours of a vertex v_i and let $NN(v_i, j)$ be a function that gives the j^{th} nearest neighbour of v_i . Let $RC(v_i)$ be a function that provides a set of vertices reciprocally connected to a vertex v_i . We also let $IE(v_i)$ be a function for determining the incoming edges directed at vertex v_i .

Let $R = \{r_1, \dots, r_{|R|}\}$ be a set of representative vertices on SG such that $\forall x, r_x \in V$ and let $\text{RSG}(W, F)$ be a directed k -nearest neighbour representative sparse graph which links the vertices $W = \{w_1, \dots, w_{|W|}\}$ via edges $F = \{f_1, \dots, f_{|F|}\}$. An edge in F is an ordered pair $\langle u, v \rangle$ of vertices such that $v, u \in R$. Let $\text{NN}_R(r_i)$, $\text{NN}_R(r_i, j)$ and $\text{RC}_R(r_i)$ be functions that provide the nearest neighbours, the nearest j^{th} neighbour and the set of vertices that are reciprocally linked to a representative vertex r_i on RSG.

Definition (predictor). Let a representative r_i be a predictor if r_i satisfies the condition that $|\text{IE}(r_i)| < \frac{k}{2}$.

Definition (exemplar). Let a representative r_i be an exemplar if r_i satisfies the condition that $|\text{IE}(r_i)| \geq \frac{k}{2}$.

Definition (representative vertex). Representative vertices represent at most k non-representative vertices on the sparse graph SG. A vertex v_i is made representative if at any time $\nexists j, v_j \in \text{RC}(v_i), v_j \in R$, that is, if it is not reciprocally connected to an existing representative. Representatives are further categorised into a set of exemplar representatives $R^E = \{r_1^E, \dots, r_{|R^E|}^E\}$ and predictor representatives $R^P = \{r_1^P, \dots, r_{|R^P|}^P\}$ such that $R^P \cup R^E = R$ and $R^P \cap R^E = \emptyset$.

Clustering decisions in RepStream are made via vertices representative of regions within the cluster space. At each time step a new point p_i is observed in the data stream and added to the sparse graph $\text{SG}(V, E)$ as vertex v_i . A new vertex joins an existing cluster if it is reciprocally connected to a representative $v_j \in R$. Should no such representative vertex exist then v_i is itself made representative. The creation of the new cluster may trigger an immediate merge with an existing cluster if the conditions for merging are met.

3.2 Merging and Splitting Clusters

Cluster splits and merges are made by monitoring both the reciprocal connectivity of vertices on the representative sparse graph as well as their density based on the proximity of their nearest neighbours on SG. The trigger condition for either of these events is the creation or removal of reciprocal links.

Definition (relative density). The density of representative vertex $r_i \in R$ is determined by the function $\text{RD}(r_i) = \frac{1}{|\text{NN}(r_i)|} \sum_{j=1}^{|\text{NN}(r_i)|} \text{D}(r_i, \text{NN}(r_i, j))$.

Definition (density-related). Given a density scaler α , two representatives r_i and r_j are density-related if: $\text{D}(r_i, r_j) \leq \text{RD}(r_i) \cdot \alpha$, and $\text{D}(r_i, r_j) \leq \text{RD}(r_j) \cdot \alpha$ and $r_j \in \text{RC}_R(r_i)$.

Merges are therefore triggered when an update to the connectivity of vertices on RSG sees the creation of a new reciprocal connection that is also density-related or when the addition or removal of a vertex affects the density of two existing representatives that are reciprocally connected such that their density-related status is altered. Monitoring the connectivity and relative density of representatives enables the algorithm to evolve with changes in the data.

Split checks are executed when the loss of a density-related link between two vertices on RSG is detected. A standard $O(n^2)$ region growing algorithm that follows the density-related links of the representative vertices was employed to perform split checks.

3.3 Knowledge Repository

A significant aim of RepStream is to retain those representative vertices that prove, over time, to be useful in representing the shapes and distributions of clusters. Such vertices are retained for as long as possible (subject to available resources) via a repository defined as an ordered vector of vertices $S = \langle s_1, \dots, s_{|S|} \rangle$ sorted in ascending usefulness.

Definition (representative usefulness). The usefulness of a representative vertex r_i is defined by the decay function: $\text{usefulness}(r_i, \text{count}) = \log(\lambda) \cdot (\text{current time} - \text{creationTime}(r_i) + 1) + \log(\text{count} + 1)$. Here λ is a user specified decay rate and count is the representative vertex's reinforcement count. This count is incremented when an incoming vertex is found to be a nearest neighbour of r_i .

The decay function ensures a monotonic ordering of vertices in the repository with respect to the passing of time. In our implementation of RepStream we chose to index the repository using a AVL binary search tree [8]. Updating the reinforcement count of a representative vertex that has already been added to the repository requires only two tree operations: the removal of the vertex and then its subsequent reinsertion following an increment to its reinforcement count. The least useful representative vertex can be rapidly found by traversing to the AVL tree node with the lowest usefulness score.

New additions to the repository are made whenever a new representative vertex is created until resource constraints have been reached. At this point only the most useful repository members are retained. This is achieved by comparing the least useful repository member with other non-repository representatives whenever their reinforcement count is incremented. Vertices retired from the repository are immediately unlinked from both graphs and archived to disk.

3.4 Singularities

The occurrence of many identical points within a data stream is captured via *singularities*, a special case of representative vertices intended to succinctly and efficiently represent such occurrences.

Definition (singularity). A representative vertex $r_i \in R$ is termed a singularity when $\sum_{j=1}^k D(r_i, \text{NN}(r_i, j)) = 0$ and $|\text{NN}(r_i)| = k$.

Singularities represent a collection of identical points that offer no new information to the clustering process, yet whose inclusion in the sparse graphs would require the retirement of otherwise useful vertices. New points that are identical to a singularity are therefore immediately deleted in order to avoid the overhead of unnecessary sparse graph updates and to maintain the information value of

the repository. The occurrence of identical points is not lost, however, as they are represented by a singularity's reinforcement count.

Singularities are unable to be assigned non-zero density measures and as such do not lose their singularity status once it is acquired. This ensures that the presence of a singularity is permanently captured by the algorithm even though its nearest neighbours may be retired over time. Representative vertices are unable to form density-related links to singularity vertices.

3.5 Data Retirement

Processing and memory constraints require the algorithm to discard information over time. This is accomplished by prioritising the disposal of data such that the least useful information for clustering is removed first. Non-representative vertices are queued on a first in, first out basis and removed whenever resource limitations are reached. Representative vertices that are not stored in the repository are considered to have little retentive value and are also removed via the deletion queue. All other representative vertices remain in memory; their deletion is instead managed via the repository update procedure.

The removal of a vertex requires updates to the sparse graph and the representative sparse graph. Graph updates are made to ensure that any vertices with edges directed at the removed vertex are updated with a new nearest neighbour. Representative vertices are also updated to ensure that their local density is adequately maintained.

4 Experimental Results

The performance of RepStream was evaluated using synthetic and real world data sets. Our real world data sets consisted of the KDD-99 Cup network intrusion data and the forest canopy type data described in [9]. The synthetic data was designed to test the algorithm's capacity to cluster a difficult set containing a variety of arbitrarily shaped clusters of different densities. The real world data sets, in contrast, were selected to investigate the practical application of the approach on large evolving data streams.

Cluster purity [10] was used to measure how well data is classified over a horizon of the previous h data points. The purity of a cluster c_i is defined as: $CP(c_i) = \frac{1}{|c_i|} \max_k (\sum_{j=1}^{|c_i|} r(v_j, k))$ where $r(v_j, k)$ is 1 if $\text{class}(v_j) = k$, else 0.

The total clustering purity is then found by averaging over all clusters via: $TCP(C) = \frac{1}{|C|} \sum_{i=1}^{|C|} CP(c_i)$.

The algorithm was constrained to using only 10 MiB of memory and the decay factor used in all experiments was set to $\lambda = 0.99$. The chosen purity horizons were selected to correspond with previous work in clustering data streams [12]. The KD-Tree [11] was used to perform nearest neighbour searches.

4.1 Synthetic Data

The clustering quality of RepStream was first compared against an incremental version of DBSCAN [5] using the hand crafted synthetic data set. DBSCAN

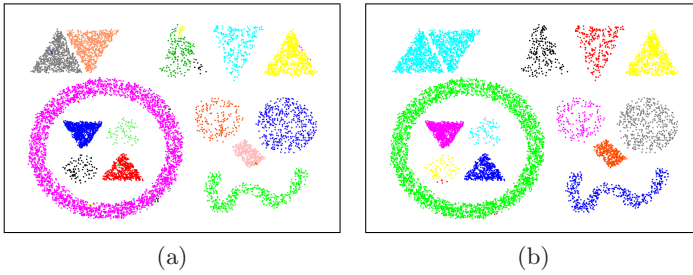


Fig. 1. RepStream clustering of the synthetic data highlighting the performance difference between a neighbourhood connectivity of (a) $k = 4$ and (b) $k = 5$ when $\alpha = 4.0$

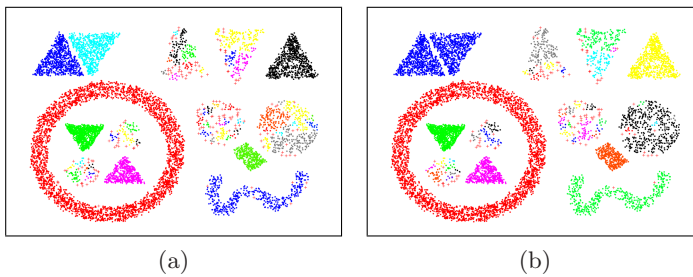


Fig. 2. DBSCAN clustering of the synthetic data set with (a) $\epsilon = 15$ and (b) $\epsilon = 16$

was selected for comparison as this algorithm employs a density based method of clustering known to perform well with arbitrarily shaped clusters. However, DBSCAN is limited to operating at a single density and is therefore expected to exhibit difficulties when dealing with this data set. As DBSCAN relies on a priori knowledge of the optimal cluster density, we repeated each of the DBSCAN experiments using a variety of values for ϵ . The minimum number of points required to form a cluster was set to 5. The data was presented to the algorithms using a randomised point ordering and the Manhattan distance was used to compute the similarity between points.

Figure 1 depicts the RepStream clustering of the data using the optimal parameter set $k = 4$ and $\alpha = 4.0$. These results show that the algorithm was able to cluster the arbitrarily shaped clusters well. The discovered clusters are sub-optimal, however, with some minor fragmentation evident. The separate clustering of these points is not considered an error, however, as their location and density suggests that these points may, indeed, belong to separate clusters when compared to the remaining points.

Increasing the density scaler from $\alpha = 4.0$ to a higher value of $\alpha = 6.0$ did not correct this clustering. Decreasing the scaler did, however, result in increased fragmentation. An increase of the neighbourhood connectivity successfully overcame the fragmentation issue as shown in Figure 1(b).

In contrast, DBSCAN was found to produce well formed higher density clusters with an ϵ -neighbourhood parameter of $\epsilon = 15$. The lower density clusters, however, were found to be highly fragmented with the presence of a significant number of unclustered points as shown in Figure 2(a). Decreasing the density with $\epsilon = 16$ marginally decreased the cluster fragmentation, as seen in Figure 2(b), though at the expense of the incorrect merging of the two top left triangular clusters.

4.2 Network Intrusion Data

The KDD Cup-99 data set features 494,020 network connection records derived from seven weeks of raw TCP logs consisting of both regular network traffic as well as 24 types of simulated attacks within a military local area network. Of the dimensions available, 34 continuous valued features were used for clustering and a single outlier point was removed.

RepStream was tested using a purity horizon of $h = 1,000$. The Manhattan distance function was used to compute the similarity of data points from features that were normalised on-the-fly. A point $p_i = \{p_{i,1} \dots p_{i,D}\}$ of D dimensions was normalised in each dimension d using the formula $p'_{i,d} = \frac{p_{i,d}}{\sum_{j=1}^{|P|} p_{j,d}}$ where $|P|$ refers to the number of points in memory at any given time. The nearest neighbourhood connectivity was set to $k = 9$ with $\alpha = 1.5$.

The purity results in Figure 3 show that RepStream is able to accurately differentiate between different types of attack connections. The accuracy of RepStream was also evaluated against published results reported on the same data set for the HPStream, DenStream and CluStream algorithms. The results of the

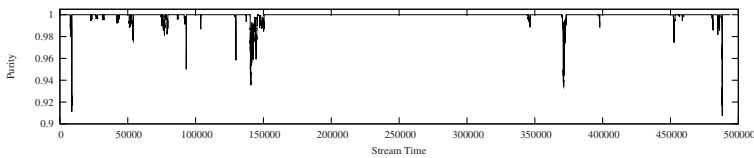


Fig. 3. RepStream purity throughout the network intrusion data stream

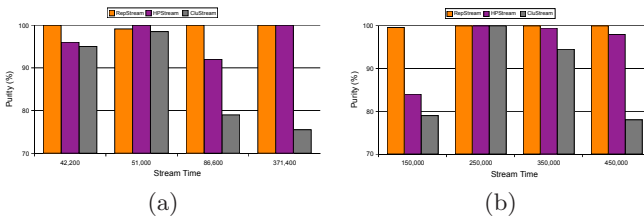


Fig. 4. Purity measures of RepStream, HPStream and CluStream using available published results on the KDD Cup 1999 data set with (a) $h = 200$ and (b) $h = 1,000$

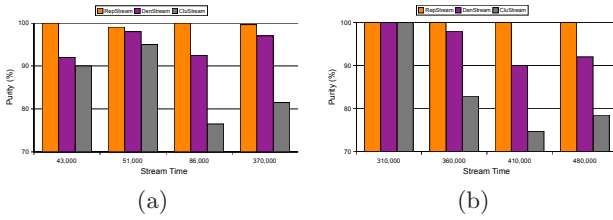


Fig. 5. Purity measures of RepStream, DenStream and CluStream using available published results on the KDD Cup 1999 data set with (a) $h = 200$ and (b) $h = 1,000$

comparisons, depicted in Figure 4 and in Figure 5, shows that in most cases RepStream was able to classify network connections as well as or with higher accuracy than HPStream, DenStream and CluStream. The data stream sample times were chosen to match those reported in [12].

4.3 Forest Cover Data

The forest cover data set contained 581,012 records consisting of a total of 54 geological and geographical features that describe the environment in which trees were observed. Records also included the ground truth as to which of seven different types of canopy were present on the trees. Attributes consisted of a mixture of continuous and Boolean valued data, the latter taking values from the set $\{0,1\}$. Dimensions were normalised as described in Section 4.2 and the Manhattan distance function was used to measure the similarity between points. Parameters used on this data set were $k = 9$ and $\alpha = 1.5$.

Figure 6 shows the purity measured over the data stream with $h = 1,000$. RepStream is seen to classify the canopy types with an accuracy typically $\geq 85\%$. The jagged appearance of the purity plots suggest that the algorithm is coping with a more dynamic data set than compared to the network intrusion experiment in Section 4.2; a premise confirmed through inspection of the data. RepStream’s purity measurements were evaluated against HPStream and CluStream using the results published in [2]. Figure 7 depicts the result of this comparison, showing that the algorithm was able to classify the tree data with consistently more accuracy than the competing algorithms.

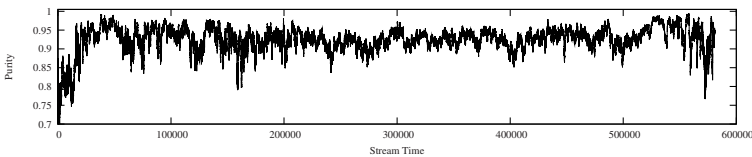


Fig. 6. RepStream purity throughout the tree cover data stream

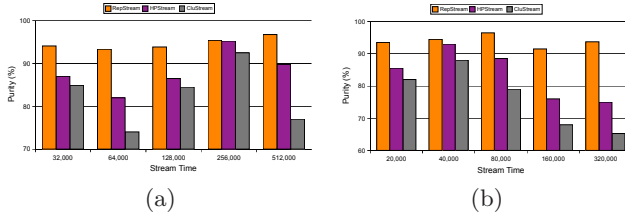


Fig. 7. Purity measures of RepStream, HPStream and CluStream using available published results on the forest tree cover data set with (a) $h = 200$ and (b) $h = 1,000$

4.4 Scale-Up Experiments

We investigated the execution time of the algorithm with respect to neighbourhood connectivity and the length of the data stream. Scale up experiments were executed on Mac OS 10.4 running on an Intel 2.33GHz Core 2 Duo processor.

A near linear relationship between connectivity and execution time was discovered in the network intrusion results in Figure 8a. The forest data set produced a similar relationship as shown in Figure 8b. Execution time with respect to the length of the data stream is shown in Figure 9.

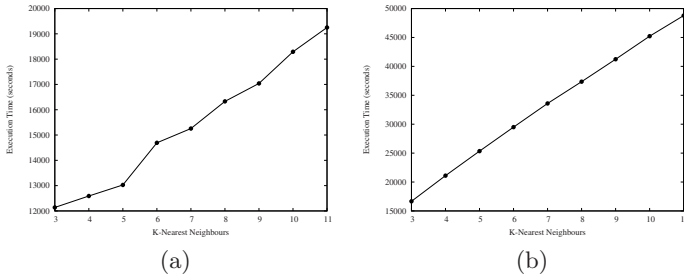


Fig. 8. Execution time of RepStream clustering (a) the network intrusion data and (b) the forest canopy data as the k -nearest neighbours are increased

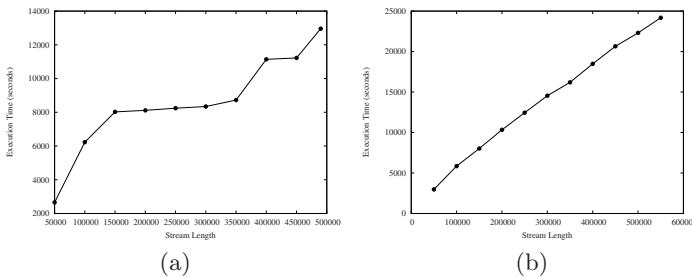


Fig. 9. Execution time of RepStream clustering the (a) network intrusion data and (b) the forest canopy data as the stream length is increased

Whereas the tree data set in Figure 9b shows an expected linear relationship between the number of points processed and the execution time, the network data set in Figure 9a displays significant flattening out due to efficient processing of identical points within the stream. Connectivity was set to $k = 5$ and a density scaler of $\alpha = 1.5$ was used to process both data sets.

5 Conclusions

This paper has introduced a graph-based incremental algorithm for clustering evolving stream data. Experimental results demonstrated that the algorithm was able to effectively classify both synthetic and real world data sets. The algorithm was compared against an incremental implementation of DBSCAN and shown to robustly handle clusters of complex shapes, sizes and densities. DBSCAN, in contrast, was shown to be hampered by a static density threshold ill suited towards stream processing. Results on real world data sets showed that RepStream was able to more accurately classify well known network intrusion and forest canopy data sets than three of the most popular stream data clustering algorithms: DenStream, HPStream and CluStream.

References

1. Aggarwal, C.C., Han, J., Wang, J., Yu, P.: A framework for clustering evolving data streams. In: Proc. 29th Int'l Conf. Very Large Data Bases (2003)
2. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for projected clustering of high dimensional data streams. In: Proc. Very Large Data Bases, pp. 852–863 (2004)
3. 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'l Conf. Knowledge Discovery and Data Mining, pp. 226–231 (1996)
4. Cao, F., Ester, M., Qian, W., Zhou, A.: Density-based clustering over an evolving data stream with noise. In: Proc. Sixth SIAM Int'l Conf. Data Mining (2006)
5. Ester, M., Kriegel, H.P., Sander, J., Wimmer, M., Xu, X.: Incremental clustering for mining in a data warehousing environment. In: Proc. 24rd Int'l Conf. Very Large Data Bases, pp. 323–333 (1998)
6. Karypis, G., Han, E.H., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. *Computer* 32(8), 68–75 (1999)
7. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: Application in VLSI domain. *IEEE Trans. Very Large Scale Integration (VLSI) Systems* 7(1), 69–79 (1999)
8. Knuth, D.: *The Art of Computer Programming*, 3rd edn., vol. 3 (1997)
9. Blackard, J.A., Dean, D.J.: Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. *Computers and Electronics in Agriculture* 24(3), 131–151 (1999)
10. Aggarwal, C.C.: A human-computer interactive method for projected clustering. *IEEE Trans. Knowledge and Data Engineering* 16(4), 448–460 (2004)
11. Bentley, J.L.: Multidimensional binary search trees used for associative searching. *Communications of the ACM* 18(9), 509–517 (1975)

Learning User Purchase Intent from User-Centric Data

Rajan Lukose¹, Jiye Li², Jing Zhou³, and Satyanarayana Raju Penmetsa¹

¹ HP Labs, Palo Alto, California

{[rajan_lukose](mailto:rajan_lukose@hp.com), [satyanarayana.raju](mailto:satyanarayana.raju@hp.com)}@hp.com

² Faculty of Computer Science and Engineering, York University

jiye@cse.yorku.ca

³ Belk College of Business, UNC Charlotte

jzhou7@email.uncc.edu

Abstract. Most existing personalization systems rely on site-centric user data, in which the inputs available to the system are the user’s behaviors on a specific site. We use a dataset supplied by a major audience measurement company that represents a complete user-centric view of clickstream behavior. Using the supplied product purchase metadata to set up a prediction problem, we learn models of the user’s probability of purchase within a time window for multiple product categories by using features that represent the user’s browsing and search behavior on all websites. As a baseline, we compare our results to the best such models that can be learned from site-centric data at a major search engine site. We demonstrate substantial improvements in accuracy with comparable and often better recall. A novel behaviorally (as opposed to syntactically) based search term suggestion algorithm is also proposed for feature selection of clickstream data. Finally, our models are not privacy invasive. If deployed client-side, our models amount to a dynamic “smart cookie” that is expressive of a user’s individual intentions with a precise probabilistic interpretation.

1 Introduction

Clickstream data collected across all the different websites a user visits reflect the user’s behavior, interests, and preferences more completely than data collected from one site. For example, one would expect that it would be possible to better model and predict the intentions of users who we knew not only searched for a certain keyword on a search engine S but also visited website X and the website Y , than if we knew only one of those pieces of information. The complete data set is termed user-centric data [8], which contains site-centric data as a subset. Most existing research on clickstream data analysis is based on site-centric data.

For the important task of personalization we seek to demonstrate rich, predictive user models induced from user-centric data, and quantify their advantages to site-centric approaches. We use a dataset supplied by a major audience measurement company that represents a complete user-centric view of clickstream

behavior. The main contribution of our work is the first demonstration that accurate product category level purchase prediction modeling (regardless of the site of purchase) can be done from user-centric data. Using the supplied product purchase metadata to set up a prediction problem, we learn models of the user’s probability of purchase within a time window for multiple product categories by using features that represent the user’s behavior on all websites. Our model outperforms a reasonable and commercially meaningful baseline model learned from site-centric data restricted to a major search engine. We also propose a novel behaviorally (as opposed to syntactically) based search term suggestion algorithm which was an effective part of the feature selection strategy we used. Additionally, we explicitly consider the issue of prediction latency and show that even when predictions are made with long lead times, effective predictions can still be made. Finally, our models are not privacy invasive and we propose the idea of “smart cookies” motivated by our results. The success of our clickstream modeling approach should point the way to more personalization applications driven by clickstream modeling.

We first review the related background work in clickstream modeling and current research on personalization in Section 2. We then introduce our proposed online product purchase model and describe our experimental data in Section 3. Section 4 provides the experimental design and results.

2 Related Work

In the computer science literature, two main motivations have driven research on clickstream analysis: personalization and caching. Caching and prefetching to improve web server performance is obviously an important task and so site-centric clickstreams from web server logs have been analyzed to improve performance [4]. This line of work has emphasized the use of Markov models to predict page accesses. Despite a broad and deep interest, little direct work has been done on mining user-centric clickstream data for personalization. Site-centric personalization efforts have used clickstream analysis to cluster users [1,2] which enables site-specific content recommendation within user clusters. Additional work has been done in the marketing science literature [6] and [7]. User-centric clickstream data has been used in web personalization tasks such as personalized search [10], where clickstream data was part of the data used to help re-rank search results. Padmanabhan, [8] demonstrated the predictive value of user-centric data versus site-centric data. Their work attempted to provide predictions of “purchase” or “no-purchase” at a given website (regardless of specific product category) based on user or site-centric data as inputs. In our work, we focus on the more widely useful and more difficult task of predicting specific purchases at website. Furthermore, we consider search data as an important feature whose value as a prediction variable we are able to quantify and which was not used in this prior work.

3 Purchase Intent Model

This work focuses on developing general models that can effectively learn and predict users' online purchase intent. In these models, user-centric data is collected and stored in a database. After data preprocessing, features reflecting user online purchase intentions are constructed. The search terms that users input into general search engines, and the search terms they use on the leading online shopping stores are considered as indications of their purchasing interests (see [5] for more details). Then algorithms, such as decision trees, regression prediction algorithms are applied for predicting online purchase intent for various product categories on the processed data composed of the constructed features. We further explain the experimental dataset used, a search term suggestion algorithm, data preprocessing, feature construction and evaluations for the modeling process in the rest of this section.

3.1 Experimental Data

Nielsen Online MegaPanel data¹ is used as our testbed for purchase intent modeling. Nielsen is an online audience measurement company, which is a premier provider of high-quality internet data. The MegaPanel data is raw user-centric clickstream data, which includes, for example, online search behavior on leading search engines (such as Google, Yahoo) and shopping websites (such as Amazon, BestBuy). The data collection is processed to make the average customer's online behaviors consistent with a representative sampling of internet users. All personally identifying data is filtered from our dataset by Nielsen.

The data collected over 8 months amounted to approximately 1 terabyte from more than 100,000 households. For each URL there are time stamps for each internet user's visit. Retailer transaction data (i.e. purchase metadata) contains more than 100 online leading shopping destinations and retailer sites. These data records show for a given user who makes a purchase online, the product name, the store name, the timestamp, the price and so on. Users' search terms can also be inferred from the URL data, which are collected from top search engines and comparison shopping sites (more details are given in [5]).

3.2 Behavior Based Search Term Suggestion Algorithm

Automatic discovery of relevant search terms can help construct features to distinguish buyers from non-buyers given a product category. The search terms users input into websites are indications of their purchasing intent, but it is a challenge to determine automatically which terms are relevant for a given product category. Current keyword suggestion tools are *query-based*, typically suggesting variations of queries that include a given seed search term. For example, for the purchase of "laptop", suggested keywords may include "laptops". Our approach is *context-based*, does not use any information about syntactic

¹ <http://www.nielsen-netratings.com/>

Table 1. a) Top 10 Significant Terms for Sample Product Categories, b) Decision Table for Classifications

Apparel	Auto-motives	Books	Child BabyCare	Watch & Jewelry	Computer Hardware	Computer Software
granby	rotonda	amazon	thum	Seiko	dell	cafepress
coupon	civic	barnes	cravens	watches	dotnetnuke	panel
centreville	rotundra	books	aod	ebay	ati	hdtv
coupons	hfp	noble	mysterie811	movado	radeon	flfat
shirts	ep3	goya	hohider	overstock.com	behringer	scripps
wrightsville	rechenberg	miquelon	strollers	watche	agp	plasma
clothing	bove	amnapolis	pomade	xbox	laborer	kingman
pajamas	exhaust	diseases	dragonflies	Timex	hp	software
transat	switchers	autograph	toolady	Watchband	breakin	scroll
shirt	ifinder	griffie	gumball	Necklaces	blau	1080i

User ID	Condition Attributes 28 Features						Decision Attribute {buyer, non-buyer}
ID	G1a	G1b	...	G14c	G11	G16	{buyer, non-buyer}
1	Yes	2	...	7	5200	No	buyer
2	Yes	5	...	2	413	Yes	non-buyer
3	No	0	...	0	622	No	buyer
...
83,635	Yes	3	...	0	342	No	buyer

(a)

(b)

variation of queries, and does not even require seed terms. For example, related keywords under this method may include brand names such as “HP laptop”, “Dell” and “Lenovo” with no syntactic relationship to “laptop”.

We used the following algorithm to automatically generate a set of representative search terms. First, given a product category, we counted the frequencies of all the search terms observed from buyers over a certain period of time. Then we found which search terms are significantly different in frequency within the buyer population of our training data from the search terms which appear in the general population of buyers and non-buyers by using a χ^2 test on each of the 26 product categories.

December 2005 data is used as our experimental data. We list the top 10 significant terms for sample product categories² in Table 1(a). This algorithm⁵ was used for constructing useful features for our models in an automated way, but is also effective as a search term suggestion algorithm in more general contexts. For example, as can be seen in Table 1(a), this method identifies terms that do not include, and have no syntactic similarity to the word “watch” such as simple brand names like “seiko”, “movado”, and “timex” as well as misspellings such as “watche” and even other terms like “necklace”.

3.3 Feature Construction

We focus on constructing features that can reflect the users’ browsing and searching behaviors across multiple websites using user-centric data. There are 26 on-line product categories available in our experimental data. In this experiment, we consider the online purchasing product category to be personal computers, including both desktops and laptops.

We construct 28 features that are used in the following experiments for predicting purchase of personal computers. Such features include “whether searched laptop keywords before purchasing on Google”, “# of sessions this user searched laptop keywords before purchasing”, “whether this user made a purchase (of any product category) in the past month” and so on (all features are listed in 5). December 2005 data is used for this experiment.

² Note that random characters sequences are removed from the results.

4 Experiments

We discuss briefly the input data, experimental design, and evaluation metrics for the classification algorithms.

Input Data for Prediction. December 2005 data is used for this experiment. We consider the 28 features as condition attributes, and whether a person is a buyer or non-buyer for personal computers as the decision attribute. For a decision table $T = (C, D)$, $C = \{28 \text{ features}\}$, $D = \{\text{buyer, non-buyer}\}$. With 83,635 users and 28 features, we create a decision table as shown in Table 1(b) as input to prediction algorithms for discovering users purchasing intent.

Experiment Design. For the complete data in the form of a decision table 83635×29 as shown in Table 1(b), we performed 10-fold cross validation through all the experiments.

Evaluation Metrics. We use the following evaluation metrics to evaluate classification performance. An individual can be classified as a buyer (denoted as P) or non-buyer (denoted as N). A classification is either correct (denoted as T) or false (denote as F). Thus, an individual who is an actual buyer but is classified as non-buyer is denoted by FN; an actual buyer and classified as a buyer is denoted as TP; an actual non-buyer but classified as buyer is denoted as FP; an actual non-buyer and classified as non-buyer is denoted as TN. Therefore, we have $Recall = \frac{TP}{TP+FN}$, $Precision = \frac{TP}{TP+FP}$, $TruePositiveRate = \frac{TP}{TP+FN}$, and $FalsePositiveRate = \frac{FP}{FP+TN}$.

4.1 Classification Experiments

In order to accomplish the prediction task, we conducted the following experiments using classification algorithms including decision trees, logistic regression and Naïve Bayes.

Decision Tree. Decision trees can be used to construct classifiers for predictions. We assume only buyer or non-buyer as the two classes in our discussion. C4.5 decision tree [9] implementation is used for classification rule generation. We obtained precision 29.47%, and recall 8.37% for decision tree learning.

Logistic Regression. We use Weka's [8] logistic regression implementation for creating the classifier. By measuring the capabilities of each of the independent variables, we can estimate the probability of a buyer or non-buyer occurrence. The default cutoff threshold of predicting a buyer is $p = 0.5$. The precision is 18.52% and recall is 2.23%. Figure 1(a) shows the precision and recall curve for the user-centric classifier generated by logistic regression.

Figure 1(b) shows the ROC curve for the user-centric classifier generated by logistic regression. Figure 2(a) shows the tradeoff between the cutoff threshold and precision/recall for the user-centric classifier generated by logistic regression. This plot can be used for determining the suggested cutoff threshold in order to reach a satisfied precision and recall towards certain classification applications.

³ Downloaded from <http://www.cs.waikato.ac.nz/ml/weka/>

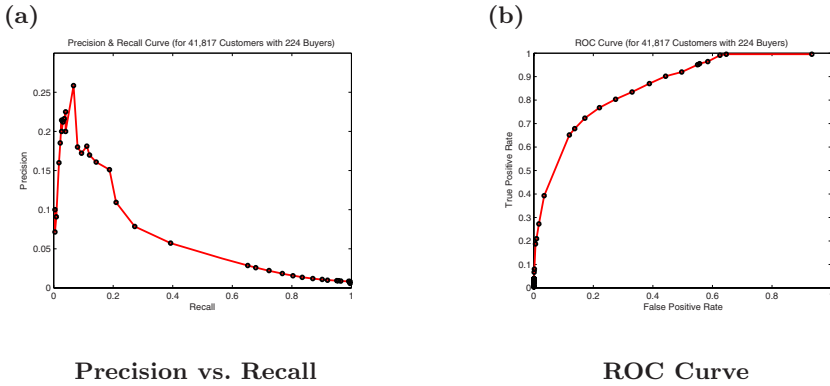


Fig. 1. Experimental Results

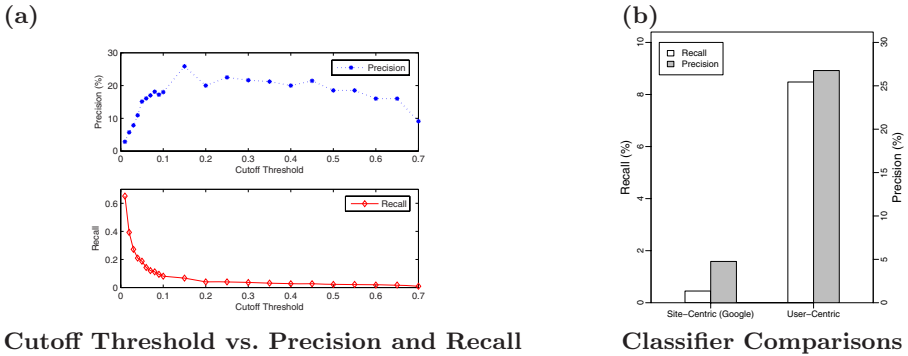


Fig. 2. Experimental Results

Naïve Bayes. Previous studies have shown that a simple Naïve Bayesian classifier has comparable classification performance with decision tree classifiers [3]. We use Weka’s Naïve Bayes classifier implementation for our experiments [11]. We obtained the classification results as precision 3.52% and recall 23.2%.

Discussions. The classification experimental results demonstrate effective product level prediction. Classifiers can be created based on user-centric features to predict the potential buyers. From our experiment on predicting product purchases, we observed that decision tree algorithm can obtain the highest prediction precision. The branching nodes in the tree splitting a potential buyer and non-buyer can be detected and used for suggesting personalized relevant content. Logistic regression can be used as a flexible option to adjust the precision and recall for the classifiers.

4.2 Site and User-Centric Comparison Experiments

To help quantify the benefits of user-centric classifiers for this task, we compare the performance of a decision tree classifier based on 28 user-centric features to

the best site-centric feature as a single classifier from a major search engine (i.e. “users who searched laptop keywords on Google before purchasing and searched more than one session”). The precisions for the user-centric and site-centric classifiers are 26.76% vs. 4.76%, and recall are 8.48% vs. 0.45%. The comparison figures is shown in Figure 2(b).

The result indicates that user-centric classifiers provide a much higher prediction precision (without loss of recall) than site-centric classifiers for predicting purchasing intent. Indeed, our discussions with industry experts indicate that even $\sim 5\%$ precision is an extremely good number in online marketing campaigns executed through search advertising. The fact that our models can increase precision, often with an increase in recall as well, demonstrates the rich value contained in user-centric data for widely applicable prediction problems.

4.3 Prediction Latencies

A key question for models of user intent is the prediction latency, defined as the period of time before the intended action that a prediction can be made. It may not be useful for many applications if good predictions can only be made over very short latent periods (e.g., a purchase prediction 10 seconds before it happens). To address this concern we performed latency experiments using November and December 2005 data. We used the feature “whether searched laptop keywords on all NNR before purchasing a personal computer”, to make predictions using SQL aggregations. The experimental results indicate that 20.15% of computer transactions can be predicted by this feature. Among these predicted transactions, only 15.59% transactions have the latent period less than one day (we call this same-day-purchase) and 39.25% transactions have 1-7 days of latent period (we call this first-week-purchase). This experiment shows that online-shopping customers usually do not just come and immediately buy. They spend some time (mostly, more than one day) doing research before their final purchase decisions, which gives time to detect purchasing interests based on behaviors, make predictions, and suggest information.

4.4 Smart Cookies

Our results indicate that useful models of intent can be learned from offline panel data and could be deployed client-side through simple classification algorithms. Client-computed outputs such as “the probability that the user will purchase product type P within the next month” could be used as intentional signals for a variety of personalization tasks such as personalizing search or serving relevant advertising in a variety of contexts. These models need not be privacy invasive. A dynamic, intentionally expressive “smart cookie” could be one mechanism to deploy our models on the client-side. Whereas browser cookies often contain simple information such as identities, etc., we imagine an implementation using models such as the ones we have demonstrated which can augment the cookie data with user-centric data. (See 5 for more details).

For example, Google now employs a feature called “web history”, which automatically collects and stores on central servers the entire clickstream of participating users. Presumably, some users would be more comfortable than others, and our methods show how to learn useful models from such data which can be deployed client-side on users who do not participate in such collection.

5 Conclusion

We demonstrated very effective product category level purchase prediction models (regardless of the site of purchase) for user-centric clickstream data. Comparison experiments show that the such models strongly outperform site-centric models, and predictions can be made ahead of time. Our models are fully automatable, and can be thought of as key enabling functionality for a “smart cookie” mechanism which could be deployed client-side and therefore would mitigate privacy concerns. It is worth noting that the baseline we established, the site-centric view of the search engine Google, was, by industry standards, quite good at predicting. Nevertheless, the user-centric models we created were able to outperform that important baseline by wide margins.

References

1. Banerjee, A., Ghosh, J.: Clickstream clustering using weighted longest common subsequences. In: Proc. of the Web Mining Workshop at the 1st SIAM Conference on Data Mining, Chicago (2001)
2. Gunduz, S., Ozsü, M.: A web page prediction model based on click-stream tree representation of user behavior. In: KDD 2003, pp. 535–540 (2003)
3. Huang, J., Lu, J., Ling, C.X.: Comparing naive bayes, decision trees, and svm with auc and accuracy. In: ICDM 2003, p. 553 (2003)
4. Li, K., Qu, W., Shen, H., Wu, D., Nanya, T.: Two cache replacement algorithms based on association rules and markov models. In: SKG, p. 28 (2005)
5. Lukose, R., Li, J., Zhou, J., Penmetsa, S.R.: Learning user purchase intent from user-centric data. Technical report, Hewlett-Packard Labs (2008)
6. Moe, W.W., Fader, P.S.: Dynamic conversion behavior at e-commerce sites. *Management Science* 50(3), 326–335 (2004)
7. Montgomery, A.L., Li, S., Srinivasan, K., Liechty, J.C.: Modeling online browsing and path analysis using clickstream data. *Marketing Science* 23(4), 579–595 (2004)
8. Padmanabhan, B., Zheng, Z., Kimbrough, S.O.: Personalization from incomplete data: what you don’t know can hurt. In: KDD 2001, pp. 154–163 (2001)
9. Quinlan, J.R.: C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco (1993)
10. Teevan, J., Dumais, S.T., Horvitz, E.: Personalizing search via automated analysis of interests and activities. In: SIGIR 2005, pp. 449–456 (2005)
11. Witten, I.H., Frank, E.: *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann Publishers Inc., San Francisco (2005)

Query Expansion for the Language Modelling Framework Using the Naïve Bayes Assumption

Laurence A.F. Park and Kotagiri Ramamohanarao

Department of Computer Science and Software Engineering,
The University of Melbourne, 3010, Australia
{lapark,rao}@csse.unimelb.edu.au
<http://www.csse.unimelb.edu.au>

Abstract. Language modelling is new form of information retrieval that is rapidly becoming the preferred choice over probabilistic and vector space models, due to the intuitiveness of the model formulation and its effectiveness. The language model assumes that all terms are independent, therefore the majority of the documents returned to the ser will be those that contain the query terms. By making this assumption, related documents that do not contain the query terms will never be found, unless the related terms are introduced into the query using a query expansion technique. Unfortunately, recent attempts at performing a query expansion using a language model have not been in-line with the language model, being complex and not intuitive to the user. In this article, we introduce a simple method of query expansion using the naïve Bayes assumption, that is in-line with the language model since it is derived from the language model. We show how to derive the query expansion term relationships using probabilistic latent semantic analysis (PLSA). Through experimentation, we show that using PLSA query expansion within the language model framework, we can provide a significant increase in precision.

Keywords: query expansion, language model, naïve Bayes.

1 Introduction

Many information retrieval systems make use of the assumption that each term is independent of each other in order to achieve fast query times and small storage. Unfortunately, this assumption also reduces the quality of retrieval. By using the assumption that every term is independent of each other term, we cause the retrieval system to disregard the term relationships. This implies that only those documents that contain the query terms will be retrieved, even if other documents are relevant to the query. Therefore, by making the term independence assumption, we are placing a large importance on the process of user query term selection. If the wrong terms are chosen for the query, the wrong documents will be retrieved.

An effective method of introducing term relationships to these fast retrieval systems is to modify the query before a search takes place. Query expansion is

a method of introducing related terms into the users query, in order to retrieve documents containing the related terms that would have otherwise not been found. Query expansion was first introduced to the vector space model [1], and later applied to the probabilistic model of information retrieval [2] due to its simplicity and effectiveness.

Language models for information retrieval [3] are a new method of information retrieval that have found recent attention due to the intuitiveness of their formulation. To date, there have been several attempts at applying query expansion to language models, but they have only caused an increase in the language model complexity.

In this article, we introduce a new form of query expansion that is derived from within the language modelling framework. We show how to use the query expansion and also how we can generate the term relationships to use for the expansion. This article makes the following contributions:

- a method of query expansion for language models, using the naïve Bayes assumption, that fits the language modelling framework (section 3.1)
- the application of PLSA term-term probabilities for query expansion in language models (section 3.2)
- an introduction to query term compensation during query expansion (section 4)

The article will proceed as follows: Section 2 will provide a brief description of language models and their use for information retrieval, section 3 describes a simple new method of query expansion for language models and provides methods of computing term-term probabilities to use within it. Section 4 shows a problem that is inherit in deriving probabilistic term relationships and provides simple methods to overcome it. Finally, section 5 contains the experimental performance of the language model query expansion and a discussion of the results.

2 Language Models for Information Retrieval

Rather than computing the relevance of a document when given a query, the language modelling approach to information retrieval is to compute the probability of a query being generated from a given document model. By assuming term independence, we are able to decompose the language modelling method into the product of query term probabilities:

$$P(Q|M_d) = \prod_{t_i \in Q} P(t_i|M_d)$$

The value $P(t_i|M_d)$ is the probability of generating the term t_i using document model M_d , therefore it is a measure of the similarity of term t_i to document d . The language modelling approach stats that every document is generated using a document model. The text within document d is sampled from the document model, based on the probability distributions within the model. Therefore, for

us to provide $P(t_i|M_d)$, we must estimate the distribution of the terms in the document model M_d . To do so, must use the term frequency values within the document collection; the only evidence that we have of the term distributions within the document model.

By simply using the term frequencies $(f_{d,t})$ to compute $P(t_i|M_d)$, we limit our probability estimations to the sampled terms within the document and we also assign a zero probability to those terms that were not sampled from the document model. This constraint is not valid, since there is a chance that there are many terms that are found in to document model M_d , but not found in this particular sample. To obtain a more global term probability, we could observe the frequency of the term in the document collection; this value provides us with a measure of the rarity of the term, but is not specific to the document. Therefore, to obtain a better approximation of the term distributions within the document model, a mixture of the document term frequency and the collection term frequency is used to compute $P(t_i|M_d)$:

$$P(t_i|M_d) = \lambda P(t_i|d) + (1 - \lambda)P(t_i|C) \quad (1)$$

where $\lambda \in [0, 1]$ is the smoothing parameter, $P(t_i|d)$ is the probability of choosing term t_i from document d .

3 Query Expansion within Language Models

The language modelling framework provides us with a method of computing the probability of a document generating a query, even if the query terms do not exist within the document. We showed in the previous section that this is possible by observing the global document collection term probability as well as the local document specific term probability.

Unfortunately, this method of term probability computation does not take into account the relationship of the term to any other term in the document set. The probabilities are computed based only on the frequency of the term itself. By ignoring term relationships, the language modelling approach will provide high probability to those queries who's terms appear in the given document and low probability to queries who's terms do not appear in the given document, regardless of the content of the document. Therefore the document retrieval process requires the user to use the . . . query terms, even though the user is likely to be unfamiliar to the requested information. As a simple example, a search for . . . will retrieve documents containing the term corn, but not the equally relevant documents containing the word . . .

In order to retrieve documents containing related term, we must be able to:

1. use the term relationships as a query expansion within the retrieval process, and
2. identify the term relationships to use as a query expansion

There have been attempts to include query expansion in the language modelling retrieval process [4,5], but they greatly increase the complexity of the model and hence negated the simplicity that makes the language modelling method desirable.

In this section we will deduce a simple method of including a query expansion within the language modelling framework by applying the naïve Bayes assumption, and we will explore two methods of computing the term relationships from the document collection.

3.1 Query Expansion Using Naïve Bayes

In order to use term relationships within the language modelling framework, we must be able to derive a model that reflects the simplicity of a language model. A query expansion process computes the set of terms that are related to the query and then uses those terms to perform the retrieval. Put into the language model framework, we compute the probability of generating the query, given the expansion terms and the document model.

To compute the set of term probabilistic relationships, we will use the document set statistics. If we choose to use the joint probability values, we would over fit our term relationships to the document set. Therefore, to generalise the relationship modelling and hence remove the over fitting, we use naïve Bayes modelling to remove the dependence of the terms on the set of documents.

To obtain the probability of generating term t_i , given term t_j and document model M_d , we use the following equation:

$$\begin{aligned}
 P(t_i|M_d) &= \sum_{t_j \in T} P(t_i, t_j|M_d) \\
 &= \sum_{t_j \in T} P(t_i|t_j, M_d)P(t_j|M_d) \\
 &= \sum_{t_j \in T} P(t_i|t_j)P(t_j|M_d)
 \end{aligned} \tag{2}$$

where $P(t_i|t_j, M_d) = P(t_i|t_j)$, using the naïve Bayes assumption that t_i and M_d are conditionally independent given t_j , and T is the set of unique terms. Using this equation, we can compute the probability of document model M_d generating term t_i from the probability of document model M_d generating term t_j and the probability of term t_i given term t_j .

Equation 2 provides us with a query expansion method for language models, where $P(t_i|t_j)$ is used to compute the relationship of each term to the query term and hence the query expansion, and $P(t_j|M_d)$ is the language model term probability shown in equation 1, which is used to compute the probability of generating the expansion terms given the document model.

Note that although we use Dirichlet smoothing throughout this article, the above derived query expansion within the language modelling framework can be used with any smoothing method.

3.2 Computing the Query Expansion

Now that we have set up a general framework for query expansion within the language modelling method of information retrieval, we will examine methods of

computing the term relationships that are needed in order to perform the query expansion. In this section, we present two forms of query expansion; the first is based on the probabilities produced using language models, and the second is based on the probabilities produced using probabilistic latent semantic analysis.

Probabilistic latent semantic based query expansion. Probabilistic latent semantic analysis (PLSA) [6] is a probabilistic method of discovering hidden topics within a document collection using maximum likelihood. Given the estimated probability of document d_i and term t_j as:

$$\hat{P}(d_i, t_j) = \frac{f_{d_i, t_j}}{\sum_{d \in D} \sum_{t \in T} f_{d, t}}$$

we want to compute the actual probability of a term and a document, given the model:

$$P(d, t) = \sum_z P(d|z)P(z)P(t|z)$$

where $P(d|z)$ and $P(t|z)$ are the probability of document d given topic z and the probability of term t given topic z respectively, and $P(z)$ is the probability of topic z .

It was recently shown that PLSA information can be used effectively as a query expansion by observing only the $P(t|z)$ and $P(z)$ values [7]. We can show:

$$\begin{aligned} P(t_i|t_j) &= \sum_{z \in Z} P(t_i, z|t_j) \\ &= \sum_{z \in Z} P(t_i|z, t_j)P(z|t_j) \\ &= \sum_{z \in Z} P(t_i|z)P(z|t_j) \\ &= \sum_{z \in Z} P(t_i|z)P(t_j|z)P(z)/P(t_j) \\ &= \frac{\sum_{z \in Z} P(t_i|z)P(t_j|z)P(z)}{\sum_{z \in Z} P(t_j|z)P(z)} \end{aligned} \quad (3)$$

where $P(t|z)$ and $P(z)$ are computed using PLSA, and $P(t_i|z, t_j) = P(t_i|z)$ using the naïve Bayes assumption that term t_i and term t_j being conditionally independent given topic z .

4 Query Term Compensation

The set of probabilities of terms T are disjoint when given term t_j . This can be seen by the property that:

$$\sum_i P(t_i|t_j) = 1$$

Table 1. PLSA query expansion within the language model framework, using PLSA add compensation with various values for the compensation factor α on the Associated Press document collection. The baseline measure (language model without query expansion) provides a MAP of 0.2749. The * and ** shows a statistically significant change in MAP at the 0.1 and 0.05 levels, compared to the language model without query expansion.

Compensation (α)	0	0.1	0.3	0.5	0.7	0.9	1
MAP	0.0669**	0.2715	0.2797	0.2803**	0.2793**	0.2788**	0.2788**

Given that $P(t_i|t_j) > 0$, we find that $P(t_i|t_j) < 1$ for all i and j , including the case where $i = j$. From this we can see that the probability of a term given itself is less than one. We may also find that the $P(t_i|t_j)$ where $i \neq j$ is greater than $P(t_i|t_i)$, implying that other terms are more related to the term than the term itself.

The effect of a term having a low probability given itself, may cause problems during a query expansion. We may find that other terms introduced from the expansion have a higher probability than the original query terms. Therefore the query terms may become lost in the expansion.

To compensate for this reduction in query term probability, we have explored the method of adding 1 to the computed probability of a term given itself. This compensation is as though we have included the original query in the query expansion, where the add method adds the expansion probability of the query terms in the expansion to the query terms.

Therefore, using the PLSA-based query expansion, we provide the following methods of compensation for the conditional probabilities:

$$\text{PLSA add: } P(t_i|t_j) = \begin{cases} \frac{\sum_a P(t_i|M_a)P(t_j|M_a)}{\sum_a P(t_j|M_a)} & \text{if } i \neq j \\ \frac{\sum_a P(t_i|M_a)P(t_j|M_a)}{\sum_a P(t_j|M_a)} + \alpha & \text{if } i = j \end{cases}$$

where α is the compensation factor, and the probability for $i \neq j$ is taken from our derivation earlier in equation 3.

5 Experiments

Our set of experiments examines PLSA query expansion using add query compensation within the language model framework on a collection of 84,678 documents from the associated press found in TREC disk 10. Experiments were performed using the values 0, 0.1, 0.3, 0.5, 0.7, 0.9 and 1 for the compensation factor (α). The results are shown in table 1.

We can see from the results that the MAP peaks at $\alpha = 0.5$ and that the results are statistically significant at the 0.05 level for larger values of α . We can also see that the result for $\alpha = 0$ is very poor. Using the add query compensation,

¹ <http://trec.nist.gov>

where $\alpha = 0$ is equivalent to using no query compensation, so we can see that it is essential to use query compensation on large and small document sets.

The significant increase in MAP shows that using PLSA query expansion with query compensation is a useful addition when used within the language model framework.

6 Conclusion

Within the field of information retrieval, language models have shown to be competitive with other models of retrieval, while offering an intuitive and simple formulation. To simplify the model, language models include the assumption that all terms are independent. This assumption places great importance on the user's choice of query terms. To introduce term relationships into the language modelling framework others have applied query expansion, but the complexity of the expansion removed the simplicity from the language model formulation.

In this article, we introduced a method of query expansion for language models which uses the naïve Bayes assumption to produce generalised probabilistic term relationships. To compute the term relationships, we examined a probabilistic latent semantic analysis (PLSA) method. Experiments on a document collection showed us that the the PLSA query expansion within the language modelling framework provided a significant increase in precision over the language model with no expansion. Therefore the PLSA query expansion was also effective for larger document sets.

References

1. Buckley, C., Salton, G., Allan, J., Singhal, A.: Automatic query expansion using SMART: TREC 3. In: Harman, D. (ed.) *The Third Text REtrieval Conference (TREC-3)*, Gaithersburg, Md. 20899, National Institute of Standards and Technology Special Publication 500-226, pp. 69–80 (1994)
2. Robertson, S.E., Walker, S.: Okapi/keenbow at TREC-8. In: Voorhees, E.M., Harman, D.K. (eds.) *The Eighth Text REtrieval Conference (TREC-8)*, Gaithersburg, Md. 20899, National Institute of Standards and Technology Special Publication 500-246, Department of Commerce, National Institute of Standards and Technology, pp. 151–162 (1999)
3. Ponte, J.M., Croft, W.B.: A language modeling approach to information retrieval. In: *SIGIR 1998: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 275–281. ACM Press, New York (1998)
4. Cao, G., Nie, J.Y., Bai, J.: Integrating word relationships into language models. In: *SIGIR 2005: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 298–305. ACM Press, New York (2005)

5. Bai, J., Song, D., Bruza, P., Nie, J.Y., Cao, G.: Query expansion using term relationships in language models for information retrieval. In: CIKM 2005: Proceedings of the 14th ACM international conference on Information and knowledge management, pp. 688–695. ACM Press, New York (2005)
6. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pp. 50–57. ACM Press, New York (1999)
7. Park, L.A.F., Ramamohanarao, K.: Query expansion using a collection dependent probabilistic latent semantic thesaurus. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 224–235. Springer, Heidelberg (2007)

Fast Online Estimation of the Joint Probability Distribution

J.P. Patist

Vrije Universiteit Amsterdam, Computer Science,
De Boelelaan 1081a, 1081 HV Amsterdam, The Netherlands
jpp@cs.vu.nl

Abstract. In this paper we propose an algorithm for the on-line maintenance of the joint probability distribution of a data stream. The joint probability distribution is modeled by a mixture of low dependence Bayesian networks, and maintained by an on-line EM-algorithm. Modeling the joint probability function by a mixture of low dependence Bayesian networks is motivated by two key observations. First, the probability distribution can be maintained with time cost linear in the number of data points and constant time per data point. Whereas other methods like Bayesian networks have polynomial time complexity. Secondly, looking at the literature there is empirical indication [1] that mixtures of Naive-Bayes structures can model the data as accurate as Bayesian networks. In this paper we relax the constraints of the mixture model of Naive-Bayes structures to that of the mixture models of arbitrary low dependence structures. Furthermore we propose an on-line algorithm for the maintenance of a mixture model of arbitrary Bayesian networks. We empirically show that speed-up is achieved with no decrease in performance.

1 Introduction

In recent years, the emergence of applications involving massive data sets such as customer click streams, telephone records, multimedia data, has resulted in extensive research on analyzing data streams within the field of data mining [2]. The computational analysis of data streams is constrained by the limited amount of memory and time.

Furthermore the process, which generates the data stream, is changing over time. Consequently the analysis should cope with changes in the data distribution.

Probability distribution estimation is used widely as a component of descriptive analysis, outlier -and change detection, etc. Thus, it is a necessarily component in almost any data stream analysis system. Popular models used for representing the joint probability distribution are Bayesian networks, dependency models, mixture models, markov models and wavelets.

In this paper we restrict ourselves to the following models: dependency models, Bayesian networks and mixture of Bayesian networks. The optimal dependency tree can be found efficiently using the The Chow-Liu [3] algorithm. Bayesian

network exhibit more expressive power than dependency tree, but learning is more expensive. So, on the one hand, we have a simple model which can be maintained efficiently and on other hand a powerful model but with larger maintenance cost. Our solution, namely mixture models, provides a powerful model and a less demanding estimation procedure. The parameters are learned by an online EM-algorithm and the structure is adjusted by randomly adding components and dependencies.

The paper is structured as follows. First we report on related work on online EM-algorithms and joint probability distribution estimation. Hereafter we define the problem as well the batch and online algorithm. Experiments on synthetic data show the performance of the system compared to two baseline models, namely dependency trees and Bayesian networks.

2 Related Work

In reporting on related work we restrict ourselves to the topics of the estimation of the joint probability distribution in the context of data stream mining and on-line or one-pass versions of the EM-(Expectation Maximization) algorithm [4].

A major part of the research on on-line versions of the EM-algorithm like [5] is based on stochastic approximation and can be seen as special types of the Robbins-Monro algorithm.

In [6] a one-pass EM algorithm is proposed for the estimation of Gaussian mixtures in large data sets. Data points are compressed into sufficient statistics or discarded based on their clusteriness. The method seems scalable in the number of records and more accurate than sampling.

In [7] a method is proposed to sequentially update the dependency structure of a Bayesian network. The Bayesian network is updated using a search buffer for possible neighborhood structures.

In [8] a density estimation procedure is proposed based on kernel estimation. A buffer is maintained in which kernels over data point are stored. The buffer is compressed whenever it reached the maximum buffer size. The procedure has linear time complexity.

In [9] the wavelet density estimation technique is adapted to the context of data streams. The wavelet density estimators require fixed amount of memory and is updated in an on-line manner.

In [11] the joint probability distribution is estimated using mixtures of Naive-Bayes models. The Naive-Bayes basis model is a Bayesian network with the constraint of independence between all 'non-target' variables given the 'target' variable. The mixture model was comparable in accuracy to Bayesian networks.

3 Problem Description

We define a data stream s as a possibly infinite sequence of random variables X_i^d, \dots, X_n^d from the discrete nominal domain N^d . The objective is to have an any time up-to-date accurate estimate of the underlying joint probability

distribution of s . Furthermore, the estimation is constrained by time and space and must have low time and space complexity.

We translate this problem to finding the probability mass function F which maximized $L(X^t, F)$ indexed by t . The likelihood function L is defined as: $L(X_{i..n}, F) = \prod_i^n F^{t=i}(X_i)$.

In this paper, Bayesian networks, mixture models, dependency trees and mixture of independence models constitute for F . In the on-line setting we want to optimize $L(X_{i..n}, F^t)$, the likelihood over the sequence X_i, \dots, X_n , where F^t is the probability mass function at time t .

4 Mixture Models

A mixture model is a convex combination of k probability mass function: $P(X_i) = \sum_{s=1}^k \alpha_s P_s(X_i)$. A d -dimensional mixture of arbitrary Bayesian networks is a probability mass function on N^d that is given by a convex combination of k Bayesian networks: $P_s(X_i) = \prod_j^d P_s(X_i^j | \text{parents}(X_i^j))$, where $\text{parents} \subset \cup_d X^d$ and the underlying dependency structure of the data distribution does not contain loops. The Likelihood of a data record i given a Bayesian network is equal to the product of local probabilities. X_i^j is the j -th attribute of record i . Note that dependency trees are Bayesian models only with an extra constraint on the dependency structure.

Given a set $D = \{X_1, \dots, X_n\}$ of independent and identically distributed samples from $P(X)$, the learning task is to estimate the parameter vector $\Theta = \{\alpha_s, CPT_{k,i}\}_{s=1}^k$ components that maximizes the log-likelihood function $L(\Theta; D) = \sum_{i=1}^k \log P(X_i)$. Maximization of the data log-likelihood $L(\Theta)$ is usually achieved by running the EM-algorithm. The standard batch-EM algorithm starts with parameters set to some initial values, and then iteratively repeats two steps (the E-step and the M-step) trying to improve the value of $L(\Theta)$ by adjusting. It terminates after a pre-specified number of iterations, or when the improvement rate drops below a certain threshold. In the E-step, the EM-algorithm finds the contributions $q_i(s)$ of the points X_i to all mixture components, $q_i(s) = p(s_j | X_i)$.

4.1 Estimating Mixture Models Using On-Line EM

In [10] improvements are investigated such as block-EM. In block-EM both the E and M-step are performed over blocks of data. Per block the sufficient statistics are stored and replaced when it is used to update the model. This enables the model to incorporate information faster.

In this paper we extend the work of [10] by exponentially weighting sufficient statistics. We update the model after observing a buffer of b data points. Then the formulas become: $S_{b, \alpha_k}^{(t+1)} = \sum_{x \in b} P(s_k | x^{t+1})$, $S_{b, \theta_{v,k}^{(t+1)}} = \sum_{x \in b} P_v(s_k | x_v^{t+1})$ and $\alpha_{k,b} = (1 - \lambda)\alpha_{k,t} + \lambda S_{\alpha_{b,k}}^{t+1}$, $\theta_{k,b} = (1 - \beta)\theta_{k,t} + \beta S_{\theta_{b,k}}^{t+1}$. S_α, S_θ are sufficient statistics for α and θ (=CPT).

The advantage of updating the model after b points is two-fold. Firstly, it saves computational effort by not having to update the model parameters after each point, but only after each b points. Secondly, because of the more reliable estimates of the expectations improvement of the performance is more stable and faster.

4.2 Adapting Model Structure

To facilitate faster adaptation to changes in the true data distribution we use four different techniques: adding and deleting components and, adding and deleting dependencies in basis models. Components are added by probability dependent on the number of components in the mixture model and is bounded by a maximum. In practice the formula equals: $P(\text{add component} | n_{\text{comp}}=n) = \eta(1 - \frac{n}{n_{\text{max}}})$. It follows that if $n = n_{\text{max}}$, the probability of adding a component is 0. The new component is a full-independence model with uniform random parameters. The prior of the new component is set equal to prior α_{min} . The priors of the remaining basis models are adjusted relatively, such that $\sum \alpha_i = 1$.

A Component is deleted when its prior drops below some threshold t . To ensure that components are not deleted immediately after insertion, deletion is constrained by min_{age} . After deletion the priors α . are normalized.

The structure of individual basis models is changed by removing and adding dependencies in the Bayesian model. Dependencies are added randomly from the set of eligible dependencies. The maximal number of parents is 1. The probabilities of the new dependency is set such the marginal distribution is equal the distribution before adding. Assuming dependencies which are not supported by the data can be harmful.

We represent every conditional probability table (CPT) by a mixture model of two components, $CPT_{X|Y} = P(X|Y) = \lambda \hat{P}(X) + (1 - \lambda) \hat{P}(X|Y)$. This mixture representation is only used in the on-line mixture model. Parameters are estimated using the EM-algorithm. We interchange $P(X|Y)$ by $P(X)$, when $\lambda > h$. This corresponds to the deletion of the dependency $X|Y$, the prior is set to zero and the dependency deleted.

4.3 Bayesian Networks and Dependency Trees

Dependency trees and Bayesian networks are use as a reference to our methods. Learning in Bayesian networks is divided into two tasks: structure learning and parameter estimation. Structure learning is the learning of the optimal topology of the network. All parameters can be estimated by determining frequencies of combinations of variable values. Examples of structure learning algorithms are: *K2*, *MCMC*, etc.. We use the *K2* algorithm [1]. Dependency trees can be constructed more efficiently than Bayesian networks. A dependency tree is a Bayesian network with the extra constraint that each variable has only one parent. The optimal tree can be constructed in polynomial time [3].

4.4 Time Complexity

The complexity of the Chow-Liu algorithm is $O(nd^2 + d^2c^2)$, of K2 it is $O(n \max_p^2 d^2c)$, Batch EM for mixture of Bayesian networks: $O(iknd + \frac{ikcnd}{b})$ and On-line EM for mixture of Bayesian networks: $O(knd + \frac{kcnnd}{b})$. The variables n , d , i , c , max_p correspond to the number of data points, the dimensionality, the number of iterations of the EM-algorithm, the maximum number of values per variable and the maximum number of parents.

The advantage of the on-line EM-algorithm is the spread of the computational effort over the data points, resulting in an $O(kdc/b)$ average time complexity per data point. The parameter b corresponds to the number of data points between two model updates. In the experiments $b = 50$.

5 Experiments

In the first experiment we investigate the performance of the different methods on different kind of stationary artificial data sets. In the second experiment we explore the performance on dynamic data.

The models that are compared are: dependency trees, Bayesian networks and the proposed on-line EM-algorithm for arbitrary low complexity Bayesian networks as well as the batch EM variant. Performance is measured by the log likelihood on a test set.

We generated 50 different data sets differing in cardinality, complexity and dimensionality. The cardinality, the number of different values per variable is, 2, 5 and 10 values and equal for all variables. The complexity of the underlying Bayesian models is expressed in the number of dependencies. The number of dependencies in the experiments is set to 1, 3 and 5. If the complexity is 1, a variable has one parent, if no loops are present, consequently when 3 then three parents. The adjacency matrix and the probabilities of the conditional probability tables are generated at random.

The Bayesian networks are build using the Murphy toolbox [12] for Matlab using K2. The K2-algorithm was constraint by a maximum of 3 parents. We called the algorithm 10 times with different order on the variables and selected the network with the highest log likelihood on the training set.

In case of the batch-EM algorithm, the algorithm is stopped when 100 iterations of E and M-steps are performed or when the log likelihood did not improve significantly. The structure as well the number of the basis models is fixed after initialization.

The online-EM algorithm is initialized in the same way as the batch variant. Note that in the case of the online variant we use a basis model which represents the conditional probability table as a mixture of the full probability table and the marginal probabilities. The online-EM calculates sufficient statistics over blocks of 50 data points. Every 100 data points components and dependencies are pruned and added. Components are pruned when the $prior < \frac{1}{50k}$. Components are added by probability $p = (ncomp_{max} - ncomp_{model}) / (2 \cdot ncomp_{max})$. The

learning rate is uniformly decreased from 1 to 0.01. Dependencies are removed when the prior of the marginal distribution is larger than 0.5. Dependencies are added at random when possible, thus not creating loops and not exceeding the maximum number of parents.

In the experiments the performance is determined using different sizes of data streams and different number of components. The number of components used are: 5, 10, 15 and 20 components and the sizes are: 1.000, 10.000, 20.000, 30.000, 40.000, 50.000 data points. The data dimensionality is 10 or 20.

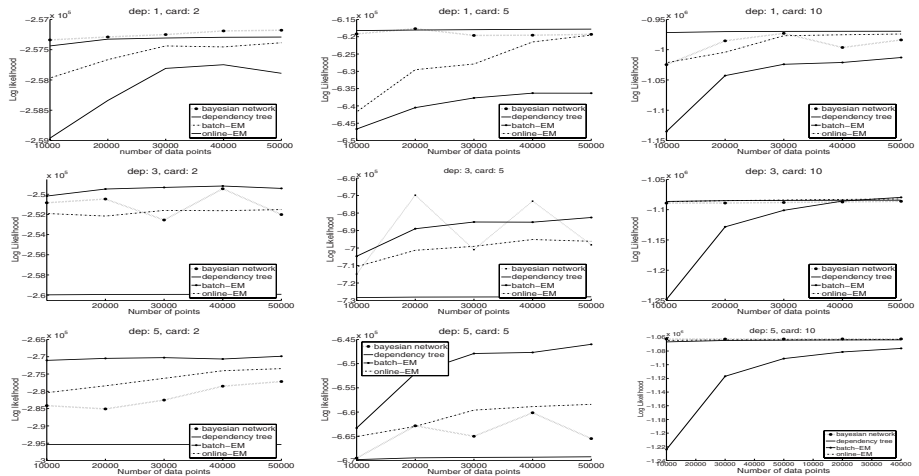


Fig. 1. The performance of Bayesian networks, dependency trees and mixture models using batch EM an on-line EM on 9 different data sets as a function of different data sizes. The data sets have 10 dimensions and differed on the number of different values per variable and the complexity of the underlying Bayesian network. The mixture models contained 20 basis model components.

6 Results

In Figure 1. is shown the typical behavior of the different algorithms on the data stream. We generated more data sets than shown, however these figures show the typical behavior. The figure shows the results of the mixture models with 20 components. A common picture, except for batch EM, is, the more components the better the performance. In case of batch-EM, the more components the slower the improvement. The online-EM did not seem to suffer from this. The batch EM-variant performed best on data streams with complexity of at least 3, on which the on-line variant and Bayesian networks are comparable.

Whenever the cardinality of the variables was 10 the 'convergence' of the basic-EM was relatively slow. This seems to hold for all complexity data sets with cardinality of 10. Both mixture models seem to perform relatively worst in

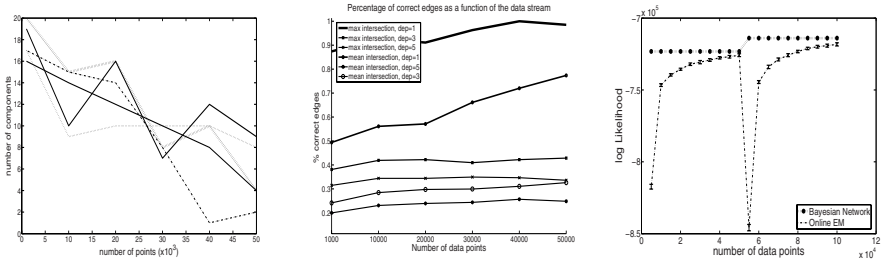


Fig. 2. From left to right: 1. The number of components as a function of the size of the data stream. The data is generated from a dependency tree. 2. The maximum and mean number of correct dependencies per component in the case of 3 different data streams. 3. The Log Likelihood as a function of a changing data stream as well the standard deviation over 10 runs initialized by 10 different structures.

the case complexity 1. The online variant approaches the highest accuracy when the cardinality is 5 and 10 and complexity is 1. This is due random search over the dependencies.

In Figure 2.1. is shown the number of components as a function of the data stream from data sets with complexity 1. There is a clear linear trend in the number of components and the size of the data stream. Probably the priors of the better fitting mixture components are growing faster than others.

In Figure 2.2. is shown the number of correctly estimated dependencies, we see an large improvement when the complexity is 1. When the number of correct dependencies is one, every basis component has the same dependency structure as the underlying Bayesian network. We plotted the maximum and average number of correct dependencies over the total number of basis models. The maximum approaches almost 1, and the average to 75 percent correct.

In the second experiment the data distribution is changed abruptly at data point 50.000. As we can see in Figure 2.3. the Log Likelihood drops and recovers. The top line is the performance of a Bayesian network build on 50.000 data points. The data is generated from Bayesian networks of complexity 2.

7 Conclusion and Future Work

We proposed a fast online-EM algorithm for the mixture of arbitrary Bayesian networks. The method iteratively changes the structure of the basis components in search of better structures. It outperforms the batch variant with respect to speed and in some cases improves on accuracy. Our method is comparable to the Bayesian network structure finding algorithm K2. When the dimensionality increases the Bayesian network is outperforming both methods. Our method outperforms the other methods with respect to speed in case the number of components are of reasonable size. In case of dynamic data streams the online-EM algorithm recovers from a change in the underlying data stream.

Random structure does not always effectively search the space. If the space grows the usefulness of random search will decrease. Thus, possible future work is a more directed randomized search procedure at the same computational cost, constraining the space of structures or using the Chow-Liu algorithm.

References

1. Lowd, D., Domingos, P.: Naive bayes model for probability estimation. In: Twenty-Second International Conference on Machine Learning, pp. 529–536 (2005)
2. Aggarwal, C.: Data Streams: Models and Algorithms. Springer, Heidelberg (2007)
3. Chow, C.K., Liu, C.N.: Approximating discrete probability distributions with dependence trees. *Transactions on Information Theory*, 462–467 (1968)
4. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 1–38 (1977)
5. Sato, M.A., Ishii, S.: On-line EM algorithm for the normalized gaussian network. *Neural Computation* 12(2), 407–432 (1999)
6. Bradley, P., Fayyad, U., Reina, C.: Scaling EM(expectation maximization) clustering to large databases. In: Technical Report MSR-TR-98-35, Microsoft Research (1998)
7. Friedman, N., Greiger, D., Goldszmidt, M.: Bayesian network classifiers. *Machine Learning*, 103–130 (1997)
8. Zhou, A., Cai, Z., Wei, L., Qian, W.: M-kernel merging: Towards density estimation over data streams. In: DASFAA 2003: Proceedings of the Eighth International Conference on Database Systems for Advanced Applications, pp. 285–292. IEEE Computer Society, Washington (2003)
9. Heinz, C., Seeger, B.: Wavelet density estimators over data streams. In: The 20th Annual ACM Symposium on Applied Computing (2005)
10. Thiesson, B., Meek, C., Heckerman, D.: Accelerating em for large databases. *Machine Learning* 45(3), 279–299 (2001)
11. Cooper, G., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 309–347 (1992)
12. Murphy (2004), <http://www.cs.ubc.ca/~murphyk/software/bnt/bnt.html>

Fast k Most Similar Neighbor Classifier for Mixed Data Based on Approximating and Eliminating

Selene Hernández-Rodríguez, J. Ariel Carrasco-Ochoa, and J. Fco. Martínez-Trinidad

Computer Science Department
National Institute of Astrophysics, Optics and Electronics
Luis Enrique Erro No. 1, Sta. María Tonantzintla, Puebla, CP: 72840, México
{selehdez, ariel, fmartine}@inaoep.mx

Abstract. The k nearest neighbor (k - NN) classifier has been a widely used non-parametric technique in Pattern Recognition. In order to decide the class of a new prototype, the k - NN classifier performs an exhaustive comparison between the prototype to classify (query) and the prototypes in the training set T . However, when T is large, the exhaustive comparison is expensive. To avoid this problem, many fast k - NN algorithms have been developed. Some of these algorithms are based on Approximating-Eliminating search. In this case, the Approximating and Eliminating steps rely on the triangle inequality. However, in soft sciences, the prototypes are usually described by qualitative and quantitative features (mixed data), and sometimes the comparison function does not satisfy the triangle inequality. Therefore, in this work, a fast k most similar neighbour classifier for mixed data (AEMD) is presented. This classifier consists of two phases. In the first phase, a binary similarity matrix among the prototypes in T is stored. In the second phase, new Approximating and Eliminating steps, which are not based on the triangle inequality, are presented. The proposed classifier is compared against other fast k - NN algorithms, which are adapted to work with mixed data. Some experiments with real datasets are presented.

Keywords: Nearest Neighbors Rule, Fast Nearest Neighbor Search, Mixed Data, Approximating Eliminating search algorithms.

1 Introduction

The k - NN [1] rule has been a widely used nonparametric technique in Pattern Recognition. However, in some applications, the exhaustive comparison between the new prototype to classify and the prototypes in the training set T becomes impractical. Therefore, many fast k - NN classifiers have been designed to avoid this problem.

Some of these fast k - NN algorithms can be classified as exact methods, because they find the same NN that would be found using the exhaustive search. Some other algorithms are approximate methods, because they do not guarantee to find the NN to a query prototype among the training set, but they find an approximation faster than the exact methods.

To avoid comparisons between prototypes during the search of the NN , different techniques have been developed: Approximating Eliminating algorithms [2-5],

Tree-based algorithms [4,6-8]. In particular, in this work, the proposed algorithm is based on an Approximating Eliminating approach.

One of the first approaches that uses approximating and eliminating steps is AESA (Approximating Eliminating Search Algorithm), proposed by Vidal [2]. In a preprocessing phase, this algorithm creates a matrix of distances between the prototypes in the training set. Given a new prototype Q to classify; a new candidate is approximated, compared against Q and, supported on the triangle inequality, those prototypes that can not be closer than the current NN are eliminated from the set T . The process finishes when all prototypes in T have been compared or eliminated.

Using AESA, good results have been obtained. However, a drawback of AESA is its quadratic memory space requirements. For this reason, in [3] an improvement (LAESA), which requires linear memory space, is proposed (LAESA). LAESA algorithm is focused on reducing the amount of information stored, but this algorithm increases the number of comparisons between prototypes. In [5] an improvement on the Approximation step is proposed, for approximating a better candidate and, therefore reducing the number of comparisons between prototypes even more than AESA.

AESA, LAESA and iAESA are exact methods to find the k - NN . However, in [5] a probabilistic approach [9] to find approximate k NN 's is also proposed. In order to reduce the amount of work, the search is stopped when certain percentage of the database has been evaluated, this method is called Probabilistic iAESA.

In [4] TLAESA algorithm is proposed. In TLAESA, a binary tree and a matrix of distances between the prototypes in T and a subset of T , are used. In [10] an improvement of TLAESA is presented.

All these methods based on Approximating and Eliminating search, were designed to work with quantitative data when the prototype comparison function satisfies the triangle inequality. However, in soft sciences as Medicine, Geology, Sociology, etc., the prototypes are described by quantitative and qualitative features (mixed data). In these cases, sometimes the comparison function for mixed data does not satisfy the triangle inequality and therefore, we can not use most of the methods proposed for quantitative prototype descriptions. Therefore, in this paper we introduce a fast approximate k most similar neighbor (k - MSN) classifier for mixed data, based on new Approximating and Eliminating steps, which are not based on the triangle inequality property of the comparison function.

This paper is organized as follows: in Section 2 the comparison function used in this work is described. In Section 3 our fast k - MSN classifier (AEMD) is introduced. Finally, we report experimental results (Section 4) and conclusions (Section 5).

2 Comparison Functions for Mixed Data

In this work, in order to compare prototypes described by mixed data, the function F [11], which does not fulfil the triangle inequality, was used. Let us consider a set of prototypes $\{P_1, P_2, \dots, P_N\}$, each of them described by d attributes $\{x_1, x_2, \dots, x_d\}$. Each feature could be quantitative or qualitative. The function F is defined as follows:

$$F(P_1, P_2) = 1 - \frac{|\{x_i \mid C_i(x_i(P_1), x_i(P_2)) = 1\}|}{d} \quad (1)$$

For qualitative data $C_i(x_i(P_1), x_i(P_2))$ is defined as follows:

$$C_i(x_i(P_1), x_i(P_2)) = \begin{cases} 1 & \text{If } x_i(P_1) = x_i(P_2) \text{ and neither } x_i(P_1) \text{ nor } x_i(P_2) \\ & \text{is a missing value} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

For quantitative data $C_i(x_i(P_1), x_i(P_2))$ is defined as follows:

$$C_i(x_i(P_1), x_i(P_2)) = \begin{cases} 1 & \text{If } |x_i(P_1) - x_i(P_2)| < \sigma_i \text{ and neither } x_i(P_1) \text{ nor } x_i(P_2) \\ & \text{is a missing value} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

Where, σ_i is the standard deviation of the attribute x_i . Using the function F , the most similar neighbor (*MSN*) of a prototype P , is the one that minimizes the function.

3 Proposed Classifier

In this section, an approximate fast k -*MSN* classifier, which considers prototypes described by mixed data, is introduced. The classifier consists of two phases: preprocessing and classification.

3.1 Preprocessing Phase

In this phase, AEDM computes the following:

1. *Similarity Matrix (SM)*. In this work, we proposed to compute and store an array of similarity per attribute among the prototypes in the training set (T), where $SM[P_a, P_b, x_i] = 1$ if, according to certain criterion, we can conclude that the prototypes P_a and P_b are similar considering the attribute x_i and $SM[P_a, P_b, x_i] = 0$, in other case; $a, b \in [1, N]$ and $i \in [1, d]$ (see figure 1). In this work, the similarity criterion described in Section 2, was used.

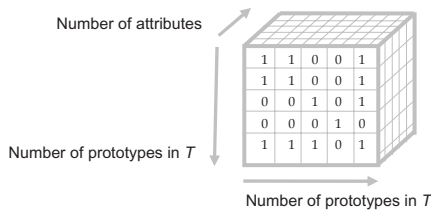


Fig. 1. *SM* matrix

The required space to store *SM* matrix is $N \times N \times d$ but each element is a bit, therefore, the needed space is $N \times N$ words of d bits.

2. *A representative prototype per class (RP_c)*. In order to obtain a first approximation during the classification phase, we propose to use a representative prototype per class, taking advantage of the class information. To compute RP_c , let $Class_c$ be the set of

prototypes in T , which belong to the class c . Then, for each prototype $P_a \in Class_c$, the following function is computed:

$$AvgSim(P_a) = \frac{\sum_{b=1, b \neq a}^{|Class_c|} F(P_a, P_b)}{|Class_c|} \tag{4}$$

$AvgSim$ evaluates the average of similarity between a fixed prototype (P_a) and the rest of the prototypes that belong to the same class. Thus, the representative prototype for class c (RP_c) is the most similar on average (or the one that minimizes $AvgSim$ function):

$$RP_c = Argmin(AvgSim(P_a)), \quad \forall a \in [1, |Class_c|] \tag{5}$$

This process is repeated for every $c \in [1, C]$, where C is the number of classes in the training set.

3. *Similarity threshold between prototypes (SimThres)*. The average value of the similarity between the prototypes belonging to the same class in T , is used as a confidence threshold to make decisions during the classification phase. This value can be a parameter given by the user. However, in this section three options to compute the confidence threshold are proposed.

To define the similarity threshold for each class c , the average of similarity, among the prototypes belonging to the same class, is computed as follows:

$$AvgValueClass_c = \frac{\sum_{a=1}^{|Class_c|} \frac{\sum_{b=1, a \neq b}^{|Class_c|} F(P_a, P_b)}{|Class_c| - 1}}{|Class_c| - 1}, \quad \forall a, b \in [1, |Class_c|] \tag{6}$$

Finally, the similarity threshold is selected following (7), (8) and (9):

$$SimThres = SimMin = Argmin(AvgValueClass_c), \quad \forall c \in [1, C] \tag{7}$$

$$SimThres = SimAvg = \frac{\sum_{c=1}^C AvgValueClass_c}{C}, \quad \forall c \in [1, C] \tag{8}$$

$$SimThres = SimMax = Argmax(AvgValueClass_c), \quad \forall c \in [1, C] \tag{9}$$

3.2 Classification Phase

Given a new prototype Q to classify, SM , RP_c and $SimThres$, computed during the preprocessing phase, are used to avoid comparisons among prototypes. The classification phase of the proposed algorithm (AEMD) is based on Approximating and Eliminating steps, which are not based on the triangle inequality.

Initial approximation step. At the beginning of the algorithm, the prototype Q is compared against the class representative prototypes to obtain a first approximation to the most similar prototype MSN and its similarity value S_{MSN} .

$$MSN = \text{ArgMin}(F(Q, RP_c)), \forall c \in [1, C] \quad (10)$$

The current MSN is eliminated from the set T . If $S_{MSN} \geq \text{SimThres}$ (where SimThres is a confidence value of similarity between prototypes belonging to the same class in T), the prototype MSN is used to eliminate prototypes from T (*Eliminating step*).

Eliminating step. In this step, given a fixed prototype (MSN) to eliminate prototypes from T , a binary representation (BR) contains the similarity per attribute, between Q and MSN is created as follows:

$$BR_i(Q, MSN) = C_i(x_i(Q), x_i(MSN)), \forall i \in [1, d] \quad (11)$$

Thus, $BR_i(Q, MSN)=1$ if Q and MSN are similar in the attribute x_i and $BR_i(Q, MSN)=0$, in other case. Using BR , those prototypes in T , which are not similar to MSN at least, in the same attributes in which MSN is similar to Q , are eliminated from T (using $SM(MSN, P_a), \forall P_a \in T$).

For example, supposed that P_0, P_1, Q and MSN , are such that $BR(Q, MSN) = [1,1,0,1,1,1,0,0]$, $SM(MSN, P_0)=[1,1,1,1,1,1,0,1]$ and $SM(MSN, P_1)=[1,0,0,0,0,1,0,1]$. Then, according to this criterion, P_0 is not eliminated because is similar to MSN in the same attributes, where MSN is similar to Q (attributes 1, 2, 4, 5 and 6). But P_1 is eliminated, without have explicitly compared it to Q , because P_1 is not similar in the same attributes, where MSN is similar to Q (MSN is similar to Q in attribute 2, but P_1 is not similar to MSN in this attribute). The similarity per attribute between MSN and P_0 ($SM(MSN, P_0)$) and the similarity per attribute between MSN and P_1 ($SM(MSN, P_1)$) are known (because these similarities were computed in the preprocessing phase). The similarity between MSN and Q , has already been computed.

After the *Initial approximation* and the *Eliminating* steps, if T is not empty, the approximation step is performed.

Approximating step. In this step, a new prototype $MSN' \in T$ is randomly selected, compared against Q ($S_{MSN'}$), eliminated from T and used to update the current MSN . If $S_{MSN'} < \text{SimThres}$ a new MSN' is randomly selected (*Approximating step*). Otherwise, if $S_{MSN'} \geq \text{SimThres}$ (where SimThres is a confidence value of similarity between prototypes belonging to the same class in T), the prototype MSN' is used to eliminate prototypes from T (*Eliminating step*). This process is repeated until the set T is empty. After finding the MSN , its class is assigned to Q .

4 Experimental Results

In this section, the performance of the proposed classifier (AEMD) is evaluated. In order to compared AEMD; the exhaustive search [1], AESA [2], LAESA ($|BPI|=20\%$ of the objects in the dataset) [3], iAESA [5], Probabilistic iAESA (using 70% as percentage threshold of the data set) [5], TLAESA [4] and modified TLAESA [10] algorithms were evaluated using the same comparison function (F , described in Section 2) instead of using a metric. To do the experiments, 10 datasets from the UCI repository

[12] were used (Mixed datasets: Hepatitis, Zoo, Flag and Echocardiogram. Qualitative datasets: Hayes, Bridges and Soybean-large. Quantitative: Glass, Iris and Wine).

In order to compare the different classifiers, the accuracy (*Acc*) and the percentage of comparisons between prototypes (*Comp*), were considered. The accuracy was computed as follows:

$$Acc = \frac{NoCorrectObj}{NoTestObj} \tag{12}$$

Where, *NoCorrectObj* is the number of correctly classified prototypes in the test set and *NoTestObj* is the size of the test set. The percentage of comparisons between objects was computed as follows:

$$Comp = \frac{NoCompFastClass * 100}{NoTrainingObj} \tag{13}$$

Where, *NoCompFastClass* is the number of comparisons done by the fast classifier, and *NoTrainingObj* is the size of the training set. According to (13), for the exhaustive classifier, the 100 % of the comparisons is done. In all the experiments, *k=1* in *k-MSN*, was used.

As first experiment, the proposed algorithm (AEMD) was evaluated. To use AEMD algorithm, *SimThres*, which corresponds to a confidence value of similarity, was tested with the values *SimThres* = 40, 60 and 80 (see table 1).

Table 1. Obtained results using AEMD, according to different values of *SimThres*

Datasets	Exhaustive <i>k</i> -NN classifier		AEMD (<i>SimThres</i> =40%)		AEMD (<i>SimThres</i> =60%)		AEMD (<i>SimThres</i> =80%)	
	Acc	Comp	Acc	Comp	Acc	Comp	Acc	Comp
Hepatitis	81,66	100	74,65	7,45	80,65	17,37	81,63	23,17
Zoo	96,00	100	78,42	8,83	94,01	9,63	95,30	28,80
Flag	54,67	100	45,15	4,11	53,85	7,62	52,08	13,11
Echocardiogram	82,44	100	80,15	7,44	81,06	9,04	81,70	25,23
Hayes	81,24	100	78,23	8,19	80,21	12,52	81,05	18,57
Soybean-large	85,40	100	65,74	7,32	84,12	8,65	83,07	8,11
Bridges	57,85	100	38,52	3,55	53,54	9,20	56,78	11,19
Glass	68,26	100	62,58	9,11	67,35	13,66	67,90	16,45
Iris	93,30	100	45,52	8,50	91,01	10,29	93,30	12,99
Wine	90,90	100	58,42	7,17	90,90	13,58	89,63	13,18
General average	79,17	100	62,74	7,17	77,67	11,16	78,24	17,08

As we can see from table 1, the bigger the value of *SimThres*, the higher the obtained accuracy. However, the percentage of comparisons is also increased. Besides, for some datasets (Echocardiogram and Hayes), good results were obtained using *SimThres*=40, while for other datasets (Flag and Soybean-large) good results are obtained using *SimThres*=60. From these results, we can not conclude an optimal value for *SimThres*. For this reason, the criteria, described in section 3.1, to establish a value for *SimThres*, were used. Thus, AEMD algorithm was evaluated with *SimThres*=*SimMin*, *SimAvg* and *SimMax* (see table 2). From table 2, we can observe that using *SimThres*=*SimAvg*, good results are obtained, for all the datasets.

Table 2. Obtained results using AEMD, according to different values of *SimThres*

Dataset	Exhaustive k -NN search		AEMD ($SimThres=SimMin$)		AEMD ($SimThres=SimAvg$)		AEMD ($SimThres=SimMax$)	
	Acc	Comp	Acc	Comp	Acc	Comp	Acc	Comp
Hepatitis	81,66	100	80,03	11,95	81,63	13,52	81,66	82,46
Zoo	96,00	100	94,10	9,31	94,00	24,76	96,00	91,00
Flag	54,67	100	52,60	16,87	52,05	17,41	54,67	56,32
Echocardiogram	82,44	100	81,05	13,56	81,70	18,50	82,44	85,60
Hayes	81,24	100	81,16	13,80	81,05	12,97	81,24	79,40
Soybean-large	85,40	100	84,21	15,70	83,07	23,46	85,40	57,55
Bridges	57,85	100	56,12	15,21	57,00	12,54	57,85	66,75
Glass	68,26	100	66,74	12,50	67,92	11,17	68,26	95,33
Iris	93,30	100	93,30	15,04	93,30	12,52	93,30	93,20
Wine	90,90	100	89,63	13,77	89,63	16,32	90,90	78,62
General average	79,17	100	77,89	13,77	78,14	16,32	79,17	78,62

Table 3. Obtained results using different classifiers

Dataset	Exhaustive k -NN search		AESA		LAESA		i AESA	
	Acc	Comp	Acc	Comp	Acc	Comp	Acc	Comp
Hepatitis	81,66	100	80,57	52,96	80,54	61,86	81,03	52,78
Zoo	96,00	100	96,00	23,50	96,00	15,26	96,00	19,36
Flag	54,67	100	51,45	28,02	51,45	25,73	51,36	27,65
Echocardiogram	82,44	100	81,77	62,04	81,77	68,23	81,05	63,62
Hayes	81,24	100	81,24	24,82	81,24	23,32	80,77	17,62
Soybean-large	85,40	100	85,40	2,51	85,40	4,49	85,40	1,96
Bridges	57,85	100	57,85	25,62	57,85	36,10	57,85	25,07
Glass	68,26	100	66,45	14,02	67,92	20,83	66,34	12,62
Iris	93,30	100	93,30	9,22	93,30	6,86	93,30	7,54
Wine	90,90	100	89,01	15,46	90,90	25,26	90,01	10,62
General average	79,17	100	78,30	25,82	78,64	28,79	78,31	23,88

Table 4. Obtained results using different classifiers

Dataset	Probabilistic i AESA		TLAESA		Modified TLAESA		PROPOSED CLASSIFIER AEMD ($SimThres=SimA$)	
	Acc	Comp	Acc	Comp	Acc	Comp	Acc	Comp
Hepatitis	80,64	32,44	81,33	87,54	81,66	72,65	81,63	13,52
Zoo	94,00	17,51	96,00	42,74	96,00	23,95	94,00	24,76
Flag	49,62	26,41	52,84	48,41	52,09	32,95	52,05	17,41
Echocardiogram	80,06	63,08	81,77	71,58	82,44	44,62	81,70	18,50
Hayes	80,07	16,74	80,54	46,42	81,06	24,05	81,05	12,97
Soybean-large	82,15	2,04	85,40	47,51	85,40	16,85	83,07	23,46
Bridges	56,95	25,064	56,74	46,75	57,23	38,74	57,00	12,54
Glass	66,21	12,06	67,92	62,47	67,72	22,85	67,92	11,17
Iris	93,30	8,01	93,30	41,51	93,30	11,65	93,30	12,52
Wine	90,90	10,54	90,90	39,75	90,90	12,64	89,63	16,32
General average	77,39	21,39	78,67	53,47	78,78	30,10	78,14	16,32

In order to compare the classifiers proposed in this work, different classifiers, based on Approximating and Eliminating, were considered. In table 3 and 4, the obtained results are shown.

Form table 3 and 4, we can observe that when the comparison function does not satisfy the triangle inequality, AESA, LAESA, iAESA, TLAESA and modified TLAESA algorithms are inexact methods (the obtained results are not the same as using the exhaustive search). However, the percentage of comparisons is, on average,

reduced from 100%, done by the exhaustive search, to 25.82 %, 28.79 %, 23.88%, 53.47 % and 30.10%, respectively.

5 Conclusions

In this work, a fast approximated k -MSN classifier for mixed data, based on Approximating and Eliminating approach, applicable when the comparison function does not satisfy metric properties was proposed. In order to compare our method, AESA, LAESA, iAESA, probabilistic iAESA, TLAESA, modified TLAESA algorithms were implemented using the same prototype comparison function for mixed data. Based on our experimental results, it is possible to conclude that, our classifier (AEMD) obtained competitive accuracy, but with a smaller number of comparisons between prototypes.

As future work, we are going to look for an strategy to reduce the memory space required to store the similarity matrix (SM).

References

1. Cover, T.M., Hart, P.E.: Nearest neighbor pattern classification. *Trans. Information Theory* 13, 21–27 (1967)
2. Vidal, E.: An algorithm for finding nearest neighbours in (approximately) constant average time complexity. *Pattern Recognition Letters* 4, 145–157 (1986)
3. Micó, L., Oncina, J., Vidal, E.: A new version of the nearest-neighbour approximating and eliminating search algorithm (AESA) with linear preprocessing-time and memory requirements. *Pattern Recognition Letters* 15, 9–17 (1994)
4. Mico, L., Oncina, J., Carrasco, R.: A fast Branch and Bound nearest neighbor classifier in metric spaces. *Pattern Recognition Letters* 17, 731–739 (1996)
5. Figueroa, K., Chávez, E., Navarro, G., Paredes, R.: On the last cost for proximity searching in metric spaces. In: Álvarez, C., Serna, M.J. (eds.) *WEA 2006*. LNCS, vol. 4007, pp. 279–290. Springer, Heidelberg (2006)
6. Fukunaga, K., Narendra, P.: A branch and bound algorithm for computing k -nearest neighbors. *IEEE Trans. Comput.* 24, 743–750 (1975)
7. Gómez-Ballester, E., Mico, L., Oncina, J.: Some approaches to improve tree-based nearest neighbor search algorithms. *Pattern Recognition Letters* 39, 171–179 (2006)
8. Yong-Sheng, C., Yi-Ping, H., Chiou-Shann, F.: Fast and versatile algorithm for nearest neighbor search based on lower bound tree. *Pattern Recognition Letters* 40(2), 360–375 (2007)
9. Bustos, B., Navarro, G.: Probabilistic proximity search algorithms based on compact partitions. *Journal of Discrete Algorithms (JDA)* 2(1), 115–134 (2003)
10. Tokoro, K., Yamaguchi, K., Masuda, S.: Improvements of TLAESA nearest neighbor search and extension to approximation search. In: *ACSC 2006: Proceedings of the 29th Australian Computer Science Conference*, pp. 77–83 (2006)
11. García-Serrano, J.R., Martínez-Trinidad, J.F.: Extension to C-Means Algorithm for the use of Similarity Functions. In: Żytkow, J.M., Rauch, J. (eds.) *PKDD 1999*. LNCS (LNAI), vol. 1704, pp. 354–359. Springer, Heidelberg (1999)
12. Blake, C., Merz, C.: *UCI Repository of machine learning databases*, Department of Information and Computer Science, University of California, Irvine, CA (1998), <http://www.uci.edu/mllearn/databases/>

Entity Network Prediction Using Multitype Topic Models

Hitohiro Shiozaki¹, Koji Eguchi², and Takenao Ohkawa²

¹ Graduate School of Science and Technology, Kobe University,
1-1 Rokkodai, Nada, Kobe, 657-8501, Japan

hitohiro@cs25.scitec.kobe-u.ac.jp

² Graduate School of Engineering, Kobe University,

1-1 Rokkodai, Nada, Kobe, 657-8501, Japan

eguchi@port.kobe-u.ac.jp

Abstract. Conveying information about *who*, *what*, *when* and *where* is a primary purpose of some genres of documents, typically news articles. To handle such information, statistical models that capture dependencies between named entities and topics can serve an important role. Although some relationships between *who* and *where* should be mentioned in such a document, no statistical topic models explicitly addressed the textual interactions between a *who*-entity and a *where*-entity. This paper presents a statistical model that directly captures dependencies between an arbitrary number of word types, such as *who*-entities, *where*-entities and topics, mentioned in each document. We show how this multitype topic model performs better at making predictions on entity networks, in which each vertex represents an entity and each edge weight represents how a pair of entities at the incident vertices is closely related, through our experiments on predictions of *who*-entities and links between them.

1 Introduction

The primary purpose of the documents that report factual events, such as news articles, is to convey information on *who*, *what*, *when* and *where*. For this kind of documents, statistical entity-topic models [8] were proposed to capture dependencies between *who/where* (i.e. named entities such as persons, organizations, or locations) and *what* (i.e. topics) mentioned in each article. In spite of the fact that each entity type has different characteristics and so it has different distribution, these models represented all types of entities as a single class. This paper attempts to directly capture dependencies between multiple types of entities, such as *who*-entities (i.e. persons, organizations, or nationalities) and *where*-entities (i.e. locations, geographical/social/political entities, or facilities), and general words.

In this paper, we review a series of graphical models that extended a statistical topic model called Latent Dirichlet Allocation (LDA) [4] to explicitly model entities mentioned in text. As in [8] we take advantage of recent developments in named entity recognition to identify entities mentioned in articles. We then develop a multitype topic model that can explicitly capture dependencies between an arbitrary number of word

types, such as who-entity type, where-entity type and general word type. We demonstrate that our model can predict who-entities more effectively, comparing with two other different topic models. We also exhibit that links between entities can be effectively predicted using our model.

2 Related Work

Statistical topic models (e.g., [7,4,11,6,10]) are based on the idea that documents are mixtures of topics, where a topic is a probability distribution over words. Blei et al. [4] proposed one of the topic models called Latent Dirichlet Allocation (LDA), introducing a Dirichlet prior on multinomial distribution over topics for each document. To estimate the LDA model, they used Variational Bayesian method. Instead of using the Variational Bayesian method, Griffiths et al. [6] applied the Gibbs sampling method to estimate the LDA model.

More recently, Newman et al. [8] proposed several statistical entity-topic models, extending the LDA model. Those models attempted to capture dependencies between entities and topics, where the entities are *mentioned* in text; however, the models did not distinguish specific types of entities, such as who-entities and where-entities. Therefore, those models are hardly sufficient to represent an *event* that consists of multiple types of entities. On the other hand, our goal is to model the events that are mentioned in text. As a step towards this goal, this paper develops a multitype topic model by extending the models mentioned above to represent dependencies between an arbitrary number of word types, such as who-entity type, where-entity type and general word type. To estimate our model, we use the Gibbs sampling method, following [6].

3 Models

In this section we describe three graphical models. We start with LDA, followed by SwitchLDA and GESwitchLDA. The LDA is a popular model that can automatically infer a set of topics from a collection of documents [4]. The SwitchLDA was modeled by extending the LDA to capture dependencies between entities and topics, and its prediction performance was shown to be stable over different corpora [8]. The third model, GESwitchLDA is our model that aims to better fit multi-class textual data, such as of who-entities, where-entities and general words, by generalizing the SwitchLDA model. We use the LDA [4] as a baseline model for comparing with our GESwitchLDA in the experiments in Section 4. We also use the SwitchLDA as another baseline model.

Here we introduce the notation used in graphical models, generative processes and Gibbs sampling equations in the rest of this paper: D is the number of documents, T is the number of topics, N_d is the total number of words in document d , α and β are Dirichlet prior hyper-parameters, γ is a Beta or Dirichlet prior hyper-parameter, θ is the topic-document distribution, ϕ is the word-topic distribution, z_i is a topic, and w_i is a word or entity. In the case of the SwitchLDA, a tilde mark is used to denote the entity version of a variable. In the case of the GESwitchLDA, a tilde mark and a hat mark are used to denote the who-entity version and where-entity version, respectively.

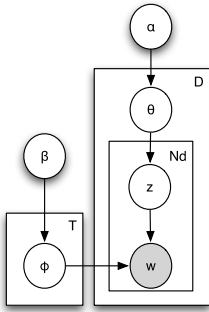


Fig. 1. LDA

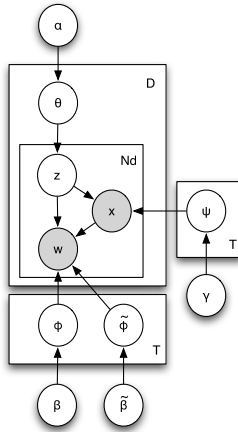


Fig. 2. SwitchLDA

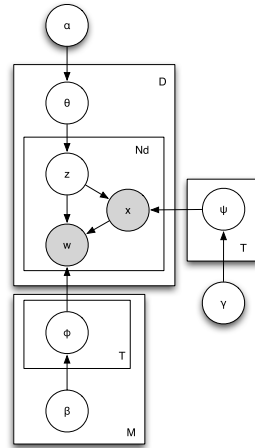


Fig. 3. GESwitchLDA

3.1 LDA

To explain the differences between the three graphical models, let us start with the LDA model shown in Fig. 1. The LDA's generative process is:

1. For all d documents sample $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For all t topics sample $\phi_t \sim \text{Dirichlet}(\beta)$
3. For each of the N_d words w_i in document d :
 - (a) Sample a topic $z_i \sim \text{Multinomial}(\theta_d)$
 - (b) Sample a word $w_i \sim \text{Multinomial}(\phi_{z_i})$

Some estimation algorithms were applied to the LDA [4,6]. Following [6], we use the Gibbs sampling to estimate the LDA model. Note that the LDA does not distinguish specific types of words, and so this distinction was made at post-processing stage (i.e. outside of the model) when we made predictions about who-entities in Section 4.

3.2 SwitchLDA

SwitchLDA model shown in Fig. 2 was introduced in [8], extending the LDA model. In this model, an additional Binomial distribution ψ (with a Beta prior of γ) was incorporated to control the fraction of entities in topics. The generative process of the SwitchLDA is:

1. For all d documents sample $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For all t topics sample $\phi_t \sim \text{Dirichlet}(\beta)$, $\tilde{\phi}_t \sim \text{Dirichlet}(\tilde{\beta})$ and $\psi_t \sim \text{Beta}(\gamma)$
3. For each of the N_d words w_i in document d :
 - (a) Sample a topic $z_i \sim \text{Multinomial}(\theta_d)$
 - (b) Sample a flag $x_i \sim \text{Binomial}(\psi_{z_i})$
 - (c) If $(x_i = 0)$ sample a word $w_i \sim \text{Multinomial}(\phi_{z_i})$
 - (d) If $(x_i = 1)$ sample an entity $w_i \sim \text{Multinomial}(\tilde{\phi}_{z_i})$

The estimation algorithm for the SwitchLDA followed the Gibbs sampling approach, as described in [8]. Note that the SwitchLDA does not distinguish more specific types of entities, and so this distinction was made at post-processing stage (i.e. outside of the model) when we made predictions about who-entities in Section 4.

3.3 GESwitchLDA

In our GESwitchLDA model shown in Fig. 3 we generalize the SwitchLDA to handle an arbitrary number (M) of word types. Therefore, we redefine ψ as Multinomial distribution with the Dirichlet prior γ . The generative process of the GESwitchLDA is:

1. For all d documents sample $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For all t topics:
 - (a) Sample $\psi_t \sim \text{Dirichlet}(\gamma)$
 - (b) For each word type $y \in \{0, \dots, M-1\}$, sample $\phi_t^y \sim \text{Dirichlet}(\beta^y)$
3. For each of the N_d words w_i in document d :
 - (a) Sample a topic $z_i \sim \text{Multinomial}(\theta_d)$
 - (b) Sample a flag $x_i \sim \text{Multinomial}(\psi_{z_i})$
 - (c) For each word type $y \in \{0, \dots, M-1\}$:
 - If $(x_i = y)$ sample a type- y word $w_i \sim \text{Multinomial}(\phi_{z_i}^y)$

We use the Gibbs sampling approach to estimate the GESwitchLDA model using the equations in Appendix.

In the experiments in Section 4 we divided entities into two classes, who-entity and where-entity, and thus the number of word types $M = 3$ in this case. The GESwitchLDA’s generative process when $M = 3$ is:

1. For all d documents sample $\theta_d \sim \text{Dirichlet}(\alpha)$
2. For all t topics sample $\phi_t \sim \text{Dirichlet}(\beta)$, $\tilde{\phi}_t \sim \text{Dirichlet}(\tilde{\beta})$, $\hat{\phi}_t \sim \text{Dirichlet}(\hat{\beta})$ and $\psi_t \sim \text{Dirichlet}(\gamma)$
3. For each of the N_d words w_i in document d :
 - (a) Sample a topic $z_i \sim \text{Multinomial}(\theta_d)$
 - (b) Sample a flag $x_i \sim \text{Multinomial}(\psi_{z_i})$
 - (c) If $(x_i = 0)$ sample a word $w_i \sim \text{Multinomial}(\phi_{z_i})$
 - (d) If $(x_i = 1)$ sample a who-entity $w_i \sim \text{Multinomial}(\tilde{\phi}_{z_i})$
 - (e) If $(x_i = 2)$ sample a where-entity $w_i \sim \text{Multinomial}(\hat{\phi}_{z_i})$

4 Experiments

4.1 Data Sets

We used the TDT2 and TDT3 collections [1] that were tagged by the BBN Identifier [3] for our experiments. They originally contained a mix of broadcast news and newswire stories. We used for the experiments only the English stories in these collections, but not the stories in other languages or the metadata such as pre-defined topics or categories. We used the TDT2 for training and the TDT3 for testing. Statistics for the data sets are summarized in Table 1. We removed the 418 stopwords included in the stop list used in *InQuery* system [5], and also removed words and entities that occurred in less than 10 documents.

Table 1. Statistics for data sets

	TDT2	TDT3
Documents	45,260	26,770
Unique Words	27,685	21,954
Unique Who-entities	7,300	4,591
Unique Where-entities	1,637	1,121
Total Words	7,634,722	4,583,162
Total Who-entities	600,638	378,725
Total Where-entities	343,432	199,760

Table 2. Who-entity prediction results example. The top row shows an excerpt from an article, with redacted who-entities indicated by XXXXX. Middle row shows the list of relevant who-entities. The bottom row shows the predicted who-entity list ordered by likelihood.

The XXXXX accord and XXXXX Camry are the most popular for buyers and auto thieves. More on that from XXXXX . The latest XXXXX figures show that auto thefts were down overall in , 1997 . By 4% , in fact. But that is little solace for the owners of the cars that topped the national insurance crime bureau’s list of most stolen automobiles in the United States. The XXXXX accord and XXXXX Camry occupy the number one and two spots on the list.

actual who-entity list: Honda, Toyota, Charles Feldman, FBI, CNN

predicted who-entity list: Italian, U.N., General Motors, Pakistani, GM, Chrysler , Americans , Indian , American , Ford , Supreme Court , Smith , U.S. , VOA , Congress , Annan , United Nations , Japanese , *FBI, *CNN, Volkswagen , *Honda, European , BMW , Security Council , *Toyota

4.2 Who-Entity Prediction

Estimation. For who-entity prediction task, the three models: the LDA, the SwitchLDA and the GESwitchLDA are first trained on words, who-entities, and where-entities. The models then make predictions about who-entities using just words or both words and where-entities. We need to set hyper-parameters for the LDA [46], as well as for the SwitchLDA, and the GESwitchLDA. For all of the experiments, we set the number of topics $T = 100, 200, \text{ and } 300$ for each of the three models. We fixed Dirichlet priors $\alpha = 50/T$ and $\beta = 0.01$, which were reported to be appropriate for various collections [10]. The other hyper-parameters were empirically determined using the training data, as described in the rest of this section.

Prediction. We evaluated all three models on a specific who-entity prediction task. For this task, the models were first trained on words, who-entities, and where-entities using the TDT2. The model then makes predictions about who-entities over the TDT3 in the following two ways:

1. using words and where-entities(w+e).
2. using only words(w).

oil	0.1746	internet	0.0820	game	0.0408
prices	0.0669	web	0.0568	team	0.0379
production	0.0381	information	0.0457	coach	0.0308
price	0.0325	site	0.0454	basketball	0.0273
gas	0.0315	mail	0.0266	tournament	0.0229
crude	0.0215	sites	0.0261	national	0.0212
barrels	0.0182	online	0.0187	play	0.0179
cut	0.0176	computer	0.0166	college	0.0179
world	0.0146	service	0.0132	season	0.0166
silver	0.0142	users	0.0122	points	0.0157
gasoline	0.0133	world	0.0105	final	0.0133
barrel	0.0132	data	0.0096	win	0.0131
pipeline	0.0128	electronic	0.0092	championship	0.0122
natural	0.0123	wide	0.0092	point	0.0122
cents	0.0120	line	0.0091	four	0.0120
OPEC	0.4140	America_Online	0.1271	Duke	0.0528
Texaco	0.1185	Reuters	0.1197	John	0.0518
Berkshire	0.0893	Bloomberg	0.0540	Stanford	0.0391
Shell	0.0536	Yahoo	0.0488	Kentucky	0.0332
Exxon	0.0503	AOL	0.0443	NCAA	0.0321
crisco	0.0503	NYT	0.0392	Rutgers	0.0307
Pertamina	0.0455	Excite	0.0325	Huskies	0.0303
Buffett	0.0422	Amazon.com	0.0310	Big_East	0.0217
Caspian	0.0422	Online	0.0281	UConn	0.0184
Chevron	0.0406	Holmes	0.0229	Wildcats	0.0184
Turkmenistan	0.0972	Cambridge	0.0942	North_Carolina	0.1261
Saudi_Arabia	0.0938	Va.	0.0779	St.	0.1258
Caspian	0.0914	Honolulu	0.0747	Connecticut	0.0810
Azerbaijan	0.0868	Fla.	0.0714	Kentucky	0.0661
Olean	0.0845	Bridge	0.0649	Utah	0.0619
Venezuela	0.0752	Amazon	0.0617	Michigan	0.0474
Mexico	0.0590	San_Francisco	0.0552	Princeton	0.0455
Caspian_Sea	0.0579	Calif.	0.0487	Rhode_Island	0.0436
Ecuador	0.0556	Dayton	0.0455	Arizona	0.0409
Baku	0.0498	Mass.	0.0422	Tennessee	0.0229

Fig. 4. Examples of topics captured by GESwitchLDA. In each topic, we list most likely words and their probability at the top, who-entities at the middle, and where-entities at the bottom.

The likelihood of an entity in each test document is calculated by $p(e|d) = \sum_t p(e|t)p(t|d)$, where $p(e|t)$ is estimated during training, and the topic mixture in the test document $p(t|d)$ is estimated by resampling both all words and all where-entities (or by resampling only all words) using learned word distribution $p(w|t)$ and where-entity distribution $p(o|t)$. We estimated $p(t|d)$ using Gibbs sampling.

We illustrate the process of the who-entity prediction in Table 2 using an example from the TDT data. The first row shows an excerpt from an article of TDT3, with who-entities indicated by XXXXX. Middle row shows the list of actual who-entities. The bottom row shows the predicted who-entity list ordered by likelihood computed using both words and where-entities (or using only words). Some examples of the topics captured by GESwitchLDA are shown in Fig. 4.

Evaluation Metrics. Using the model parameters estimated in training, the models computed the likelihood of every possible entity, and then listed the who-entities in order of the likelihood. We computed MAP (mean average precision) [2], and GMAP (geometric mean average precision) [9], as well as average best rank and average median rank. The average best rank is defined as the average of the best rank of relevant who-entities, and the average median rank is the average rank of who-entities at median of relevant who-entity ranked list.

Table 3. Best results of who-entity prediction (without name identification)

model	MAP	GMAP	avg best rank	avg median rank
LDA (w+e, T=300)	0.1998	0.0818	118.10	482.93
SwitchLDA (w+e, T=300)	0.2036	0.0816	119.78	484.38
GESwitchLDA (w+e, T=300)	0.2048	0.0833	119.08	480.64
LDA (w, T=200)	0.1565	0.0558	135.13	549.86
SwitchLDA (w, T=300)	0.1603	0.0568	136.98	565.48
GESwitchLDA (w, T=300)	0.1595	0.0569	135.55	560.18

Results. The best results for LDA, SwitchLDA and GESwitchLDA are shown in Table 3. To obtain the best results, we determined through experiments that $T = 300$ was the best parameter for all three models, except the case of the LDA using only words. We determined that $T = 200$ was the best parameter for the LDA using only words. We determined the best parameters $\tilde{\beta} = \hat{\beta} = 0.01$ for both the SwitchLDA and the GESwitchLDA, $\gamma = 5.0$ for the SwitchLDA, and $\gamma = 4.0$ for the GESwitchLDA.

Given the best parameters in our experiments, our GESwitchLDA model gave the best results, in terms of both MAP and GMAP, over the other two models in the case of using both words and where-entities for prediction. In terms of MAP, the GESwitchLDA gave 2.5% improvement in this case¹, comparing with the best results of the LDA model under the same condition. We further performed the Wilcoxon signed-rank test (two-tailed) to the pair of GESwitchLDA - LDA and the pair of GESwitchLDA - SwitchLDA. In terms of MAP, the resulting p -values of these pairs were less than 0.01 in the case of using both words and where-entities. It means the the performance improvement of the GESwitchLDA over both the SwitchLDA and the LDA was statistically significant, in this case. As for the case of using only words, the improvement of the GESwitchLDA over the LDA was also statistically significant at 0.01 level; however, that over the SwitchLDA was not. In terms of average best rank and average median rank, we observed that few very bad results made performance values unfairly poor. In contrast, MAP was observed to be more stable in this sense.

We also calculated likelihood of who-entities in the manner of not using resampling. In detail, we calculated the topic mixture in a test document as $p(t|d) = \sum_w p(t|w)p(w|d)$. In this manner we can predict who-entities incrementally for a given document. The results using the GESwitchLDA are shown in Table 4. The results show that the model can predict who-entities even for incoming streams of documents, keeping fairly good prediction performance. Furthermore, we also applied some heuristics for name identification at pre-processing stage, such as, when only the first name of a person appears in a document, replacing it with his/her full name found by searching backward in the document. The results of the GESwitchLDA are shown in Table 5, where the performance was improved by applying the name identification processing.

¹ It looks relatively small, although the improvement of our model turned out to be statistically significant, as described later. We believe one reason is that the evaluation values were averaged all over a large number of test documents, as shown in Table 1, and another reason is that all predicted entities that did not appear in a document were deemed to be irrelevant even if some of those were closely related to the document content.

Table 4. Best results of who-entity prediction without resampling (without name identification)

model	MAP	GMAP	avg best rank	avg median rank
GESwitchLDA (w+e, T=300)	0.1970	0.0784	110.17	461.82
GESwitchLDA (w, T=300)	0.1554	0.0613	120.72	516.01

Table 5. Best results of who-entity prediction with name identification of GESwitchLDA

model	MAP	GMAP	avg best rank	avg median rank
GESwitchLDA (w+e, T=300)	0.2141	0.0893	114.21	439.21
GESwitchLDA (w, T=300)	0.1611	0.0605	128.28	505.01

Table 6. Results of who-entity link prediction with name identification

model	MAP	accuracy
LDA (T=100)	0.6062	0.5393
SwitchLDA (T=100)	0.6235	0.5551
GESwitchLDA (T=100)	0.6257	0.5564

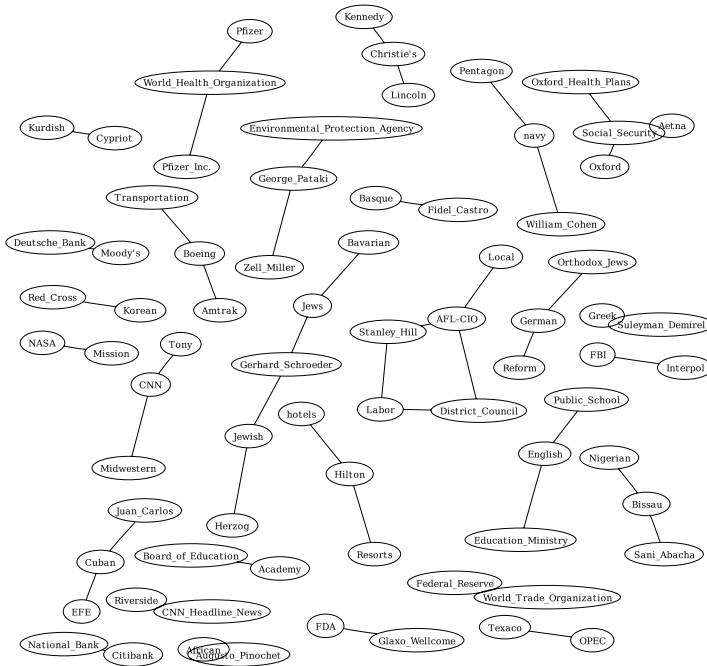


Fig. 5. Examples of predicted who-entity networks

4.3 Entity Link Prediction

We further carried out experiments on who-entity link prediction. We computed affinity of a pair of who-entities e_i and e_j by $p(e_i|e_j)/2 + p(e_j|e_i)/2$, and then listed entity pairs in order of the affinity, where $p(e_i|e_j) = \sum_t p(e_i|t)p(t|e_j)$ is estimated during training in the same manner in the previous section. Following [8], we generated two sets of entity pairs: (1) the true pairs that contain pairs that were never seen in any training document but were seen in test documents; and (2) false pairs that contain pairs that were never seen in any training or test document. The number of true pairs N_t and false pairs N_f were 104,721 and 98,977, respectively. The results can be seen in Table 6. We used a couple of evaluation metrics: mean average precision (MAP) and accuracy at top-ranked N_t predicted result. Our GESwitchLDA modestly outperformed the other two models: the LDA and the SwitchLDA, in terms of both MAP and accuracy. The maximum improvement was 3.2% in the case of MAP. Some examples of predicted who-entity networks are shown in Fig. 5, where each vertex represents an entity and each edge length represents strength of affinity between a pair of entities at the incident vertices. Although the networks of who-entities were discussed above, more specific social networks (e.g. person-entity networks) or where-entity networks can also be predicted in the same manner.

5 Conclusions

We developed a graphical model GESwitchLDA, generalizing for an arbitrary number of word types such as words, *who*-entities (i.e. persons, organizations, or nationalities) and *where*-entities (i.e. locations, geographical/social/political entities, or facilities), in order to enable to capture dependencies between them. We compared this model with two other models on *who*-entity prediction task and entity link prediction task, using real data of news articles. We showed that the GESwitchLDA achieved significant improvement over the previous models in terms of some measures that are well-accepted in information retrieval research area, by distinguishing multiple types of entities: in this case, *who* and *where*. Using our model, entity networks, such as social networks, can be effectively constructed from textual information.

This model can also be applied to other multiple types of words. For example, we can use this model to capture multiple types of entities in bio-medical articles, such as protein names, gene names and chemical compound names, even if more than two entity types are involved. In another direction of future work, we plan to extend the model to incorporate a temporal aspect of events.

Acknowledgements

We thank Giridhar Kumaran, University of Massachusetts Amherst, for providing annotation data. We also thank Atsuhiko Takasu, National Institute of Informatics, for valuable discussions. This work was supported in part by the Grant-in-Aid for Scientific Research on Priority Areas “Info-plosion” (#19024055), Young Scientists A (#17680011) and Exploratory Research (#18650057) from the Ministry of Education, Culture, Sports, Science and Technology of Japan.

References

1. Allan, J.: Introduction to Topic Detection and Tracking. In: Topic Detection and Tracking: Event-based Information Organization, ch. 1, Kluwer Academic Publishers, Dordrecht (2002)
2. Baeza-Yates, R., Ribeiro-Neto, B.: Retrieval Evaluation. In: Modern Information Retrieval, ch. 3, pp. 73–97. Addison-Wesley, Reading (1999)
3. Bikel, D.M., Schwartz, R.L., Weischedel, R.M.: An algorithm that learns what’s in a name. *Machine Learning* 34, 211–231 (1999)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet allocation. *Journal of Machine Learning Research* 3, 993–1022 (2003)
5. Callan, J.P., Croft, W.B., Harding, S.M.: The INQUERY retrieval system. In: Proceedings of the 3rd International Conference on Database and Expert Systems Applications, Valencia, Spain, pp. 78–83 (1992)
6. Griffiths, T.L., Steyvers, M.: Finding scientific topics. *Proceedings of the National Academy of Sciences of the United States of America* 101, 5228–5235 (2004)
7. Hofmann, T.: Probabilistic latent semantic indexing. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Berkeley, CA, USA, pp. 50–57 (1999)
8. Newman, D., Chemudugunta, C., Smyth, P., Steyvers, M.: Statistical entity-topic models. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, NY, USA, pp. 680–686 (2006)
9. Robertson, S.: On GMAP: and other transformations. In: Proceedings of the 15th ACM International Conference on Information and Knowledge Management, New York, NY, USA, pp. 78–83 (2006)
10. Steyvers, M., Griffiths, T.: Probabilistic Topic Models. In: Handbook of Latent Semantic Analysis, ch. 21, Lawrence Erlbaum Associates (2007)
11. Ueda, N., Saito, K.: Parametric mixture models for multi-labeled text. In: Advances in Neural Information Processing Systems, 15, Cambridge, MA, USA (2003)

Appendix

Gibbs Sampling Equations for GESwitchLDA

In the following equations. α and β are Dirichlet priors, and γ is another Dirichlet prior. β^y corresponds to Dirichlet prior for type- y words. The notation C_{pq}^{PQ} represents counts from respective count matrices, e.g. count of words in a topics, or counts of topic in a document.

$$p(z_i = t | w_i = v, x = y, z_{-i}, x_{-i}, w_{-i}, \alpha, \beta, \gamma) \propto \frac{C_{td,-i}^{TD} + \alpha}{\sum_t C_{td,-i}^{TD} + T\alpha} \frac{n_{t,-i}^y + \gamma}{n_{t,-i}^{all} + M\gamma} \frac{C_{w_y t, -i}^{W_y T} + \beta^y}{\sum_w C_{w_y t, -i}^{W_y T} + W\beta^y}$$

where $n_t^y = \sum_{w_y} C_{w_y t}^{W_y T}$, $n_t^{all} = \sum_y n_t^y$.

Using Supervised and Unsupervised Techniques to Determine Groups of Patients with Different Doctor-Patient Stability

Eu-Gené Siew¹, Leonid Churilov², Kate A. Smith-Miles³, and Joachim P. Sturmbérg⁴

¹ School of Information Technology, Monash University, Sunway Campus, Jalan Lagoón Selatan, 46150 Bandar Sunway, Selangor D.E.
siew.eu-gene@infotech.monash.edu.my

² National Stroke Research Institute, Heidelberg Heights, Victoria 3081, Australia
lchurilov@nsri.org.au

³ Faculty of Science and Technology, Deakin University, Burwood, Victoria 3125, Australia
Kate.Smith-Miles@deakin.edu.au

⁴ Department of General Practice, Monash University, Victoria 3800, Australia
jp.sturmbérg@bigpond.com

Abstract. Decision trees and self organising feature maps (SOFM) are frequently used to identify groups. This research aims to compare the similarities between any groupings found between supervised (Classification and Regression Trees - CART) and unsupervised classification (SOFM), and to identify insights into factors associated with doctor-patient stability. Although CART and SOFM uses different learning paradigms to produce groupings, both methods came up with many similar groupings. Both techniques showed that self perceived health and age are important indicators of stability. In addition, this study has indicated profiles of patients that are at risk which might be interesting to general practitioners.

Keywords: Doctor-patient stability (MCI), Classification and Regression Trees (CART), Self Organising Feature Maps (SOFM or SOM), supervised learning, unsupervised learning.

1 Introduction

Two of the most popular methods for classifying and clustering data are decision trees and self-organising maps [1]. Classification and regression tree (CART) is used to classify the data while Kohonen's self-organising map (SOFM) is to cluster. These techniques differ in their approach to grouping patients in that CART [2] uses a supervised learning approach that requires a target variable to guide its groupings, whereas SOFM [3] uses an unsupervised learning approach by grouping patients without the need to specify the desired output.

This paper compares the similarities between any groupings found between supervised and unsupervised techniques, and to identify insights into factors associated with doctor-patient stability.

Long-term doctor-patient stability is an important aspect to achieving continuity of care [4]. Continuity of care has many benefits. It has been shown to build patients' personal trust in doctors [5], to increase the knowledge of doctor and patient about each other which in turn promotes an increased understanding of the social context of the patient [6] and has been shown to be vital to patient satisfaction [7].

Overall, research in this area of doctor-patient stability mostly treats patients as a single homogenous group [8, 9]. The factors are typically treated as having a one-to-one linear relationship to the outcome doctor-patient relationship stability variable.

An additional objective of this paper is to investigate the groups of patients produced by CART and SOFM and to evaluate these groups in terms of predicting doctor-patient stability.

This paper is organized as follows: Section 2 describes the study design; research methodology is discussed in Section 3; Sections 4 and 5 present SOFM and CART results respectively; Section 6 demonstrates how the results are validated; Section 7 describes the key profiles and comparisons between SOFM and CART, while the conclusions are made in Section 8.

2 Study Design

The data is obtained from a survey of randomly selected general practices in the NSW Central Coast, Australia. This region is estimated to have up to 230,000 people and ranks as top ninth highest population in Australia [10]. The practices in the area ($n=93$), were stratified into five classes according to their size, which is categorized into solo, 2, 3 to 4, and 5 and over, doctors. 100 consecutive patients are selected from the five practices of each of the five classes. In total, twenty of the sixty-one doctors (which constitute about 33 per cent) agreed to participate. Due to the high demand placed on doctors and their patients, eight doctors who initially agreed withdrew from the study. Information about 1,122 patients and their respective doctors was collected. Data collection occurred between February and November 1999.

Table 1. Data dictionary and average characteristics of the items contained in the questionnaire

Section	Abbreviation	Mean	Everywhere except for the Age variable, the score of "1" means that...
Pre-consultation items	Time	0.75	Doctor always has enough time for me
	Age	49.98	Age (years)
	Knowdo	0.68	Knows doctor well
	Health	0.74	Patient perceived to be in excellent health
	Psysym	0.52	Psychological distress
	Soc	0.48	Social distress
	Morbidity	0.15	Poly-morbidity
Consultation items	Condif	0.28	Most difficult consultations
	MCI	0.68	Most stable doctor-patient relationship
	Consl	0.87	Longer Consultations
Post consultation items	Commun	0.6	Excellent communication with doctor
	Enable	0.31	Highest enablement
	Satisf	0.8	Highest satisfaction

The questionnaire is divided into three parts: the first were answered by the doctor, the second by the patients before consultation and the final part by the patients after consultation. The questionnaires obtained information about the health service environment, the doctor's characteristics and perceptions about the patient, patient characteristics, information about the consultation process and the outcome. Only relevant variables are shown in Table 1 which indicates the average mean values of each questionnaire variables and describes the abbreviation used.

Doctor-patient stability variable is measured as modified continuity index (MCI). MCI is developed by Godkin and Rice and it indicates the frequency and intensity of the relationship by dividing the number of different doctors visited by the number of visits in a time period [11]. It is a continuous number between 0 and 1 and is the frequency visit to a dominant doctor over the number of visits in a year. Values close to 0 would indicate poor doctor-patient stability and 1, high doctor-patient stability.

3 Research Methodology

The research design contains the following stages:

Stage 1: Application of CART and SOFM using the training data sets.

At the first stage, about 20 per cent of patients were randomly allocated into evaluation set and the rest into training set. Both data mining techniques (CART and SOFM) were applied separately to group the general practice patients based on demographics and clinical variables using the training data set. For SOFM, a software package called Viscovery was used to model the data [12].

Stage 2: Validation of the CART and SOFM models using evaluation set (holdout sample).

The models generated in the training set were then applied to the relevant evaluation set. If the models were generalisable then the performance of the evaluation sets were analogous to the training period. To make the comparison, Mean absolute deviation (MAD) [13] and the coefficient of multiple determination (R^2) were used.

Stage 3: An analysis and comparison of the results from supervised and unsupervised data mining techniques.

4 SOFM Clusters

This section describes the application of SOFM onto the training data set of which SOFM generated 10 clusters (Stage 1) which were then renumbered in ascending order of MCI. There were three broad groups of clusters: those with MCI of 0.5 to 0.6, MCI of 0.6 to 0.76 and MCI over 0.76. When the clusters were created, the stability variable was left out as inputs. This was to see how well other variables were able to predict stability.

There seems to be a strong correlation between age and stability. If the patient's age is between 30 to 38, they are likely to group in Cluster 1 to 4 and are likely to have MCI between 0.5 to 0.58.

There also seems to be a separation between two main groups of patients. Those groups (Cluster 5-10) whose average stability is 0.73 and above (high stability) and those (Cluster 1-4) whose average stability is 0.58 and below (low stability). The following describes those clusters:

Table 2. Summary of SOFM clusters based on MCI

Variables	Low stability				High stability					
	1	2	3	4	5	6	7	8	9	10
Age	31	35	39	32	53	63	63	57	69	69
Health	Good	Good	Good	Good				Poor	Good	
Morbidity		Complex	Social		Psychological			Complex		Physical
Condif	Difficult	Easy		Easy			Difficult	Easy	Easy	Easy
MCI	0.53	0.54	0.55	0.58	0.73	0.76	0.76	0.81	0.81	0.83
Consl	Shortest		Long		Longest	Short				Long
Commun		Good	Good	Poor		Poor		Good		Good
Enable	Lowest	Good	Poor	Poor		Poor				Good
Satisf		Good	Good	Poor		Poor	Good	Good	Good	Good
Cluster										
Total	47	114	130	63	44	54	96	84	106	73

After the profiles of the stability clusters were examined, SOFM was used as a prediction tool. The MAD and R² in the training set were 0.1622 and 0.2356 respectively.

5 CART Results

This section describes the results obtained from applying CART onto the training data set (Stage 1). A separate study was conducted on the patient stability variable for which CART generated 7 terminal nodes. Figure 1 shows the CART tree diagram of stability when it was run on the training data set. Terminal node 1 has the lowest stability, as measured by the MCI index, while Terminal node 7 has the highest.

CART uses three patient variables: patient’s age, knowledge of the doctor and perception of their health. In general, patients whose average age is 46.5 and less tend to have lower doctor-patient stability. Patients who consider themselves in poor health tend to have more stable doctor-patient relationships. In addition, younger and healthier patients (represented in Terminal node 1) have a lower MCI score compared with younger but not-healthier patients (represented in Terminal node 4).

Patients with a high level of knowledge of their doctor are correlated with high stability. Terminal node 5 (with good knowledge of their doctor) has a higher stability score than Terminal node 2 (with poor knowledge of their doctor) even though both represent ages between 46.5 to 64.5 and good self-perceived health.

Like SOFM, there also seems to be a separation between two main groups of patients which are those groups (Cluster 5-10) whose stability is above 0.69 and those (Cluster 1-4) whose stability is 0.69 and below. Age, health and knowledge of doctor

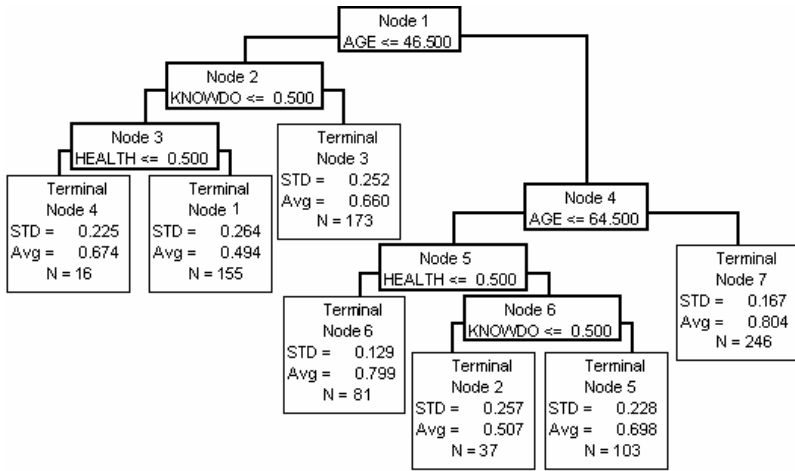


Fig. 1. CART tree of doctor-patient stability in primary care

variables can also be used to separate those 2 groups. Patients age 46.5 years and below, and also those who consider their health to be good but do not have a good knowledge about their doctor has a low stability. Once the profiles of the CART nodes were examined, CART was used as a prediction tool. The MAD and R^2 in the training set were at 0.1227 and 0.3591 respectively.

6 Validation of SOFM and CART Groupings

A comparison was made between the training and evaluation data sets to establish the generalisability of both the SOFM and CART grouping models of patient stability. A significantly higher MAD and lower R^2 in the evaluation set, compared to the training set, would indicate poor applicability of the model. Table 3 shows the comparison between MAD and R^2 of training and evaluation set of both SOFM and CART. Using an unpaired two-tailed t-test and the alpha level of .05, the null hypothesis that the MAD of the training and evaluation sets is statistically similar could not be rejected. The p-value was above 0.05 for both SOFM and CART.

Both these measures provide evidence that the SOFM model of clusters and the CART rules produced could be generalisable in creating patient groups that reflect doctor-patient stability.

Table 3. Comparison of MAD and R^2 for stability variable

Variable	Method	MAD		2 tail P-value	R^2	
		Training set	Evaluation set		Training set	Evaluation set
Stability	SOFM	0.1622	0.1827	0.0684	0.2356	0.2438
	CART	0.1227	0.1346	0.1394	0.3591	0.3977

7 Similarities between Supervised and Unsupervised

Guthrie and Wyke state that for some groups of patients stability is more important [14]. They list an example of a more serious morbidity group of patients that requires higher doctor-patient stability compared with the healthier groups. Thus, the interesting groups would be patients who have serious morbidity but for some reason choose not to have a usual general practitioner. These SOFM groupings are:

Cluster 1: Young patients (average 31.5 years) with complex morbidity and poor communication with their doctor, who have the lowest doctor-patient stability. They are the least enabled and most dissatisfied with their consultations. They represent the highest proportion that judge themselves in excellent health. Doctors find consultations with this group the most difficult.

Cluster 4: Young patients (average 32.0 years) in social distress who are dissatisfied with their consultations. This cluster has one of the lowest rates of enablement and satisfaction. Although they consider themselves in good health, they are oblivious to their social distress and are unable to understand and communicate with their doctor. They also have the second shortest consultation times and felt their doctor does not spend enough time with them.

Cluster 6: Older patients (average 62.6 years) who have negative attitudes towards holistic health care with combinations of morbidity. They have problems communicating with their doctor and have amongst the shortest consultations. They feel not enabled by and were dissatisfied with their consultation.

Those findings are to some extent consistent with CART which ranks age, self perception of health and social morbidity highly as important primary or surrogate splitters.

In addition, a comparison using Cohen Kappa[15] seems to show that SOFM and CART produced similar groupings. As mentioned earlier in Section 4 and 5, both CART and SOFM came up with two groupings of high and low stability groupings. An average of MCI 0.69 can be considered as a threshold for the broad groupings.

An assessment of inter-rater reliability using Cohen Kappa is shown in the tables below. Both SOM and CART are considered as the "raters" of the two categories based on high doctor-patient stability (above MCI 0.69) and low doctor-patient stability (MCI 0.69 and below).

Table 4. Degree of agreement between SOFM and CART

	SOFM	Clusters 1-4	Clusters 5-10	Total
CART		Avg MCI <0.69	Avg MCI >0.70	
Nodes1-4	Avg MCI <0.69	325	56	381
Nodes5-7	Avg MCI >0.70	29	401	430
Total		354	457	811

Table 5. Expected values in each cell if it were due by chance

	SOFM	Clusters 1-4	Clusters 5-10	Total
CART		Avg MCI <0.69	Avg MCI >0.70	
Nodes1-4	Avg MCI <0.69	166.31	214.69	381
Nodes5-7	Avg MCI >0.70	187.69	242.31	430
Total		354	457	811

The total agreement between CART and SOFM is 726 compared with the agreement if it were due to chance of 408.61. The Cohen Kappa in this case is 0.79 which seems to indicate that both SOFM and CART is similar to broadly group doctor-patient stability. If the groupings were due to chance, Cohen Kappa would be 0.

8 Summary and Conclusions

This paper has discussed the use of CART and SOFM to classify patients according to their stability of doctor-patient relationship. The contribution of this research is to identify groups of patients that are at different levels of stability. By doing this, it reveals key variables and profiles that are associated with the stability outcome and highlight high risk groups. There were groupings of patients with combinations of morbidity who, for some reason, consider themselves to be in good health. They do not have a principal general practitioner who can provide continuous care for them.

In addition, this research compares the performance of supervised and unsupervised learning. Both are able to come up with similar groupings based on Cohen Kappa and key attributes which are age, self perception of health and social morbidity.

There are limitations to this research. It is arguable whether the results could be applied outside the New South Wales Central Coast. In general, the central coast region tends to have patients who are predominantly native speakers of English, with less social mobility and with less availability of doctors.

Furthermore, the data on doctors and the general practice are limited. There are only twelve doctors that took part in the survey. A larger sample size of doctors and general practice would enable more association of their variables to patients. It is also probable that particular groups of doctors or even patients may be omitted from the final results because they did not participate in the survey when sampled.

Future research might include open ended questions targeting dissatisfied patients and in particular those unable to communicate with their doctors. These questions may elicit the reasons underlying the poor communication, such as poor doctor training, patient not being able to voice their opinion and doctors who felt rushed to complete as many consultations as possible.

References

1. Goebel, M., Gruenwald, L.: A Survey of Data Mining software Tools. *SIGKDD Explorations* 1(1), 20–33 (1999)
2. Steinberg, D., Phillip, C.: *CART—Classification and Regression Trees*, Salford Systems, San Diego, CA (1997)
3. Deboeck, G., Kohonen, T.: *Visual Explorations in Finance with Self-Organizing Maps*. Springer, London (1998)
4. Freeman, G., Hjortdahl, P.: What future for continuity of care in general practice? *British Medical Journal* 314(7342), 1870–1873 (1997)
5. McWhinney, I.R.: Core values in a changing world. *British Medical Journal* 316, 1807–1809 (1998)
6. Gulbrandsen, P., Hjortdahl, P., Fugelli, P.: General practitioners' knowledge of their patients' psychosocial problems: multipractice questionnaire survey. *British Medical Journal* 314, 1014 (1997)
7. Mansour, A., al-Osimy, M.H.: A study of satisfaction among primary health care patients in Saudi Arabia. *Journal of Community Health* 18, 163–173 (1995)
8. Hjortdahl, P., Laerum, E.: Continuity of care in general practice: effect on patient satisfaction. *British Medical Journal* 304(6837), 1287–1290 (1992)
9. Hjortdahl, P., Borchgrevink, C.F.: Continuity of care: influence of general practitioners' knowledge about their patients on use of resources in consultations. *British Medical Journal* 303(6811), 1181–1184 (1991)
10. New South Wales, <http://www.geocities.com/lockstar/newsouthwales.html>
11. Godkin, M.A., Rice, C.A.: A measure of continuity of care for physicians in practice. *Family Medicine* 16(4), 136–140 (1981)
12. VISCOVERY software, <http://www.eudaptics.com>
13. Weiss, S.M., Zhang, T.: Performance Analysis and Evaluation. In: Ye, N. (ed.) *The handbook of Data Mining*, pp. 425–440. Lawrence Erlbaum Assoc, Mahwah (2003)
14. Guthrie, B., Wyke, S.: Does continuity in general practice really matter? *British Medical Journal* 321, 734–736 (2000)
15. Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20, 37–46 (1960)

Local Projection in Jumping Emerging Patterns Discovery in Transaction Databases*

Pawel Terlecki and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology,
Nowowiejska 15/19, 00-665 Warsaw, Poland
{P.Terlecki, K.Walczak}@ii.pw.edu.pl

Abstract. This paper considers a rough set approach for the problem of finding minimal jumping emerging patterns (JEPs) in classified transactional datasets. The discovery is transformed into a series of transaction-wise local reduct computations. In order to decrease average subproblem dimensionality, we introduce local projection of a database. The novel algorithm is compared to the table condensation method and JEP-Producer for sparse and dense, originally relational data. For a more complete picture, in our experiments, different implementations of basic structures are considered.

Keywords: jumping emerging pattern, transaction database, local reduct, rough set, local projection.

1 Introduction

Pattern mining is one of key tasks in contemporary knowledge discovery. Although recent years have brought a wide spectrum of pattern types, discovery algorithms still follow common strategies such as Apriori, operations on concise representations ([1]), pattern trees ([2]). Regardless of a particular method, processing may involve exponentially large item set collections, which makes overall feasibility very sensitive to input data. Therefore, in our opinion, it is crucial to study, how to approach datasets of certain characteristics.

Here, we look at the problem of finding jumping emerging patterns (JEPs) in classified transaction databases. A JEP refers to an itemset that is supported in one class and absent from others. This highly discriminative idea was introduced in [3], and, since then, it has been successfully applied to business and gene expression problems. Because all JEPs constitute a convex space, the task is often perceived as finding minimal patterns. In fact, these patterns have found valuable applications to classification and clustering ([3]).

Among known algorithms, JEP-Producer ([1]) is believed to be the most efficient solution for finding a complete space of JEPs. It operates on concise representation of convex collections and employs a border differentiation operation to obtain a result set. In our previous works ([4]), it has been demonstrated

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

that reduces from the rough set theory ([5]) are closely related to JEPs. Algorithms based on these relations appeared superior in experiments for relational data ([6]). Moreover, even if data is originally given in a transactional form, a condensed decision table can be efficiently obtained by finding an approximate graph coloring in an item-conflict graph ([7]).

Following successful results for dense, originally relational data, we decided to examine opportunities for reduct-based methods against a popular class of sparse transaction databases. Note that, for large datasets, table condensation may likely deal with adverse item distribution in transactions, which results in low dimensionality reduction and inefficient discovery. The method of local projection that is put forward in this paper ascertains average dimensionality to depend only on average transaction length, not on item distribution in a database. The problem is decomposed into a series of per transaction local reduct computations in a locally projected decision table. For each subproblem only objects and attributes substantial for reduct induction are taken into account, which significantly improves overall efficiency. In addition, we propose several optimization to decrease a construction overhead of discernibility matrices.

Our experiments covered efficiency comparison between JEP-Producer, table condensation and local projection with different reduct computation methods. Approaches were tested against originally relational and sparse datasets. Since actual performance depends strongly on implementation, different structures to represent an attribute/item set were tested.

Section [2] provides fundamentals of emerging patterns and border representation. In Sect. [3] we present basic elements of the rough set theory. Local projection is introduced and proved correct in Sect. [4]. In Sect. [5] the novel algorithm is described. It also discusses optimizations for discernibility matrix computation and impact of different implementations of main structures. Section [6] covers testing procedure and experimental results. The paper is concluded in Sect. [7].

2 Emerging Patterns

Let a transaction system be a pair $(\mathcal{D}, \mathcal{I})$, where \mathcal{D} is a finite sequence of transactions (T_1, \dots, T_n) (database) such as $T_i \subseteq \mathcal{I}$ for $i = 1, \dots, n$ and \mathcal{I} is a non-empty set of items (itemspace). The support of an itemset $X \subseteq \mathcal{I}$ in a sequence $D = (T_i)_{i \in K} \subseteq \mathcal{D}$ is defined as $supp_D(X) = \frac{|\{i \in K : X \subseteq T_i\}|}{|K|}$, where $K \subseteq \{1, \dots, n\}$.

Let a decision transaction system be a tuple $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$, where $(\mathcal{D}, \mathcal{I} \cup \mathcal{I}_d)$ is a transaction system and $\forall T \in \mathcal{D} |T \cap \mathcal{I}_d| = 1$. Elements of \mathcal{I} and \mathcal{I}_d are called condition and decision items, respectively. Support in a decision transaction system $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$ is understood as support in the transaction system $(\mathcal{D}, \mathcal{I} \cup \mathcal{I}_d)$.

For each decision item $c \in \mathcal{I}_d$, we define a decision class sequence $C_c = (T_i)_{i \in K}$, where $K = \{k \in \{1, \dots, n\} : c \in T_k\}$. For convenience, the notations C_c and $C_{\{c\}}$ are used interchangeably. Note that each of the transactions from \mathcal{D} belongs to exactly one class sequence. In addition, for a database $D = (T_i)_{i \in K} \subseteq \mathcal{D}$, we define a complementary database $D' = (T_i)_{i \in \{1, \dots, n\} - K}$.

Given two databases $D_1, D_2 \subseteq \mathcal{D}$, in particular decision classes, we define a jumping emerging pattern (JEP) from D_1 to D_2 as an itemset $X \subseteq \mathcal{I}$ such as $supp_{D_1}(X) = 0$ and $supp_{D_2}(X) > 0$. A set of all JEPs from D_1 to D_2 is called a JEP space and denoted by $JEP(D_1, D_2)$.

JEP spaces can be described concisely by borders ([II]). For $c \in I_d$, we use a border $\langle \mathcal{L}_c, \mathcal{R}_c \rangle$ to uniquely represent a JEP space $JEP(C'_c, C_c)$. Members of the left bounds are minimal JEPs, whereas member of the right bounds are maximal JEPs, i.e. distinguishable transactions.

The problem of JEP discovery can be defined as computing $\{\langle \mathcal{L}_c, \mathcal{R}_c \rangle\}_{c \in I_d}$ for a decision transaction system $(\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$. Since finding of right bounds is trivial ([II]) and not interesting from a practical point of view, we focus on the collection of left bounds $\{\mathcal{L}_c\}_{c \in I_d}$.

3 Rough Sets

Let a decision table be a triple $(\mathcal{U}, \mathcal{C}, d)$, where \mathcal{U} (universum) is a non-empty, finite set of objects, \mathcal{C} is a non-empty finite set of condition attributes and d is a decision attribute. A set of all attributes is denoted by $\mathcal{A} = \mathcal{C} \cup \{d\}$. The domain of an attribute $a \in \mathcal{A}$ is denoted by V_a and its value for an object $u \in \mathcal{U}$ is denoted by $a(u)$. In particular, $V_d = \{c_1, \dots, c_{|V_d|}\}$ and the decision attribute induces a partition of \mathcal{U} into decision classes $\{U_c\}_{c \in V_d}$. Hereinafter, we use the term c to denote a condition attribute.

Consider $B \subseteq \mathcal{A}$. An indiscernibility relation $IND(B)$ is defined as $IND(B) = \{(u, v) \in \mathcal{U} \times \mathcal{U} : \forall a \in B \ a(u) = a(v)\}$. Since $IND(B)$ is an equivalence relation it induces a partition of \mathcal{U} denoted by $\mathcal{U}/IND(B)$. Let $B(u)$ be a block of the partition containing $u \in \mathcal{U}$. A B -lower approximation of a set $X \subseteq \mathcal{U}$ is defined as follows: $B_*(X) = \{u \in \mathcal{U} \mid B(u) \subseteq X\}$ and a B -positive region with respect to a decision attribute d is defined as $POS(B, d) = \bigcup_{X \in \mathcal{U}/IND(\{d\})} B_*(X)$.

A local reduct for an object $u \in \mathcal{U}$ is a minimal attribute set $B \subseteq \mathcal{C}$ such that $\forall c \in V_d (\mathcal{C}(u) \cap U_c = \emptyset \implies B(u) \cap U_c = \emptyset)$. It means that the object u can be differentiated by means of B from all objects from other classes as well as using \mathcal{C} . The set of all local reducts for an object u is denoted by $REDLOC(u, d)$.

4 Local Projection

In order to apply the rough set framework to transactional data, transformation to a respective relational form is required. We consider two representations: a binary decision table, which already found an application to negative pattern discovery ([IV]), and a locally projected form - introduced in this paper for efficient finding of positive patterns.

Hereinafter, we assume that our input data is represented by a decision transaction system $DTS = (\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$, where $\mathcal{D} = (T_1, \dots, T_n)$, $\mathcal{I} = \{I_1, \dots, I_m\}$, $\mathcal{I}_d = \{c_1, \dots, c_p\}$, $K = \{1, \dots, n\}$.

A binary decision table for a decision transaction system DTS is a decision table $BDT_{DTS} = (\mathcal{U}, \mathcal{C}, d)$ such that $\mathcal{U} = \{u_1, \dots, u_n\}$, $\mathcal{C} = \{a_1, \dots, a_m\}$, $V_d = \{c_1, \dots, c_p\}$; $a_j(u_i) = \begin{cases} 0, & I_j \notin T_i \\ 1, & I_j \in T_i \end{cases}, \forall i \in 1..n, j \in 1..m$; $d(u_i) = T_i \cap \mathcal{I}_d, \forall i \in 1..n$.

Local reducts in a binary decision table correspond to jumping emerging patterns with negation (JEPNs, [4]). JEPNs constitute a convex space that contains JEPs for the same transaction system. Note that, although $\{e, g\}$ and $\{d, f\}$ are both local reducts for u_1 , the pattern eg is a minimal JEP, whereas df is not, since it is not supported by the respective transaction.

Solving a problem of a double dimensionality and filtering positive patterns is most often expensive, thus, the idea of a table condensation was proposed ([7]). Before local reduct computation, binary attributes are aggregated into multi-valued attributes by means of an approximate graph coloring. This approach is efficient for originally relational datasets, however, remains sensitive to a distribution of items in transactions.

The table condensation leads to an alternative representation of a decision transaction system. However, one may get much higher complexity reduction if transformation is performed independently for every transaction. The following structure demonstrates how we may limit our interest only to items that are indispensable to compute complete discernibility information for a transaction.

A locally projected decision table for: DTS , a decision transaction system, and $T_i \in \mathcal{D}$, where $i = 1, \dots, |\mathcal{D}|$, a transaction, is a binary decision table $LPDT_{DTS, T_i} = BDT_{DTS, T_i}$, where $DTS_i = (\mathcal{D}_i, \mathcal{T}_i, \mathcal{I}_d)$ and $\mathcal{D}_i = (T_k \cap T_i)_{k \in K}$.

Hardness of an input decision system DTS can be characterized by average (maximal) dimensionality of subproblems, i.e. a locally projected decision table for distinguishable transactions, namely $avgDim(DTS) = \{|T| : T \in \mathcal{R}_c \wedge c \in \mathcal{I}_d\} / \sum_{c \in \mathcal{I}_d} |\mathcal{R}_c|$ and $maxDim(DTS) = max_{T \in \mathcal{R}_c \wedge c \in \mathcal{I}_d} |T|$. Note that, when all transactions are distinguishable, these parameters refer to an average (maximal) transaction length DTS .

For the sake of convenience, we use the notation: $itemPatt_{DTS, T_i}(u, B) = \{I_k \in T_i : a_k \in B \wedge a_k(u) = 1 \wedge k \in M'\}$, where $u \in \mathcal{U}$, $B \subseteq \mathcal{C}_i = \{a_k\}_{k \in M'}$, $LPDT_{DTS, T_i} = (\mathcal{U}, \mathcal{C}_i, d)$ is a locally projected decision table and a transaction $T_i = \{I_k\}_{k \in M'}$, $M' \subseteq \{1, \dots, m\}$. Note that $|itemPatt_{DTS, T_i}(u_i, B)| = |B|$. Whenever a decision transaction system is known from the context, the respective subscript is omitted.

The following theorem states that the complete JEP space for the DTS and a given class can be obtained by finding a locally projected tables for each distinguishable transaction and generating patterns for the respective objects and any attribute set in the respective table.

Theorem 1. $\forall_{c \in \mathcal{I}_d} \{itemPatt_{DTS, T_i}(u_i, R) : i \in K \wedge LPDT_{DTS, T_i} = (\mathcal{U}, \mathcal{C}_i, d) \wedge u_i \in POS(\mathcal{C}_i, d) \cap U_c \wedge R \subseteq \mathcal{C}_i\} = JEP(\mathcal{C}'_c, \mathcal{C}_c)$

The respective left bound of a JEP space can be found by applying local reducts for a given object instead of any attribute sets.

Theorem 2. $\forall_{c \in \mathcal{I}_d} \{itemPatt_{DTS, T_i}(u_i, R) : i \in K \wedge LPDT_{DTS, T_i} = (U, \mathcal{C}_i, d) \wedge u_i \in POS(\mathcal{C}_i, d) \cap U_c \wedge R \in REDLOC(u_i, d)\} = \mathcal{L}_c$

The proofs are omitted here due to space limitations.

5 JEP Computation

Minimal jumping emerging patterns in $DTS = (\mathcal{D}, \mathcal{I}, \mathcal{I}_d)$ can be computed by local reduct computation in locally condensed tables for all transactions. The actual procedure is straightforward and fully based on Theorem 2.

- 1: $\mathcal{L}_c = \emptyset$ for each $c \in \mathcal{I}_d$
- 2: **for** ($k = 1; 1 \leq |\mathcal{D}|; k++$) **do**
- 3: Construct a locally projected decision table $LPDT_{DTS, T_k}$
- 4: Compute $REDLOC(u_k, d)$ in $LPDT_{DTS, T_k}$
- 5: $\mathcal{L}_c = \mathcal{L}_c \cup \{itemPatt_{DTS, T_k}(u_k, R) : R \in REDLOC(u_k, d)\}, c = T_k \cup \mathcal{I}_d$
- 6: **end for**

Identification of minimal patterns by means of local reduct induction is the most complex part of our approach. It is normally addressed with methods used for global reducts (5). Unfortunately, all known exact solutions are pessimistically exponential.

Here, we look at two algorithms that employ a discernibility matrix. The first one reduces the problem to finding prime implicants of a monotonous boolean functions (5, RedPrime). It loops over elements of a matrix and extends a collection of reducts for rows seen so far, so that they are sufficient to discern the current row as well. The second algorithm traverses a lattice of all subsets of an attribute space using the apriori scheme (8, RedApriori). Successive collections of candidates are pruned basing on a degree of attribute set dependence, which is calculated by means of a discernibility matrix. Also, in order to optimize this stage, one may eliminate transactions that are not maximal JEPs and group transactions by their classes.

6 Experimental Results

Experiments focused on efficiency of the new algorithm, table condensation and JEP-Producer for synthetically generated sparse data and dense data obtained from relational tables. Each result was averaged over several executions.

The testing environment and algorithms were coded in Java 5. Since the rough set methods and JEP-Producer differ significantly, it is not possible to come up with one single dominant operation for time complexity representation. Therefore, in order to provide reliable time measurements, we based their implementations on mostly the same structures. In particular, all the studied approaches process large collections of attribute/item sets. To obtain results possibly independent from what a data structure was used to represent such a set, three

implementations were tested. The first two are characteristic vectors of an attributes/item space, one based on a regular byte array (Array) and the other one - on java.util.BitSet (BitSet). The third structure is a balanced binary tree implemented by means of java.util.TreeSet (TreeSet). Array is generous in memory allocation, but assures the most efficient access. Bit and dynamic structures are slower, however, they may take precedence for large attribute/item spaces when a high number of sets is created.

Table 1. Synthetic dataset summary with problem hardness characteristics

No	Trans	Items	Classes	MaxTrans	JEPs	avgDim	maxDim
1	2000	20	2	326	539	5,09	9,00
2	2000	40	3	1075	5967	6,33	14,00
3	2000	60	3	1551	19140	6,79	16,00
4	2000	80	2	1858	71250	9,37	19,00
5	5000	50	2	2918	20088	6.25	15.00
6	10000	50	3	4119	18673	5.57	12.00
7	15000	50	2	7920	94252	7.57	18.00
8	20000	50	2	10300	126162	7.63	18.00

Sparse Data. In this test local projection and JEP-Producer are compared for sparse datasets. Since itemspaces are commonly much larger than average transaction size, this kind of data is substantial for practical tasks. Unfortunately, it is hard to find publicly available classified sparse datasets, thus, the test was performed against synthetic data. Transaction databases were produced by means of the IBM generator ([9]) and, then, the CLUTO package was used to classify transactions (Tab. 1). The density of each database was set up at 5-15% of a respective item space. We studied behavior of the algorithms when a size of a database or an item space increases. In order to describe the actual hardness of each problem, additional measures were provided, in particular, a total number of JEPs over all classes, number of maximal transactions and average (maximal) dimensionality.

The local projection algorithm was tested with two different reduct computation methods: RedPrime ([5]) and RedApriori ([8]). JEP-Producer was implemented according to the scheme and optimizations described in [1]. To optimize all computations a database is always reduced to contain only maximal transactions. Measurements for all the algorithms were taken for the aforementioned implementations of an attribute/item set.

Table 2 shows that the rough set approach outperforms JEP-Producer. In particular, for RedApriori and Array, there is a difference of 1-2 orders of magnitude. In general, all the methods perform well for Array. Reduct computations are performed for locally projected tables with small attribute spaces, thus, slower structures significantly affect the overall performance. For example, for TreeSet, efficiency of RedPrime and JEP-Producer remain very close. On the other hand, JEP-Producer is sensitive to the size of a whole item space, therefore, BitSet led to slightly better results in almost all the cases.

Table 2. Execution time comparison for sparse datasets between local projection and JEP-Producer for different implementations of an attribute/item set

No	RedPrime			RedApriori			JEP-Producer		
	Array	BitSet	TreeSet	Array	BitSet	TreeSet	Array	BitSet	TreeSet
1	297	484	797	138	218	470	594	640	1296
2	4906	8938	19063	1796	3196	10262	14375	13469	28547
3	20453	34860	78406	4553	8120	29351	53281	47250	93562
4	202328	323296	810360	48469	77129	340541	217796	164594	346265
5	26532	45219	95203	6573	10669	36239	118453	97735	200094
6	28750	55906	104812	4071	7608	21565	215250	190906	390937
7	671390	1123562	2739329	162874	263418	1033108	1311203	1169141	2623266
8	877655	1468744	3582734	243205	393338	1490587	2153109	1982421	4316875

Originally Relational Data. Earlier tests demonstrated that, for dense, originally relational datasets, condensation successfully reduces dimensionality and performs better than JEP-Producer (6). Here, it is contrasted with local projection. Due to space limitations, results for RedPrime and Array-based attribute set implementation are presented. Transactional databases for this test were generated from relational tables from UCI Repository. Average time and dimensionality is given for each of the methods.

According to the results in Tab. 3, table condensation and local projection lead most often to the same subproblem dimensionality. Since databases are reduced in an analogical way, both methods achieve similar efficiency. Nevertheless, the former strongly relies on optimality of graph coloring solution. An overhead of generation and filtering of additional patterns is visible for

Table 3. Execution time comparison for originally relational datasets between table condensation and local projection with RedPrime and Array-based implementation

Dataset	Trans	Items	Class	JEPs	Table Condensation		Local Projection	
					avgDim	Time	avgDim	Time
lymn	148	59	4	6794	18.00	13156	18.00	8359
house	435	48	2	6986	16.00	27218	16.00	21141
balance	625	20	3	303	4.00	671	4.00	406
tic-tac-toe	958	27	2	2858	9.00	9203	9.00	7578
car-mod	1728	21	4	246	6.00	4109	6.00	3985
mushroom	8124	117	2	3635	23.00	608546	22.00	440656
nursery	12960	27	5	638	8.00	477484	8.00	495375
krkopt	28056	43	18	21370	6.00	1754469	6.00	1728782

7 Conclusions

In this paper we have proposed a rough set approach to discovery of jumping emerging patterns (JEPs) in classified transaction databases. The problem is

decomposed into a series of local reduct computations performed for locally projected decision tables for each transaction.

Main benefit of our approach is that only the transactions and items necessary for each computation are considered, which results in potentially significant dimensionality reduction of subproblems. In this case, additional processing can be a significant factor. The way of discernibility matrices construction can be optimized by caching of partial per-attribute results in complementary form.

Experiments have proved that the method outperforms JEP-Producer, the most popular solution for the considered problem, for sparse, synthetically generated datasets. The high efficiency is a result of a dramatic decrease in average dimensionality. This fact was observed independently from a reduct computation method. Nevertheless, the algorithm based on attribute set dependence behaves much better than the classical one searching for prime implicants and is faster than JEP-Producer by 1-2 orders of magnitude. On the other hand, for dense, originally relational data the new approach achieves at least the same dimensionality gain as the previously proposed method of table condensation and gives similar overall efficiency.

The future research will extend our method to look for derivative types of patterns and confront its efficiency with existing tree-based strategies.

References

1. Dong, G., Li, J.: Mining border descriptions of emerging patterns from dataset pairs. *Knowl. Inf. Syst.* 8(2), 178–202 (2005)
2. Li, J., Liu, G., Wong, L.: Mining statistically important equivalence classes and delta-discriminative emerging patterns. In: *KDD 2007*, pp. 430–439 (2007)
3. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. *Kn. In. Sys.* 3(2), 1–29 (2001)
4. Terlecki, P., Walczak, K.: Jumping emerging patterns with negation in transaction databases - classification and discovery. *Information Sciences* 177, 5675–5690 (2007)
5. Bazan, J., Nguyen, H.S., Nguyen, S.H., Synak, P., Wroblewski, J.: Rough set algorithms in classification problem. *Rough set methods and applications: new develop. in knowl. disc. in inf. syst.*, 49–88 (2000)
6. Terlecki, P., Walczak, K.: Local reducts and jumping emerging patterns in relational databases. In: Greco, S., Hata, Y., Hirano, S., Inuiguchi, M., Miyamoto, S., Nguyen, H.S., Słowiński, R. (eds.) *RSCTC 2006*. LNCS (LNAI), vol. 4259, pp. 358–367. Springer, Heidelberg (2006)
7. Terlecki, P., Walczak, K.: Jumping emerging pattern induction by means of graph coloring and local reducts in transaction databases. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) *RSFDGrC 2007*. LNCS (LNAI), vol. 4482, pp. 363–370. Springer, Heidelberg (2007)
8. Terlecki, P., Walczak, K.: Attribute set dependence in apriori-like reduct computation. In: Wang, G.-Y., Peters, J.F., Skowron, A., Yao, Y. (eds.) *RSKT 2006*. LNCS (LNAI), vol. 4062, pp. 268–276. Springer, Heidelberg (2006)
9. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: *VLDB 1994*, pp. 487–499 (1994)

Applying Latent Semantic Indexing in Frequent Itemset Mining for Document Relation Discovery

Thanaruk Theeramunkong¹, Kritsada Sriphaew^{1,2}, and Manabu Okumura²

¹ Sirindhorn International Institute of Technology, Thammasat University, 131 Moo 5 Tiwanont Rd., Bangkadi, Muang, Pathumthani 12000, Thailand
thanaruk@siit.tu.ac.th, kong@siit.tu.ac.th

² Precision and Intelligence Laboratory, Tokyo Institute of Technology, 4259 Nagatsuta Midori-ku Yokohama 226-8503, Japan
oku@pi.titech.ac.jp

Abstract. Word-based relations among technical documents are immensely useful information but often hidden in a large amount of scientific publications. This work presents a method to apply latent semantic indexing in frequent itemset mining to discover potential relations among scientific publications. In this work, two weighting schemes, tf and tfidf are investigated with the exploitation of latent semantic indexing. The proposed method is evaluated using a set of technical documents in a publication database by comparing the extracted document relations with their references (citations). To this end, the paper uses order accumulative citation matrices to evaluate the validity (quality) of discovered patterns. The results also show that the proposed method successfully discovers a set of document relations, comparing to the original method that uses no latent semantic indexing.

1 Introduction

Fast increasing of research publication has caused the difficulty for researchers to grasp movement or change in their area of interest. Such information overload becomes serious hindrance for researchers to position their own works against existing ones, or to find useful relations (or connections) among them. Although the publication of each work may include a list of related articles (documents) as its reference (called citation), it is still impossible to include all related works due to either intentional reasons (e.g., limitation of paper length) or unintentional reasons (e.g., naïvely unknown). Enormous meaningful connections that permeate the literatures may remain hidden. Recently, there have been two different approaches to find relations among research documents. As the first approach, the citation-based method uses expansion of bibliography or citation information in scientific publication to find indirect relations, including measurement of impact factor [1], characterization of the citation [2], support of browsing citation graph [3] and so forth. For the task of relation discovery, two basic properties of citation, called bibliographic coupling [4] and co-citation [5], can

be focused. Those previous works stated that any two documents tend to have relation with each other if they are citing to one or more documents in common (bibliographic coupling) or they are both cited by one or more documents in common (co-citation). As the second approach, the word- or term-based method exploits words or terms in a document as potential clues to detect relations between the document and other related documents. This method (later called word-based approach) discovers a set of documents with similar contents (topics) using either word co-occurrences or shared vocabularies, such as done in information retrieval, text categorization and text clustering. However, the process to find relations among two documents is computationally expensive since all combinations need to be considered for any possible relation [6]. Towards this problem, some recent works [7,8] have applied association rule mining (ARM) techniques to find n-ary document relations where a support can be set to avoid exploring all document combinations. Even such works could achieve discovery of high-quality relations to some extents, they still have some limitations due to direct use of words and terms in documents.

In this paper, we propose a method to apply latent semantic indexing in the process of discovering hidden relations among two documents. Two main objectives are (1) to study how well the word-based approach with different weighting (tf and tfidf) performs in finding relations among documents using ARM techniques, and (2) to study how much latent semantic indexing improves the conventional approach in finding useful hidden relations.

2 Frequent Itemset Mining

In the past, association rule mining (ARM) and frequent itemset mining (FIM) was known as a process to find co-occurrences (frequent patterns) in a database. In general, the conventional transactional database is presented in the term of item existences in the transaction. Although most ARM works deal with a this kind of databases, there are some attempts to extend the original framework to be able to assign the weights for items or transactions in the database, called weighted association rule mining [9]. In those works, items or transactions are independently weighted regarding to which type of discovered rules we would like to find. The higher weighted items or transactions will obtain higher priority for user interests. However, this approach gives a fixed weight to each item regardless of the transaction such item occurs. Unlike those works, our approach utilizes the term-document orientations, where the discovered frequent itemset is a set of documents which share a large number of terms as done in [7,8]. Note that a transaction corresponds to a term while an item corresponds to a document. Therefore, a “docset” (document set) is used in place of the term “itemset” in the traditional FIM approaches. The discovered results can be assumed as a term-based relation among documents where the relation is introduced by coincident terms. In Figure 1, two examples of the real-valued databases are defined in the form of well-known vector space model (VSM). The left part indicates how often

	d_1	d_2	d_3	d_4
t_1	4	2	0	0
t_2	4	2	4	0
t_3	2	0	2	2
t_4	0	4	0	1

	d_1	d_2	d_3	d_4
t_1	$4 \times \log 4/2 = 1.20$	$2 \times \log 4/2 = 0.60$	$0 \times \log 4/2 = 0.00$	$0 \times \log 4/2 = 0.00$
t_2	$4 \times \log 4/3 = 0.50$	$2 \times \log 4/3 = 0.25$	$4 \times \log 4/3 = 0.50$	$0 \times \log 4/3 = 0.00$
t_3	$2 \times \log 4/3 = 0.25$	$0 \times \log 4/3 = 0.00$	$2 \times \log 4/3 = 0.25$	$2 \times \log 4/3 = 0.25$
t_4	$0 \times \log 4/2 = 0.00$	$4 \times \log 4/2 = 1.20$	$0 \times \log 4/2 = 0.00$	$1 \times \log 4/2 = 0.30$

Fig. 1. the term-document database with tf (left) and tfidf (right) term weightings

a term occurs in each document (called term frequency - tf) while the right part shows term frequency multiplied by the inverse document frequency (tfidf).

Traditionally, the support of a docset is defined by a ratio between the number of terms that exist in all documents in the docset and the total number of distinct terms in a database. To expand this concept to a real-valued database, the definition of support is generalized as follows. Let \mathcal{D} be a set of documents (items) where $\mathcal{D} = \{d_1, d_2, \dots, d_m\}$, and \mathcal{T} be a set of terms (transactions) where $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$. Also let $w(d_i, t_j)$ represent a weight of a term t_j in a document d_i . A subset of \mathcal{D} is called a docset whereas a subset of \mathcal{T} is called a termset. Furthermore, a docset $X_k = \{x_1, x_2, \dots, x_k\} \subset \mathcal{D}$ with k documents is called k -docset. The support of X_k is defined as follows.

$$sup(X_k) = \frac{\sum_{j=1}^n \min_{i=1}^k w(x_i, t_j)}{\sum_{j=1}^n \max_{i=1}^m w(d_i, t_j)}$$

By representing the data to be mined as shown in Figure 1, the new definition of support employs the \min operation to find the weight of each term for a docset by selecting a minimum weight of such term among all documents in the docset. The \max operation is applied for finding the maximum weight of each term in the database. The support of a docset will then be calculated from the ratio between the sum of all term weights for a docset and the sum of maximum weights of all terms in the database. While this definition can be applied for general real-valued databases, it also can be used for the traditional FIM on boolean-valued databases with the same result. An example of docsets and their supports, for tf and tfidf databases, can be computed as shown in Figure 2. Besides support, a so-called confidence is used for generating confident association rules. Here, the confidence is left since it is out of scope in this work. Note that similar to conventional ARM, these generalized supports preserve two closure properties, i.e., downward closure property (“all subsets of a frequent itemset are also frequent”), and upward closure property (“all supersets of an infrequent itemset are also infrequent”). For example, $sup(d_1) \geq sup(d_1 d_2)$ and $sup(d_2) \geq sup(d_1 d_2)$. The mathematical proof can be found in [8].

3 Representation and Latent Semantic Indexing

To represent document representation, term weighting can be performed to set importance level of a term in a document. This work uses two most common non-binary weightings: term-frequency (tf) and term-frequency-inverse-document-frequency (tfidf). Moreover, latent semantic indexing is applied to reveal hidden meaning in a document or a query. In this latent semantic space, a query and a

Docset	Generalized support	
	tf	tfidf
$\{d_1\}$	10/14 = 0.71	1.95/3.15 = 0.62
$\{d_2\}$	8/14 = 0.57	2.05/3.15 = 0.65
$\{d_3\}$	6/14 = 0.43	0.75/3.15 = 0.24
$\{d_4\}$	3/14 = 0.21	0.55/3.15 = 0.17
$\{d_1 d_2\}$	4/14 = 0.29	0.85/3.15 = 0.27
$\{d_1 d_3\}$	6/14 = 0.43	0.75/3.15 = 0.24
$\{d_1 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08

Docset	Generalized support	
	tf	tfidf
$\{d_2 d_3\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_2 d_4\}$	1/14 = 0.07	0.30/3.15 = 0.10
$\{d_3 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_3\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00
$\{d_2 d_3 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00
$\{d_1 d_3 d_4\}$	2/14 = 0.14	0.25/3.15 = 0.08
$\{d_1 d_2 d_3 d_4\}$	0/14 = 0.00	0.00/3.15 = 0.00

Fig. 2. Docsets and their generalized supports (tf vs. tfidf)

document may have high cosine similarity even if they do not share any common words or terms but their terms are semantically similar. Applied the concept of Singular Value Decomposition (SVD), LSI can also be viewed as a method for dimensionality reduction by a least-squared method [10]. SVD (also LSI) translates an input matrix A and represents it as A' in a lower dimensional space such that the 'distance' between the two matrices as measured by minimizing the 2-norm (Euclidean distance), $\|A - A'\|_2$. It is possible to project an n -dimensional space of word-document matrices onto a k -dimensional space where n is the number of word types in the collection and k is relatively very small compared to n , say 100 and 150. The SVD projection is done by decomposing a document-by-term matrix $A_{t \times d}$ into the product of three matrices, $T_{t \times n}$, $S_{n \times n}$ and $D_{d \times n}$ as follows.

$$A_{t \times d} = T_{t \times n} \times S_{n \times n} \times D_{d \times n}^T$$

Here, t is the number of terms, d is the number of documents, $n = \min(t, d)$, T and D have orthonormal columns, i.e. $T \times T^T = I$ and $D^T \times D = I$, and S is a diagonal matrix, where $s_{i, j} = 0$ for $i \neq j$. Moreover, in some situations $rank(A) = r$ where $r \leq n$. In these situations, the diagonal elements of S are $\sigma_1, \sigma_2, \dots, \sigma_n$ where $\sigma_i > 0$ for $1 \leq i \leq r$ and $\sigma_i = 0$ for $r < i \leq n$. For details of how to derive $T_{t \times n}$, $S_{n \times n}$ and $D_{d \times n}$, can be found in [10]. In this work, we investigate the best combination of the four schemes.

4 The Evaluation Method

To evaluate the result, we introduce an automatic evaluation where citation graph is used to evaluate our system based on its ability to find the relations that exist in the citation graph. Although human judgment is the best method for evaluation, it is a labor-intensive and time-consuming task. To do this, a citation graph is applied. Conceptually citations among documents in scientific publication collection form a citation graph, where a node corresponds to a document and an arc corresponds to a direct citation of a document to another document. Based on this citation graph, an indirect citation can be defined using the concept of transitivity. The formulation of direct and indirect citations can be given in the terms of the u -th order citation and the v -th order accumulative citation matrix as follows.

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	0	0	0	0
d_2	1	1	1	0	1	0
d_3	0	1	1	1	1	0
d_4	0	0	1	1	0	1
d_5	0	1	1	0	1	0
d_6	0	0	0	1	0	1

1-OACM

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	1	0	1	0
d_2	1	1	1	1	1	0
d_3	1	1	1	1	1	1
d_4	0	1	1	1	1	1
d_5	1	1	1	1	1	0
d_6	0	0	1	1	0	1

2-OACM

doc.	d_1	d_2	d_3	d_4	d_5	d_6
d_1	1	1	1	1	1	0
d_2	1	1	1	1	1	1
d_3	1	1	1	1	1	1
d_4	1	1	1	1	1	1
d_5	1	1	1	1	1	1
d_6	0	1	1	1	1	1

3-OACM

Fig. 3. The 1-, 2- and 3-OACMs

Definition 1 (the u -th order citation). For $x, y \in \mathcal{D}$, y is the u -th order citation of x iff the number of arcs in the shortest path between x to y in the citation graph is u (≥ 1). Conversely, x is called the u -th order citation of y .

Definition 2 (the v -th order accumulative citation matrix). Given a set of n distinct documents, the v -th order accumulative citation matrix (for short, v -OACM) is an $n \times n$ matrix, each element of which represents the citation relation δ^v between two documents x, y where $\delta^v(x, y) = 1$ when x is the u -th order citation of y and $u \leq v$, otherwise $\delta^v(x, y) = 0$. Note that $\delta^v(x, y) = \delta^v(y, x)$ and $\delta^v(x, x) = 1$.

For example, given a set of six documents $d_1, d_2, d_3, d_4, d_5, d_6 \in \mathcal{D}$ and a set of six citations d_1 to d_2 , d_2 to d_3 and d_5, d_3 to d_5 , and d_4 to d_3 and d_6, d_2 is the first, d_3 and d_5 is the second, d_4 is the third, and d_6 is the fourth order citations of the document d_1 . The 1-, 2- and 3-OACMs can be created as shown in Figure 3. The 1-OACM can be straightforwardly constructed from the set of the first-order citation (direct citation). The $(v + 1)$ -OACM (mathematically denoted by a matrix A^{v+1}) can be recursively created from the operation between v -OACM (A^v) and 1-OACM (A^1) according to the following formula.

$$a_{ij}^{v+1} = \vee_{k=1}^n (a_{ik}^v \wedge a_{kj}^1) \tag{1}$$

where \vee is an OR operator, \wedge is an AND operator, a_{ik}^v is the element at the i -th row and k -th column of the matrix A^v and a_{kj}^1 is the element at the k -th row and j -th column of the matrix A^1 . Here, a v -OACM is a symmetric matrix.

The shorter the specific range is, the more restrict the evaluation is. With the concept of v -OACM stated in the previous section, we can realize this generalized evaluation by a so-called v -th order validity (for short, v -validity), where v corresponds to the specific range mentioned above. The formulation of the v -validity of a docset X ($X \subset \mathcal{D}$), denoted by $S^v(X)$, is defined as follows.

$$S^v(X) = \frac{\max_{x \in X} (\sum_{y \in X, y \neq x} \delta^v(x, y))}{|X| - 1} \tag{2}$$

Here, $\delta^v(x, y)$ is the citation relation defined by Definition 2. In the equation, we can observe that the v -validity of a docset is ranging from 0 to 1, i.e., $0 \leq S^v(X) \leq 1$. The v -validity achieves the minimum (i.e., 0) when there is no

citation relation among any document in the docset. On the other hand, it achieves the maximum (i.e., 1) when there is at least one document that has a citation relation with all documents in a docset. Intuitively, the validity of a bigger docset tends to have lower validity than a smaller one. Moreover, given a set of discovered docsets \mathcal{F} , its v -validity (later called v -validity), denoted by $\overline{\mathcal{S}}^v(\mathcal{F})$, can be defined as follows.

$$\overline{\mathcal{S}}^v(\mathcal{F}) = \frac{\sum_{X \in \mathcal{F}} w_X \times \mathcal{S}^v(X)}{\sum_{X \in \mathcal{F}} w_X} \quad (3)$$

where w_X is the weight of a docset (X). In this work, w_X is set to $|X| - 1$, the maximum value that the validity of a docset X can gain. For example, given the 1-OACM in Figure 3 and $\mathcal{F} = \{d_1 d_2, d_1 d_2 d_3\}$, the set 1-validity of \mathcal{F} (i.e., $\overline{\mathcal{S}}^1(\mathcal{F})$) equals to $\frac{(1 \times \frac{1}{1}) + (2 \times \frac{2}{2})}{1+2} = \frac{3}{3} = 1$.

5 Experimental Settings and Results

A set of experiments are made to investigate how efficiently universal frequent itemset mining helps in discovering document relation among scientific research publications. In this work, an evaluation material is constructed from a collection of scientific research publications in the ACM Digital Library¹. This dataset was originally used in [7]. As a seed of evaluation dataset, 200 publications are retrieved from each of the three computer-related classes, coded by B (Hardware), E (Data) and J (Computer) classes. Then the publications referred by these newly collected publications are also gathered and appended into the dataset. In total there are 10,817 publications collected as the evaluation material and used to generate citation graph under 1-OACM. As the result, only 36,626 citation edges are remained with an average of 7 citations (including both cite to and cited from other publications) per publication. For mining, we applied FP-tree algorithm, originally introduced in [11] and used the BOW library [12] as a tool for constructing an attribute-value database. The 524 stopwords and terms with very low frequency (less than 3 times) are omitted. Table 1 shows the validity of discovered document relations when either tf or tfidf are considered and LSI is applied with a thresholds of either 0.5, 0.7 or 1.0.

From the result shown in Table 1, some interesting characteristics can be observed. First, in most cases of the original space (w/o LSI), tfidf performs better than tf even there are few exceptions. The result implies that tfidf helps us obtain good representation for document relation discovery. Moreover, the result of 1-OACM becomes lower when N increases. This implies that better relations are located at higher ranks. In addition, with a higher-OACM, the method can achieve up to 90-100 % validity and has the same trend that the validity drops when N increases. Second, for both tf and tfidf, the 1-OACM performance of discovering document relations improves from 14.29 % to around 40 % for top-1000 documents when LSI is applied. Focusing on the 2-OACM and

¹ <http://www.portal.acm.org>

Table 1. Set validity of top-N rankings of discovered docsets when either tf or tfidf is used and LSI is applied with a thresholds of either 0.5, 0.7 or 1.0

Methods	N	1-OACM		2-OACM		3-OACM	
		tf	tfidf	tf	tfidf	tf	tfidf
w/o LSI	1000	14.29	25.00	85.71	100.00	100.00	100.00
	5000	37.59	38.03	87.23	95.77	95.62	97.18
	10000	18.22	38.97	58.94	87.66	87.13	93.81
	50000	6.16	16.24	35.91	60.52	75.68	94.05
	100000	4.37	14.36	31.22	55.83	74.49	93.08
LSI $_{\delta=0.5}$	1000	41.51	42.86	90.57	85.71	94.34	91.43
	5000	23.80	25.90	66.47	67.94	84.01	83.76
	10000	19.92	23.01	64.44	67.26	86.06	85.02
	50000	14.12	17.89	59.80	64.13	90.15	89.13
	100000	11.40	14.48	56.81	60.57	90.39	90.13
LSI $_{\delta=0.7}$	1000	47.14	44.15	90.00	80.32	95.71	85.64
	5000	25.95	28.28	69.09	70.86	85.98	85.72
	10000	22.26	25.59	67.80	70.64	87.52	86.95
	50000	14.77	19.91	60.76	66.72	91.43	91.27
	100000	12.09	16.06	57.51	61.73	91.52	90.98
LSI $_{\delta=1.0}$	1000	44.68	45.42	85.11	81.25	90.43	87.08
	5000	26.55	28.95	70.23	71.42	86.86	86.43
	10000	23.67	27.85	69.27	72.66	88.54	89.15
	50000	15.27	19.79	61.05	66.58	91.75	91.29
	100000	12.53	16.45	57.35	62.03	91.67	91.90

3-OACM performance, LSI is helpful to improve the validity of the discovered relations, especially for the cases of tf. In the cases of tfidf, LSI is helpful to improve validity of discovered document relations especially in the case of the 1-OACM. However, it is not useful for the 2-OACM and 3-OACM performance. This implies that LSI is helpful to increase the performance of discovering direct citations but not indirect citations. One implication is that the tfidf seems to be a good representation. Third, a stronger LSI (LSI with a higher threshold) performs better than a softer LSI (LSI with a lower threshold). This implies that LSI is useful to grasp the semantics of documents and then help increasing the discovery performance.

6 Conclusions

This work presents a new approach to discover document relations using association rule mining techniques with latent semantic indexing. Extended from the conventional frequent itemset mining, a so-called generalized support is proposed. The generalized support can serve a mining process of frequent itemsets from an attribute-value database where the values are weighted by real values, instead of boolean values as done in conventional methods. The quality of discovered document relations is measured under the concepts of the u -th order citation and the v -th order accumulative citation matrix. By experiments, we found out that tfidf seems better than tf and latent semantic indexing is helpful in discovering meaningful document relations. As future works, it is necessary

to explore other suitable term weightings and normalization techniques. More explorations are needed for different data collections.

Acknowledgement

This work has been supported by Thailand Research Fund under project number BRG50800013. This work was also supported by NECTEC under project number NT-B-22-I4-38-49-05 and Royal Golden Jubilee (RGJ) Ph.D. program of the Thailand Research Fund (TRF).

References

1. Garfield, E.: Citation analysis as a tool in journal evaluation. *Science* 178(4060), 471–479 (1972)
2. An, Y., Janssen, J., Milios, E.E.: Characterizing and mining the citation graph of the computer science literature. *Knowl. Inf. Syst.* 6(6), 664–678 (2004)
3. Chen, C.: visualising semantic spaces and author co-citation networks in digital libraries. *Information Processing and Management* 35(3), 401–420 (1999)
4. Kessler, M.M.: Bibliographic coupling between scientific papers. *American Documentation* 14, 10–25 (1963)
5. Small, H.: Co-Citation in the scientific literature: a new measure of the relationship between documents. *Journal of the American Society for Information Science* 42, 676–684 (1973)
6. Theeramunkong, T.: Applying passage in web text mining. *Int. J. Intell. Syst.* (1-2), 149–158 (2004)
7. Sriphaew, K., Theeramunkong, T.: Revealing topic-based relationship among documents using association rule mining. *Artificial Intelligence and Applications*, 112–117 (2005)
8. Sriphaew, K., Theeramunkong, T.: Quality evaluation from document relation discovery using citation information. *IEICE Transactions on Information and Systems* E90-D(8), 1131–1140 (2007)
9. Yun, U., Leggett, J.J.: Wip: mining weighted interesting patterns with a strong weight and/or support affinity. In: *Proceedings of 2006 SIAM Conference on Data Mining*, pp. 623–627. IEEE Computer Society Press, Bethesda, Maryland, USA (2006)
10. Deerwester, S.C., Dumais, S.T., Landauer, T.K., Furnas, G.W., Harshman, R.A.: Indexing by latent semantic analysis. *Journal of the American Society for Information Science* 41, 391–407 (1990)
11. Han, J., Pei, J., Yin, Y.: Mining frequent patterns without candidate generation. In: Chen, W., Naughton, J., Bernstein, P.A. (eds.) *2000 ACM SIGMOD Intl. Conference on Management of Data*, pp. 1–12. ACM Press, New York (2000)
12. McCallum, A.K.: *Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering* (1996)

G-TREACLE: A New Grid-Based and Tree-Alike Pattern Clustering Technique for Large Databases

Cheng-Fa Tsai and Chia-Chen Yen

Department of Management Information Systems,
National Pingtung University of Science and Technology,
91201 Pingtung, Taiwan
{cftsai,m9556001}@mail.npust.edu.tw

Abstract. As data mining having attracted a significant amount of research attention, many clustering methods have been proposed in past decades. However, most of those techniques have annoying obstacles in precise pattern recognition. This paper presents a new clustering algorithm termed G-TREACLE, which can fulfill numerous clustering requirements in data mining applications. As a hybrid approach that adopts grid-based concept, the proposed algorithm recognizes the solid framework of clusters and, then, identifies the arbitrary edge of clusters by utilization of a new density-based expansion process, which named “tree-alike pattern”. Experimental results illustrate that the new algorithm precisely recognizes the whole cluster, and efficiently reduces the problem of high computational time. It also indicates that the proposed new clustering algorithm performs better than several existing well-known approaches such as the K-means, DBSCAN, CLIQUE and GDH algorithms, while produces much smaller errors than the K-means, DBSCAN, CLIQUE and GDH approaches in most the cases examined herein.

Keywords: data clustering, data mining, hybrid clustering algorithm.

1 Introduction

Cluster analysis in data mining is a critical business application, which has recently become a highly active topic in data mining research [1]-[7]. Most of existing clustering techniques have high computational time, or may have pattern recognition problems when using large databases. To solve limitations of the previous existing clustering methods, this work presents a new algorithm named “**Grid-based and TREe-Alike Clustering technique for Large databasEs**” (G-TREACLE) by integrating with grid-based, density-based and hierarchical clustering approaches. Performance studies show that the proposed G-TREACLE approach is a highly robust clustering technique.

2 Preliminaries

Several clustering algorithms regarding this work are described as follows.

K-means is the one of popular partitional algorithm [4]. It takes the input parameter, k , and partitions a set of n objects into k clusters. K-means always converges to a local optimum and it can not filter noise.

The grid-based clustering algorithm defines clusters as a multiresolution grid data structure. It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed. The major advantage of the approach is its fast processing time. CLIQUE is one of the most famous grid-based techniques [7]. However, its cluster boundaries are either horizontal or vertical, due to the nature of the rectangular grid.

To identify clusters with arbitrary shape, density-based clustering approaches have been proposed. Those typically regard clusters as dense regions of objects in the data space that are separated by regions of low density (representing noise). DBSCAN is the one of well-know density-based approaches. Although it can accurately recognize any arbitrary pattern and different size clusters, and filters noise [5]. However, the time complexity of DBSCAN is high when the database size is large.

GDH integrates the idea of grid-based, density-based and hierarchical clustering methods, developed by Wang [2]. GDH refers the conception of density function and gradient decrease and concept of sliding window [2]. Although GDH can significantly eliminate the problem of indentation boundaries resulted from traditional grid-based algorithms, it may fail in grouping objects to the right position if two clusters are the same time in the populated hypercube.

3 The Proposed G-TREACLE Clustering Algorithm

This section describes the concepts of the proposed new G-TREACLE clustering algorithm. Ideally, the G-TREACLE algorithm creates a feature space through “hypercubes map constructing” in which all of objects are located on appropriate

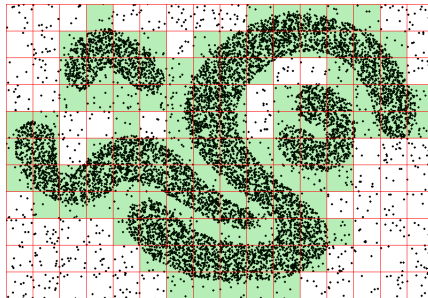


Fig. 1. In the 2-D hypercubes map, the hypercubes with dark colors are termed populated hypercube [6]

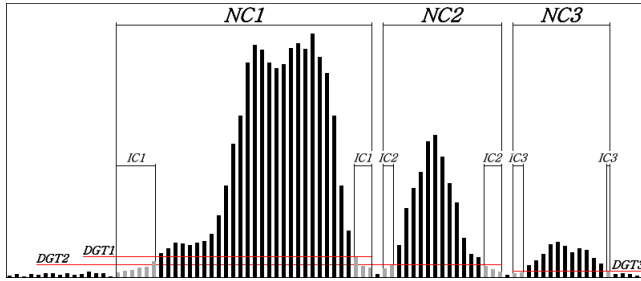


Fig. 2. Sample of solid framework recognizing in 1-D feature space

position. Then, “recognizing solid framework” is employed to fleetly identify the framework of clusters, and subsequently adopt “tree-alike pattern” within “edge shaping” to discover “blurred region”, which may contain noises and cluster objects. Finally, the parts resulted from the above concepts will be integrated to acquire the complete clusters. The implemented details of concepts are illustrated with four parts as follows:

(1) **Hypercubes map constructing:** Reducing the number of searching spaces is the main idea of this step. Initially, G-TREACLE constructs a hypercubes map by splitting the feature space in accordance with a hypercube’s length. Then, each object is assigned to an appropriate hypercube. If the total number of objects in the hypercube is greater than the threshold Hd , this hypercube is named “populated hypercube” [6]. Fig. 1 illustrates the concept. The searching expansion through the initial point will be performed. Notably, a populated hypercube is called “initial point” of search space if it has the highest number of objects among all populated hypercubes.

(2) **Recognizing solid framework:** This investigation adopts the “dynamic-gradient-threshold” as a measure of hypercube-volume, namely the number of objects in the populated hypercube, detecting preprocesses to discover the solid framework of clusters excluding the blurred region. The dynamic-gradient-threshold is obtained as follows:

$$DGT = |HC| \times PSV \tag{1}$$

where $|HC|$ indicates the number of objects in the most populated hypercube HC in the cluster, and PSV is the percentage of the submontane value, which is an input parameter. Fig. 2 depicts an example of the usage of dynamic-gradient-threshold. Every bar in Fig. 2 indicates the number of objects in each populated hypercube. Since every bar within a cluster may be different, dynamic-gradient-threshold can dynamically determine whether a populated hypercube can be treated as the solid framework of clusters in which every object can be assigned to a cluster without calculation. In Fig. 2, $NC1$, $NC2$ and $NC3$ represent the complete cluster. After computing the dynamic-gradient-threshold, such as $DGT1$, $DGT2$ and $DGT3$ in Fig. 2, for each cluster, the solid framework of clusters will be identified and assigned directly to a cluster but excluding

the “blurred region” representing the areas whose number of objects is under dynamic-gradient-threshold, given as $IC1$, $IC2$, $IC3$ and the areas between the clusters. Subsequently, the edge shaping step has to be utilized to detect those “blurred region”, as displayed on populated hypercubes A, B, D, F and G of Fig. 3.

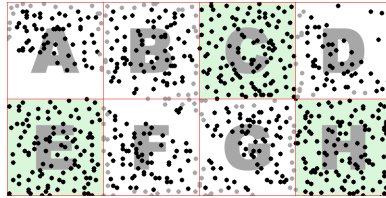


Fig. 3. Illustration of border objects for edge shaping in 2-D hypercubes map

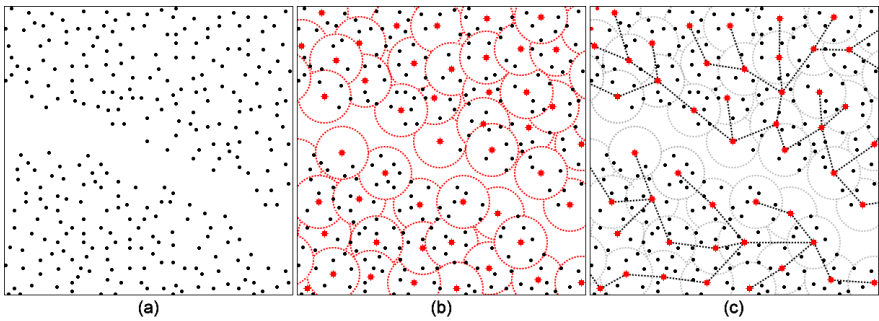


Fig. 4. Concept of searching expansion through the tree-like pattern. (a) The original datasets (b) The neighbor-area set (c) The tree-like pattern.

(3) **Edge Shaping:** The aim of this step is to define accurately the blurred region of a cluster. In this work, the new density-based clustering method is proposed. In contrast to conventional density-based clustering algorithms, e.g., DBSCAN, the proposed density-based method processes searching expansion through a “tree-like pattern” comprising many centroids for each cluster, thus decreasing time complexity. Fig. 4 displays the procedure of how does the proposed density-based method work. In the 2-D hypercubes map, displayed in the diagram (a) of Fig.4, there is a given original data set $D = \{x_1, x_2, \dots, x_m\}$, and a centroid set $C = \{c_1, c_2, \dots, c_n\}$. For an object x_j picked from D , the centroid c_i choosing process is defined as:

$$c_i = \{x_j, \text{if } C = \phi\} \tag{2}$$

or

$$c_i = \{x_j, \text{if } d(x_j, c_p) > w, c_p \in C, p = 1, \dots, i - 1\} \tag{3}$$

where w is the radius of the search circle and the distance function $d(x_j, c_p)$ is the Euclidean distance function:

$$d(x_j, c_p) = \sqrt{\sum_{r=1}^k (x_{jr} - c_{pr})^2} \tag{4}$$

where k represents the dimension. If the centroid set C is empty or the distance between the object x_j and each centroid c_p in C is greater than w , the object x_j is chosen as new centroid. Otherwise, the object x_j is assigned to its closest centroid c_p in C . As displayed in the diagram (b) of Fig. 4, each zone surrounded by dotted circle is termed “neighbor-area” in which the largest point is illustrated as centroid. And the neighbor-area NA_p must satisfy:

$$NA_p \supset \{x_j \in D, c_p \in C : d(x_j, c_p) \leq w\} \tag{5}$$

where c_p is the centroid of NA_p . Subsequently, we need to identify which neighbor-area consisting of noise. In order to achieve this purpose, the density of every neighbor-area NA_p is determined by deriving density function [6] rather than directly counting the number of objects contained in the neighbor-area. The assumption is that the density value of the neighbor-area (namely region) comprising noise is generally lower than that of the populated neighbor-area containing normal clusters objects since its distribution is always sparser than that of the populated neighbor-area [6]. In other words, this means that although the neighbor-areas consisting of noise have the same number equivalent to the ones consisting of normal clusters objects, but the derived density value of former generally lower than that of latter. Consider some neighbor-areas within the clusters displayed in the diagram (b) of Fig. 4 that are not surrounded completely by dotted circle, those areas consist of fewer normal objects but cannot be labeled as noise-area since the density of those areas is greater than the density of noise-areas that not belong to any cluster.

In [6], influence function is defined as a mathematical description that the influence of an object has within its neighborhood, while the density function is defined as the sum of influence function of all objects in the region, and can be any arbitrary function. For simplicity, this work applies the Euclidean density function and Gaussian representation. The Gaussian density function is given by [6]:

$$f_{Gauss}^D(x) = \sum_{i=1}^N e^{-\frac{d(x_i, x_j)^2}{2\sigma^2}}, \tag{6}$$

where N represents the number of objects within the region, $d(x_i, x_j)$ denotes the distance between x_i and x_j , and σ is the standard deviation. If derived density value of the neighbor-area is greater than the threshold $MinDensityVal$, it will be preserved as a “node”. Otherwise, the neighbor-area will be pruned and labeled as noise-area.

After the pruning process, each node searches its neighbor nodes and links them through the virtual edges, which are illustrated in the diagram (c) of Fig. 4.

The connection between the nodes means that their distance is less than twice the w stated above. After neighbor nodes searching recursively, a “tree-like pattern” can be constructed as a cluster mapping. On the other hand, a broken connection between the patterns makes them into different clusters or noises. The complete algorithm is described as follows.

```
TAClustering(PartialDataSets,Width,MinDensityVal)
  NeighborAreaSet = null;
  FOR i FROM 1 TO PartialDataSets.Size DO
    Object = PartialDataSets.get(i);
    IF NeighborAreaSet.Size <> Empty
      FOR j FROM 1 TO NeighborAreaSet.Size DO
        NeighborArea = NeighborAreaSet.get(j);
        IF Object.isCloseToCentroid(NeighborArea,Width) == TRUE
          Object.assignTo(NeighborArea);
          Object.isAssigned = TRUE;
          break;
        END IF
      END FOR
      IF Object.isAssigned == FALSE
        NeighborAreaSet.setCentroid(Object);
      END IF
    ELSE
      NeighborAreaSet.setCentroid(Object);
    END IF-ELSE
  END FOR

  FOR i FROM 1 TO NeighborAreaSet.Size DO
    IF NeighborAreaSet.get(i).DensityValue < MinDensityVal
      NeighborAreaSet.prune(i);
    END IF
  END FOR

  FOR i FROM 1 TO NeighborAreaSet.Size DO
    Centroid = NeighborAreaSet.getCentroid(i);
    searchNeighborNode(Centroid,2*Width,NeighborAreaSet);
  END FOR
END TAClustering
```

`PartialDataSets` represents a partial dataset. `Width` is a search radius, and `MinDensityVal` denotes the minimal density threshold value in the region.

The neighbor node searching process `searchNeighborNode()` is as follows:

```
searchNeighborNode(CCentroid,DWidth,NeighborAreaSet)
  FOR i FROM 1 TO NeighborAreaSet.Size DO
    NCentroid = NeighborAreaSet.getCentroid(i);
    IF NCentroid.PROCESSED == FALSE && NCentroid.isCloseTo(CCentroid,DWidth) == TURE
      NCentroid.linkTo(CCentroid);
      NCentroid.PROCESSED = TURE;
      searchNeighborNode(NCentroid,DWidth,NeighborAreaSet);
    END IF
  END FOR
END searchNeighborNode
```

After running the new density-based clustering method `TAClustering()`, a set of sub-clusters can be gained from the populated hypercube that not belongs to the solid framework of the cluster. These populated hypercubes may contain objects belonging to two different clusters, as mentioned above and depicted on populated hypercubes F and G in Fig. 3. Border objects of sub-cluster and noise can be recognized at the same time [5]. In order to produce the precise combination, the proposed algorithm connects sub-cluster resulted from

TAClustering() run with the solid framework of cluster through the border objects of sub-cluster. Border objects are redefined as objects resulting from a TAClustering() run that are close to the populated hypercube's border. This redefinition shortens the computational time in TAClustering(). The light color objects (on the border) on populated hypercubes A, B, D, F and G of Fig. 3 indicate border objects.

(4) **Consolidation stage:** After the edge shaping stage, the algorithm merges the parts resulted from method TAClustering() with the solid framework of the cluster, depending on which border objects are close to the solid framework of cluster. The proposed algorithm repeats the process to recognize all clusters.

The complete clustering algorithm described as follows:

```
G_TREACLE(DataSets,Cl,PSV,Hd,Width,MinDensityVal)
  Initialization();
  ClusterId = 1;
  constructHCubeMap(Cl);
  PopulHCubeSet = getPopulHCubeSet(DataSets,PSV,Hd);
  WHILE(TRUE) DO
    IPHCube = getInitialPoint(PopulHCubeSet);
    IF IPHCube == NULL
      END ALGORITHM
    END IF
    DGT = IPHCube.ObjcetNumber * PSV;
    changeClusterId(IPHCube,ClusterId);
    searchNeighborHCubes(IPHCube,ClusterId,DGT);
    ClusterId++;
  END WHILE
END G_TREACLE
```

DataSets is an entire database. Cl represents the length of a hypercube, PSV denotes the percentage of the submontane value, and Hd is the threshold of the populated hypercube's volume. Width represents a search radius, and MinDensityVal denotes the minimal density threshold value in the region.

The neighbor searching process searchNeighborHCubes() is as follows:

```
searchNeighborHCubes(HCube,ClusterId,DGT)
  NeighborHCubes = getNeighborHCubes(HCube);
  WHILE NeighborHCubes.Size <> Empty DO
    CurrHCube = getHighestVolumeNeighborHCubes(NeighborHCubes);
    IF CurrHCube.ObjectNumber > DGT
      changeClusterId(CurrHCube,ClusterId);
      searchNeighborHCubes(CurrHCube,ClusterId,DGT);
    ELSE
      NCs = TAClustering(CurrHCube,Width,MinDensityVal);
      FOR i FROM 1 TO NCs.Size DO
        IF NCs.getSubCluster(i).Borders.areNear(HCube) == TRUE
          changeClusterId(NCs.getSubCluster(i),ClusterId);
        END IF
      END FOR
      searchNeighborHCubes(CurrHCube,ClusterId,DGT);
    END IF-ELSE
    NeighborHCubes.deleteNeighborHCube(CurrHCube);
  END WHILE
END searchNeighborHCubes
```

The process is repeated to construct the entire cluster.

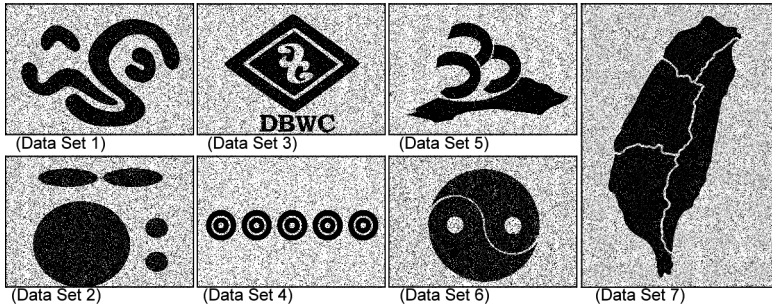


Fig. 5. The original datasets for experiment

4 Performance Studies

In this study, G-TREACLE was implemented in a Java-based program, and run on a desktop computer with 256MB RAM, an Intel 1.5GHz CPU on Microsoft MS Windows XP professional Operational System. For simple visualization, seven synthetic 2-D datasets were utilized to evaluate the performance of the proposed algorithm [3]. Among these datasets, the patterns of dataset 1, 2 and 4 were sampled from [2] and [5], Fig. 5 shows the original datasets. The results of the proposed algorithm were compared with DBSCAN, K-means, CLIQUE and GDH. Four kinds of data sizes in seven synthetic 2-D datasets, with 11,500, 115,000, 230,000 and 575,000 objects in seven synthetic 2-D datasets, and all with 15% noise, were employed in this experiment. For clustering performance comparisons, the clustering correctness rate (CCR) and noise filtering rate (NFR) are introduced. Notably, CCR represents the percentage of cluster objects correctly recognized by algorithm, while NFR denotes the percentage of noise objects correctly filtered by algorithm. Due to the computational time of DBSCAN increases significantly as the number of databases increases, hence Table 1 does not list the simulation results for DBSCAN (N/A means that the simulations were not performed). Table 1 shows the clustering experimental results with G-TREACLE, K-means, DBSCAN, CLIQUE and GDH by utilizing 575,000 object datasets. Owing to the limitation of length, not all experimental results are shown. It is observed that G-TREACLE can handle arbitrary patterns for clustering, while K-means cannot recognize arbitrary shapes. Although CLIQUE and GDH could handle the complex patterns in Dataset 4 to 7, CLIQUE could not smoothly identify clusters' edge due to the nature of the rectangular grid, and then it caused in inaccurate results. Additionally, the gradient decrease function in GDH placed some clusters the wrong position if the populated hypercubes were neighbors but the gradient decrease between the populated hypercubes was too high. In complex datasets such as DataSets 4, 5, 6 and 7, GDH and CLIQUE need to set small capacity of populated hypercube for distinction between cluster's borders that are close to each other. Therefore, the time cost of GDH and CLIQUE raises with increasing numbers of populated

Table 1. Comparisons with G-TREACLE, K-means, DBSCAN, CLIQUE and GDH using 575,000 objects data sets with 15% noise; item 1 represents time cost (in seconds); item 2 denotes the CCR (%), while item 3 is NFR (%).

Algorithm	Item	DataSet-1	DataSet-2	DataSet-3	DataSet-4	DataSet-5	DataSet-6	DataSet-7
K-means	1	18.531	16.391	36.625	59.437	43.203	7.828	19.906
	2	49.925%	51.149%	25.887%	60.837%	57.612%	50.007%	54.49%
	3	0%	0%	0%	0%	0%	0%	0%
DBSCAN	1	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	2	N/A	N/A	N/A	N/A	N/A	N/A	N/A
	3	N/A	N/A	N/A	N/A	N/A	N/A	N/A
CLIQUE	1	5.016	8.031	8.906	12.281	30.094	31.219	46
	2	98.763%	99.104%	98.615%	95.926%	97.274%	95.647%	93.547%
	3	95.92%	98.149%	97.568%	99.305%	99.608%	99.79%	99.805%
GDH	1	8.188	9.516	10.063	13.359	31.75	26.297	51.469
	2	99.213%	99.642%	98.884%	98.299%	98.153%	96.456%	96.4%
	3	96.618%	97.477%	97.387%	98.932%	99.408%	99.736%	99.71%
G-TREACLE	1	6.156	5.594	6.547	7.766	8.469	10.64	15.75
	2	99.392%	99.511%	99.138%	98.376%	99.767%	99.754%	99.127%
	3	98.694%	99.051%	98.998%	98.894%	98.377%	98.74%	98.949%

hypercubes to be searched and processed. As shown in Table 1, G-TREACLE usually yields more accurate results and performs fast than K-means, DBSCAN, CLIQUE and GDH.

5 Conclusion

This work develops a new clustering algorithm named G-TREACLE for data mining. It can accurately identifies large patterns that are close to each other by using tree-alike pattern and is capable of successfully eliminate edge indention, so that it may improve the clustering performance of large databases as well as eliminate outliers. In addition, simulation results demonstrate that the proposed new clustering approach performs better than some existing well-known methods such as the K-means, DBSCAN, CLIQUE and GDH algorithms.

Acknowledgments. The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 96-2221-E-020-027.

References

1. Tsai, C.F., Tsai, C.W., Wu, H.C., Yang, T.: ACODF: A Novel Data Clustering Approach for Data Mining in Large Databases. *Journal of Systems and Software* 73, 133-145 (2004)
2. Wang, T.P., Tsai, C.F.: GDH: An Effective and Efficient Approach to Detect Arbitrary Patterns in Clusters with Noises in Very Large Databases. In: Degree of master at National Pingtung University of Science and Technology, Taiwan (2006)

3. Tsai, C.F., Yen, C.C.: ANGEL: A New Effective and Efficient Hybrid Clustering Technique for Large Databases. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 817–824. Springer, Heidelberg (2007)
4. McQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. In: Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, pp. 281–297 (1967)
5. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, pp. 226–231 (1996)
6. Hinneburg, A., Keim, D.A.: An Efficient Approach to Clustering in Large Multimedia Databases with Noise. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining, pp. 58–65 (1998)
7. Agrawal, R., Gehrke, J., Gunopulos, D., Raghavan, P.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, pp. 94–105. ACM Press, Seattle, Washington (1998)

A Clustering-Oriented Star Coordinate Translation Method for Reliable Clustering Parameterization

Chieh-Yuan Tsai and Chuang-Cheng Chiu

Department of Industrial Engineering and Management, Yuan-Ze University, Taiwan, R.O.C.
cytsai@saturn.yzu.edu.tw

Abstract. When conducting a clustering process, users are generally concerned whether the clustering result is reliable enough to reflect the actual clustering phenomenon. The number of clusters and initial cluster centers are two critical parameters that influence the reliability of clustering results highly. We propose a Clustering-Oriented Star Coordinate Translation (COSCT) method to help users determining the two parameters more confidently. Through COSCT all objects from a multi-dimensional space are adaptively translated to a 2D star-coordinate plane, so that the clustering parameterization can be easily conducted by observing the clustering phenomenon in the plane. To enhance the cluster-displaying quality of the star-coordinate plane, the feature weighting and coordinate arrangement procedures are developed. The effectiveness of the COSCT method is demonstrated using a set of experiments.

Keywords: Clustering, Data visualization, Star coordinate, Gaussian mixture model, Expectation maximization, Particle swarm optimization.

1 Introduction

Clustering aims at grouping objects into clusters so that objects in a cluster are similar to each other and are different from objects in different clusters. Users always raise their question whether the generated clustering results are reliable enough to reflect actual clustering phenomenon. They first ask themselves how many clusters are proper to reveal the actual clustering phenomenon [1]. Generally, the number of clusters is decided based on various cluster validity indexes through a trial-and-error validation process. However, how to adopt an appropriate index could be a dubious and data-dependent problem. Besides, the initial cluster centers are regarded as another critical parameterization factor that influences the reliability of clustering results. Typically, random initial-center generation is the most common method. However, it tends to lead the clustering result converge to a local optimum.

We propose a Clustering-Oriented Star Coordinate Translation (COSCT) method to solve the above two drawbacks. Similar to the Star Coordinate Translation (SCT) method [2], COSCT is further enhanced in many aspects. SCT is a data visualization method translates objects from a multi-dimensional space into points in a 2D star-coordinate plane. By observing the visualizing result displayed in the plane, users can infer the clustering phenomenon existing in the original multi-dimensional space.

Fig. 1 shows the translation result of SCT that maps objects from a nine-dimensional space into a star-coordinate plane. For each recognized cluster, moreover, the points close to its cluster center are also discerned roughly, so that the location of the cluster center in original multi-dimensional space is inferred from the locations of these near points in the original space. However, it is still hard for users to manipulate SCT since adjustments for the rotation angle and length of each coordinate axis could be time-consuming and tedious. Only when the lengths and rotation angles are adjusted well as shown in Fig. 1, the visualizing result in the plane just reveals a meaningful clustering phenomenon.

Therefore, the purpose of COSCT is to make the star-coordinate plane show real clustering phenomenon without human manipulation. That is, the constructed plane will be clustering-oriented. Different to SCT, COSCT involves two new procedures: feature weighting and coordinate arrangement. In the feature weighting procedure, a weight measure that evaluates the importance of each feature to clustering is developed from the Gaussian mixture model of data dissimilarity. The evaluated weight of each feature represents the unit length of its corresponding coordinate axis so as to solve the length adjustment problem. In the coordinate arrangement procedure, the rotation angle arrangement is modeled as an optimization problem systematically, and is solved by the Particle Swarm Optimization (PSO) algorithm [3] efficiently. Finally, all objects are translated into points in the star-coordinate plane constructed by COSCT. Therefore, users can be more confident of deciding the number of clusters and initial cluster centers by observing the plane, so that it makes clustering algorithms yield good clustering results successfully.

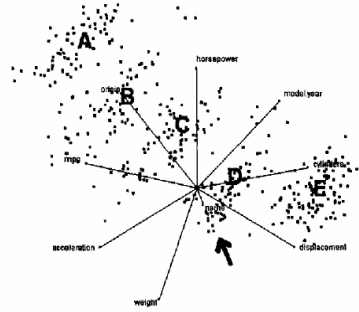


Fig. 1. The translation result of SCT which maps objects from a nine-dimensional space into a star-coordinate plane [2]. Five clusters displayed in the plane can be recognized confidently.

2 The Clustering-Oriented Star Coordinate Translation Method

2.1 Feature Weighting Procedure

A dataset $\mathbf{X}=\{\mathbf{x}_i|i=1,\dots,I\}$ contains I objects and a feature set $\mathbf{F}=\{\mathbf{f}_m|m=1,\dots,M\}$ comprises M features. Each object $\mathbf{x}_i=(x_{i1},\dots,x_{im},\dots,x_{iM})$ is described by the M feature values in \mathbf{F} where x_{im} is the feature value of \mathbf{x}_i in terms of the feature \mathbf{f}_m . For a feature \mathbf{f}_m , its I feature values of all objects must be normalized in advance for unifying the scales of all M features. In this paper, a weight measure that evaluates the importance of each \mathbf{f}_m to clustering is developed from the dissimilarities of all pairs of objects in \mathbf{f}_m . The dissimilarity between \mathbf{x}_i and \mathbf{x}_j in terms of \mathbf{f}_m is formulated as: $\text{Dissimilarity}_m(\mathbf{x}_i,\mathbf{x}_j)=|x_{im}-x_{jm}|$. We can obtain $N=I(I-1)/2$ dissimilarities for each \mathbf{f}_m . Let the n th calculated dissimilarity in \mathbf{f}_m be diss_n^m where $n=1,2,\dots,N$. All N diss_n^m values are considered as the samples drawn from a mixture consisting of two Gaussian distributions \mathbf{G}_1^m and \mathbf{G}_2^m which represent the ‘‘intra-cluster’’ and ‘‘inter-cluster’’ dissimilarity variables respectively. Assume μ_1^m and var_1^m are the mean and

variance associated with G_1^m , and μ_2^m and var_2^m are the mean and variance for G_2^m . The mixture probability density function of $diss_n^m$, $p(diss_n^m)$, is expressed as:

$$p(diss_n^m) = \alpha_1^m \times p(diss_n^m | \mu_1^m, var_1^m) + \alpha_2^m \times p(diss_n^m | \mu_2^m, var_2^m) \tag{1}$$

where α_1^m and α_2^m are the occurrence proportions of G_1^m and G_2^m respectively, and $\alpha_1^m, \alpha_2^m \geq 0; \alpha_1^m + \alpha_2^m = 1$. Given the N $diss_n^m$ values in \mathbf{f}_m , the logarithm transform for the likelihood function of $\Phi \equiv \{\alpha_1^m, \alpha_2^m, \mu_1^m, \mu_2^m, var_1^m, var_2^m\}$ is shown as Eq. (2).

$$\log L(\Phi) = \sum_{n=1}^N \log(\alpha_1^m \times p(diss_n^m | \mu_1^m, var_1^m) + \alpha_2^m \times p(diss_n^m | \mu_2^m, var_2^m)) \tag{2}$$

We use the expectation maximization (EM) algorithm [4] to infer Φ so that $\log L(\Phi)$ can be maximized. For each $diss_n^m$, let p_{n1}^m and p_{n2}^m be the likelihood from the two distributions G_1^m and G_2^m where $p_{n1}^m + p_{n2}^m = 1$. So, Eq. (2) can be expressed as Eq. (3):

$$\log L(\Phi) = \sum_{n=1}^N [p_{n1}^m (\log p(diss_n^m | \mu_1^m, var_1^m) + \log \alpha_1^m) + p_{n2}^m (\log p(diss_n^m | \mu_2^m, var_2^m) + \log \alpha_2^m)] \tag{3}$$

To infer the optimal Φ , the expectation (E) step and maximization (M) step are alternately performed in EM. Let $\Phi(t) = \{\alpha_1^m(t), \alpha_2^m(t), \mu_1^m(t), \mu_2^m(t), var_1^m(t), var_2^m(t)\}$ be the estimate of Φ at the iteration t . Initially, i.e. $t=0$, $\Phi(0)$ is generated randomly. At the iteration t , the E step computes the expectations of likelihood $p_{n1}^m(t)$ and $p_{n2}^m(t)$, $n=1, 2, \dots, N$, by including the current $\Phi(t)$ into Eq. (3), which are shown as Eq. (4) and Eq. (5) respectively. Then, the M step infers $\Phi(t+1)$ which will be used at the next iteration ($t+1$) by including $p_{n1}^m(t)$ and $p_{n2}^m(t)$ at the current iteration t into Eq. (3). The Eq. (6), Eq. (7), and Eq. (8) are used to calculate the six parameters in $\Phi(t+1)$. The E and M steps are alternately repeated until $|\log L(\Phi(t+1)) - \log L(\Phi(t))| \leq \epsilon$ where ϵ is the stop criteria. When EM is stopped at the iteration t , the estimate $\Phi(t)$ serves as the optimal Φ .

$$p_{n1}^m(t) = \frac{\alpha_1^m(t) \times (1/var_1^m(t)) \times \exp(-[diss_n^m - \mu_1^m(t)]/var_1^m(t))}{\sum_{a=1}^2 [\alpha_a^m(t) \times (1/var_a^m(t)) \times \exp(-[diss_n^m - \mu_a^m(t)]/var_a^m(t))]} \text{ for } n = 1, \dots, N \tag{4}$$

$$p_{n2}^m(t) = 1 - p_{n1}^m(t) \text{ for } n = 1, \dots, N \tag{5}$$

$$\begin{cases} \mu_1^m(t+1) = \left(\sum_{n=1}^N p_{n1}^m(t) \times diss_n^m \right) / \sum_{n=1}^N p_{n1}^m(t) \\ \mu_2^m(t+1) = \left(\sum_{n=1}^N p_{n2}^m(t) \times diss_n^m \right) / \sum_{n=1}^N p_{n2}^m(t) \end{cases} \tag{6}$$

$$\begin{cases} var_1^m(t+1) = \left(\sum_{n=1}^N p_{n1}^m(t) \times (diss_n^m - \mu_1^m(t+1))^2 \right) / \sum_{n=1}^N p_{n1}^m(t) \\ var_2^m(t+1) = \left(\sum_{n=1}^N p_{n2}^m(t) \times (diss_n^m - \mu_2^m(t+1))^2 \right) / \sum_{n=1}^N p_{n2}^m(t) \end{cases} \tag{7}$$

$$\begin{cases} \alpha_1^m(t+1) = \frac{1}{N} \times \sum_{n=1}^N p_{n1}^m(t) \\ \alpha_2^m(t+1) = 1 - \alpha_1^m(t+1) \end{cases} \tag{8}$$

Each of the $N \text{diss}_n^m$ values in \mathbf{f}_m can be decomposed as two components. One is the “intra-cluster” dissimilarity with a relative small value, which means the two objects could be in the same cluster. Another is the “inter-cluster” dissimilarity with a relative large value, which means the two objects should be in different clusters. Further, the mean of the N “intra-cluster” dissimilarities in \mathbf{f}_m , μ_1^m , and the mean of the N “inter-cluster” dissimilarities in \mathbf{f}_m , μ_2^m , are both known from the optimal Φ after performing EM. When μ_1^m is smaller and μ_2^m is larger simultaneously, the importance of \mathbf{f}_m to clustering quality should be higher. Hence, we can define a weight measure used to evaluate the importance of \mathbf{f}_m to clustering as Eq. (9). After this procedure, the unit length of each coordinate axis can be represented by its corresponding feature weight, so that the length adjustment problem occurred in SCT can be solved.

$$fw(\mathbf{f}_m) = \mu_2^m - \mu_1^m \quad \text{for } m = 1, \dots, M \tag{9}$$

2.2 Coordinate Arrangement Procedure

The procedure aims at finding the optimal arrangement for the rotation angles of all coordinate axes in the star-coordinate plane. The influence of the included angle of two axes on the cluster-displaying quality is first analyzed systematically. Then, it is modeled as the coordinate arrangement optimization problem and solved by PSO.

2.2.1 Influence of the Included Angle of Two Coordinate Axes on the Cluster Displaying Quality

Let $\mathbf{SC} = \{SC_m | m = 1, \dots, M\}$ be the set of M coordinate axes associated with the M features in \mathbf{F} where SC_m is the coordinate axis associated with \mathbf{f}_m . In addition, $\Theta = \{\theta_m | m = 1, \dots, M\}$ be the set of the rotation angles of M coordinate axes in \mathbf{SC} where $\theta_m, 0^\circ \leq \theta_m < 360^\circ$, is the rotation angle of SC_m turning from the positive x-direction in the Cartesian coordinate system. Therefore, the included angle of SC_m and SC_n can be formulated as: $Ang(\theta_m, \theta_n) = |\theta_m - \theta_n|$. Let the cosine value for $Ang(\theta_m, \theta_n)$ be termed as $\cos(Ang(\theta_m, \theta_n))$. When $0^\circ \leq Ang(\theta_m, \theta_n) < 90^\circ$ or $270^\circ < Ang(\theta_m, \theta_n) < 360^\circ$, $Ang(\theta_m, \theta_n)$ is acute so that $0 < \cos(Ang(\theta_m, \theta_n)) < 1$. On the other hand, when $90^\circ < Ang(\theta_m, \theta_n) < 270^\circ$, $Ang(\theta_m, \theta_n)$ is obtuse so that $-1 < \cos(Ang(\theta_m, \theta_n)) < 0$. If $Ang(\theta_m, \theta_n) = 90^\circ$ or 270° , $\cos(Ang(\theta_m, \theta_n)) = 0$. The correlation coefficient between \mathbf{f}_m and \mathbf{f}_n , $\rho(\mathbf{f}_m, \mathbf{f}_n)$, can describe the distribution of objects in terms of \mathbf{f}_m and \mathbf{f}_n , which is defined as Eq. (10):

$$\rho(\mathbf{f}_m, \mathbf{f}_n) = \frac{\sum_{i=1}^I (x_{im} - \bar{x}_m)(x_{in} - \bar{x}_n)}{\sqrt{\sum_{i=1}^I (x_{im} - \bar{x}_m)^2} \times \sqrt{\sum_{i=1}^I (x_{in} - \bar{x}_n)^2}} \tag{10}$$

where \bar{x}_m and \bar{x}_n are the means of the all feature vaules of the I objects in terms of \mathbf{f}_m and \mathbf{f}_n respectively, and $-1 \leq \rho(\mathbf{f}_m, \mathbf{f}_n) \leq 1$. If $0 < \rho(\mathbf{f}_m, \mathbf{f}_n) \leq 1$, \mathbf{f}_m and \mathbf{f}_n are correlated positively; If $-1 \leq \rho(\mathbf{f}_m, \mathbf{f}_n) < 0$, \mathbf{f}_m and \mathbf{f}_n are correlated negatively. If $\rho(\mathbf{f}_m, \mathbf{f}_n) = 0$, \mathbf{f}_m and \mathbf{f}_n are independent. By adjusting the included angle $Ang(\theta_m, \theta_n)$ between SC_m and SC_n , we observe the following situations that can enhance the cluster-displaying quality:

- If $0 < \rho(\mathbf{f}_m, \mathbf{f}_n) \leq 1$, shown as Fig. 2(a), and we tighten $Ang(\theta_m, \theta_n)$ as an acute angle, shown as 45° in Fig. 2(b), the two clusters are separated more distinctly.
- If $-1 \leq \rho(\mathbf{f}_m, \mathbf{f}_n) < 0$, shown as Fig. 3(a), and we loosen $Ang(\theta_m, \theta_n)$ as an obtuse angle, shown as 135° in Fig. 3(c), the two clusters are separated more distinctly.

- If $\rho(\mathbf{f}_m, \mathbf{f}_n) = 0$, shown as Fig. 4(a), no matter $Ang(\theta_m, \theta_n)$ becomes an acute or obtuse angle, shown as Fig. 4(b) and Fig. 4(c), it is unable to enhance the cluster-displaying quality.

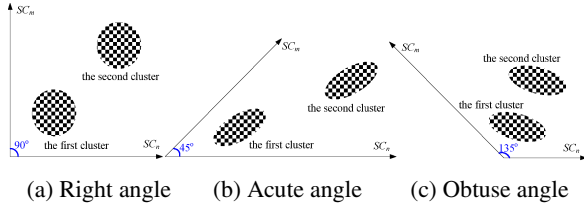


Fig. 2. If $0 < \rho(\mathbf{f}_m, \mathbf{f}_n) \leq 1$, let $Ang(\theta_m, \theta_n)$ be acute to enhance the cluster-displaying quality

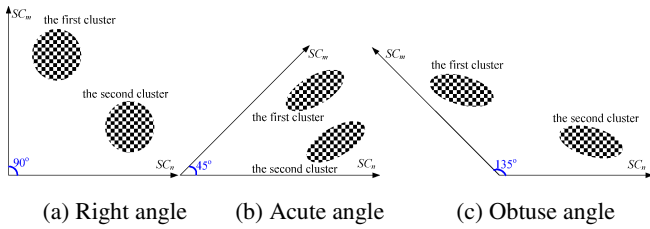


Fig. 3. If $-1 \leq \rho(\mathbf{f}_m, \mathbf{f}_n) < 0$, let $Ang(\theta_m, \theta_n)$ be obtuse to enhance the cluster-displaying quality

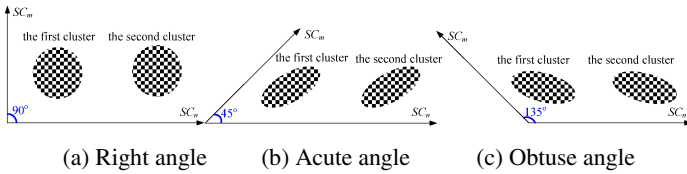


Fig. 4. If $\rho(\mathbf{f}_m, \mathbf{f}_n) = 0$, no matter $Ang(\theta_m, \theta_n)$ is acute or obtuse does not enhance the cluster-displaying quality

According to the above analysis, the adjustment for the rotation angles between SC_m and SC_n for optimizing the cluster-displaying quality of the visualizing result can be modeled as an optimization problem:

$$\text{Minimize } [\rho(\mathbf{f}_m, \mathbf{f}_n) - \cos(Ang(\theta_m, \theta_n))]^2 \tag{11}$$

subject to $0^\circ \leq \theta_m, \theta_n < 360^\circ$. Similarly, we extend it as an optimization problem in which the rotation angles of all M coordinate axes in SC are taken in account simultaneously. In addition, the M corresponding feature weights are also considered. That is, the coordinate arrangement of all coordinate axes for optimizing the cluster-displaying quality of the visualizing result can be modeled as:

$$\text{Minimize } \sum_{m=1}^{M-1} \sum_{n=m+1}^M fw(\mathbf{f}_m) \times fw(\mathbf{f}_n) \times [\rho(\mathbf{f}_m, \mathbf{f}_n) - \cos(\text{Ang}(\theta_m, \theta_n))]^2 \quad (12)$$

subject to $0^\circ \leq \theta_m < 360^\circ$ for $m=1,2,\dots,M$. The coordinate arrangement procedure aims at finding out the optimal angle set $\theta = \{\theta_1, \dots, \theta_m, \dots, \theta_M\}$ that minimizes Eq. (12).

2.2.2 Using the PSO Algorithm for Optimizing the Coordinate Arrangement

In the coordinate arrangement procedure, we adopt the particle swarm optimization (PSO) algorithm [3] to find the optimal angle set $\theta = \{\theta_1, \dots, \theta_m, \dots, \theta_M\}$ for Eq. (12). In PSO, all particles in a swarm are evolved by cooperation and competition among themselves through generations. Let a swarm $\mathbf{G} = \{\mathbf{p}_r, r=1, \dots, R\}$ consist of R particles. Each particle $\mathbf{p}_r = (\theta_{r1}, \dots, \theta_{rm}, \dots, \theta_{rM})$ represents a feasible solution for θ where θ_{rm} is the rotation angle of the m th coordinate axis found in the r th particle. Initially, the value of θ_{rm} in \mathbf{p}_r is randomly generated such that $0^\circ \leq \theta_{rm} < 360^\circ$ for $m=1,2,\dots,M$. Moreover, the current flying velocity of \mathbf{p}_r is represented as $\mathbf{v}_r = (v_{r1}, \dots, v_{rm}, \dots, v_{rM})$ in which the initial value of each velocity v_{rm} is randomly generated such that $-10^\circ \leq v_{rm} \leq 10^\circ$ for $m=1,2,\dots,M$. The fitness of \mathbf{p}_r , termed as $\text{fitness}(\mathbf{p}_r)$, is defined as:

$$\begin{aligned} \text{fitness}(\mathbf{p}_r) &= \text{fitness}((\theta_{r1}, \dots, \theta_{rm}, \dots, \theta_{rM})) \\ &= 1 / \sum_{m=1}^{M-1} \sum_{n=m+1}^M fw(\mathbf{f}_m) \times fw(\mathbf{f}_n) \times [\rho(\mathbf{f}_m, \mathbf{f}_n) - \cos(\text{Ang}(\theta_{rm}, \theta_{rn}))]^2 \end{aligned} \quad (13)$$

During the overall iterations, each \mathbf{p}_r maintains its optimal M rotation angles, $\mathbf{p}_r^{\text{best}} = (\theta_{r1}^{\text{best}}, \dots, \theta_{rm}^{\text{best}}, \dots, \theta_{rM}^{\text{best}})$, that generate the individual maximum fitness when $\mathbf{p}_r^{\text{best}}$ is taken into Eq. (13). Similarly, the swarm \mathbf{G} also maintains the optimal M angles, termed as $\mathbf{g}^{\text{best}} = (\theta_1^{\text{best}}, \dots, \theta_m^{\text{best}}, \dots, \theta_M^{\text{best}})$, found by all particles during the overall iterations. That is, we can obtain the global maximum fitness among all particles when taking \mathbf{g}^{best} into Eq. (13). At each iteration, each v_{rm} in \mathbf{v}_r of \mathbf{p}_r must be updated using Eq. (14). The ratio for the self-cognition and social interaction parts, c_1 and c_2 in Eq. (14), should be one [3]. Therefore, we set $c_1=c_2=2$ in this study. In addition, each new v_{rm}^{new} obtained by Eq. (14) must obey the restriction of Eq. (15) to prevent violent movement for particles. Similarly, each θ_{rm} in \mathbf{p}_r must be updated using Eq. (16) at each iteration, and each new θ_{rm}^{new} must obey the restriction of Eq. (17) to prevent θ_{rm}^{new} from being infeasible. At each iteration, after \mathbf{p}_r has changed its position using Eq. (16), if its fitness is larger than the fitness of $\mathbf{p}_r^{\text{best}}$, $\mathbf{p}_r^{\text{best}}$ will be replaced by \mathbf{p}_r . Further, if its fitness is also larger than the fitness of \mathbf{g}^{best} , \mathbf{g}^{best} will be also replaced by \mathbf{p}_r . PSO will stop when reaching the maximum number of iterations N_{iter} . At the moment, $\mathbf{g}^{\text{best}} = (\theta_1^{\text{best}}, \dots, \theta_m^{\text{best}}, \dots, \theta_M^{\text{best}})$ is considered as the optimal rotation angles of the M coordinate axes. The pseudo-code of PSO is shown as Fig. 5.

$$v_{rm}^{\text{new}} = v_{rm} + c_1 \times \text{rand}_1 \times (\theta_{rm}^{\text{best}} - \theta_{rm}) + c_2 \times \text{rand}_2 \times (\theta_m^{\text{best}} - \theta_{rm}) \quad \text{for } m=1, \dots, M \quad (14)$$

$$v_{rm}^{\text{new}} = \begin{cases} 10, & \text{if } v_{rm}^{\text{new}} > 10 \\ -10, & \text{if } v_{rm}^{\text{new}} < -10 \\ v_{rm}^{\text{new}}, & \text{otherwise} \end{cases} \quad \text{for } m=1, \dots, M \quad (15)$$

$$\theta_{rm}^{\text{new}} = \theta_{rm} + v_{rm}^{\text{new}} \quad \text{for } m=1, \dots, M \quad (16)$$

$$\theta_m^{new} = \begin{cases} \theta_m^{new} - 360^\circ, & \text{if } \theta_m^{new} \geq 360^\circ \\ \theta_m^{new} + 360^\circ, & \text{if } \theta_m^{new} < 0^\circ \\ \theta_m^{new}, & \text{otherwise} \end{cases} \quad \text{for } m = 1, \dots, M \quad (17)$$

2.3 Star Coordinate Translation Procedure

After the feature weighting and coordinate arrangement procedures, the star coordinate translation procedure is activated to translate all objects from a multi-dimensional space into 2D points in the star-coordinate plane. When considering the feature weights (i.e. the unit lengths of coordinate axes), the point position of a object x_i in the star-coordinate plane, $\langle X(x_i), Y(x_i) \rangle$, is defined as Eq. (18) and Eq. (19).

```

Input a feature set F contains the  $M$  features; the stop criterion  $N_{iter}$ 
Output a set  $\theta^* = \{\theta_1^*, \dots, \theta_m^*, \dots, \theta_M^*\}$  consists of  $M$  optimal rotation angles for the
       $M$  coordinate axes associated with the  $M$  features in F
(1) For each pair of two features in F { // the number of pairs is  $M(M-1)/2$ 
(2) Calculate their correlation coefficient using Eq. (10) }
(3) Generate a swarm  $G = \{p_1, \dots, p_r, \dots, p_R\}$  consists of  $R$  particles
(4) For each particle  $p_r = (\theta_{r1}, \dots, \theta_{rm}, \dots, \theta_{rM})$  in G {
(5) Randomize the initial value of each rotation angle  $\theta_{rm}$  so that  $0^\circ \leq \theta_{rm} < 360^\circ$ 
(6) Randomize the initial value of each velocity element  $v_{rm}$  in  $\mathbf{v}_r$  so that  $-10^\circ \leq v_{rm} \leq 10^\circ$ 
(7) Calculate the fitness  $fitness(p_r)$  using Eq. (12)
(8) Consider  $p_r$  as  $p_r^{best}$  }
(9) Select the particle with maximum fitness from the swarm G, and consider it as  $\mathbf{g}^{best}$ 
(10) While  $i$  is not equal to  $N_{iter}$  { // the initial value of  $i$  is 0
(11) For each  $p_r$  in G {
(12) Update  $\mathbf{v}_r$  using Eq. (14) and Eq. (15)
(12) Update  $p_r$  using Eq. (16) and Eq. (17)
(14) Calculate  $fitness(p_r)$  using Eq. (12)
(15) If  $fitness(p_r) > fitness(p_r^{best})$  { Replace  $p_r^{best}$  by  $p_r$  }
(16) If  $fitness(p_r) > fitness(\mathbf{g}^{best})$  { Replace  $\mathbf{g}^{best}$  by  $p_r$  }
(17)  $i = i + 1$  }
(18) Return  $\mathbf{g}^{best} = (\theta_1^{best}, \dots, \theta_m^{best}, \dots, \theta_M^{best})$  as  $\theta^* = \{\theta_1^*, \dots, \theta_m^*, \dots, \theta_M^*\}$ 
    
```

Fig. 5. The pseudo-code of PSO for optimizing the coordinate arrangement

$$X(x_i) = \sum_{m=1}^M fw(\mathbf{f}_m) \times x_{im} \times \cos(\theta_m) \quad \text{for } i = 1, \dots, I \quad (18)$$

$$Y(x_i) = \sum_{m=1}^M fw(\mathbf{f}_m) \times x_{im} \times \sin(\theta_m) \quad \text{for } i = 1, \dots, I \quad (19)$$

where $fw(\mathbf{f}_m)$ is the feature weight of \mathbf{f}_m in **F**, and θ_m is the rotation angle of SC_m associated with \mathbf{f}_m . Fig. 6 shows an example of translating an object from a five-dimensional space into a point in the plane. Through COSCT, all I objects are translated as 2D points, which makes users observe clearly whether any clustering phenomenon of these points displays in the star-coordinate plane.

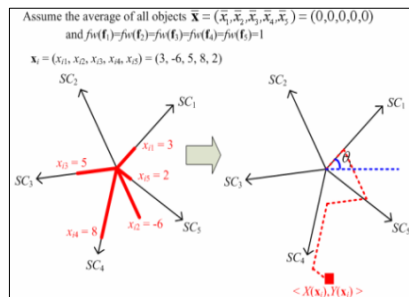


Fig. 6. Translating a 5-dimensional object as a point in the star-coordinate plane

3 Demonstrations

3.1 Determining the Cluster Numbers

The Breast cancer, Iris, and Wine datasets got from [5] are used to demonstrate whether determining the number of clusters and initial cluster centers through COSCT are reliable. Principle Component Analysis (PCA) [6] and Multi-dimensional Scaling (MDS) [7], two common data dimension reduction methods, serve as the comparisons with COSCT. With PCA, the first two components are extracted to construct a 2D plane. Similarly, only two dimensions are retrieved from MDS to form a 2D plane. The number of particles R is set as 100 and the maximum number of iterations N_{iter} is set as 200 in COSCT for all experiments.

Fig. 7 shows the cluster-displaying results using the three methods for the Breast cancer dataset. In each result, obviously, a small and dense cluster is easily observed while a large cluster grouped from other distributed points. No matter which methods are used, the two clusters can be clearly identified. Then, the cluster-displaying results using the three methods for the Iris dataset are shown in Fig. 8. With PCA and MDS, the identified number of clusters is two by observing Fig. 8(a) and Fig. 8(b). Although the suggestion is reasonable, no additional information can reveal the larger cluster could be further partitioned into two smaller clusters. On the contrary, the result outputted by COSTC reveals the number of clusters could be set as 2 or 3, depending on whether the larger cluster is partitioned into two smaller clusters. In this case, COSCT can provide more information for determining the number of clusters, which should cause more reliable clustering results. Finally, the cluster-displaying results

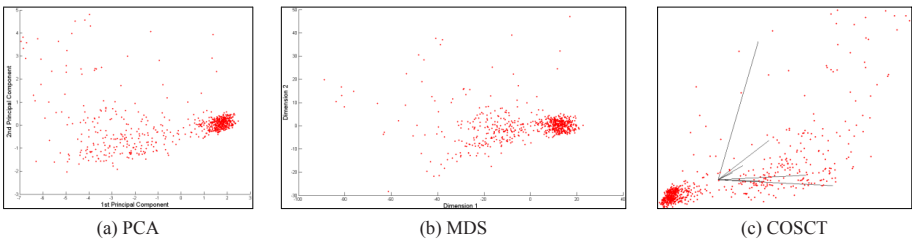


Fig. 7. The cluster-displaying results using the three methods for the Breast cancer dataset

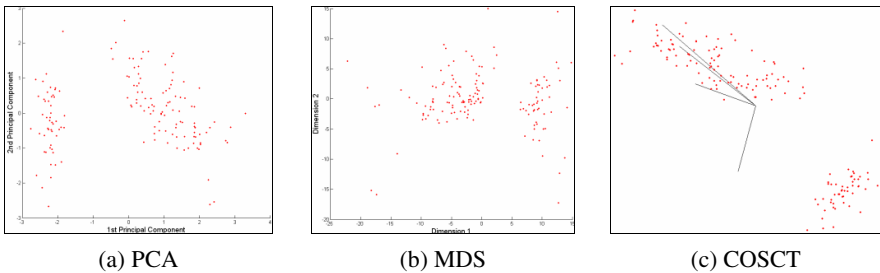


Fig. 8. The cluster-displaying results using the three methods for the Iris dataset

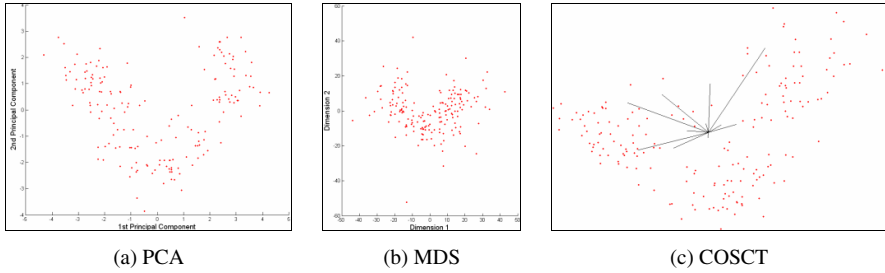


Fig. 9. The cluster-displaying results using the three methods for the Wine dataset

using the three methods for the Wine dataset are shown in Fig. 9. The three clusters can be recognized most clearly using COSCT, shown in Fig. 9(c), since the discrimination among clusters are the most obvious. It means determining the number of clusters through the proposed COSCT method is reliable.

3.2 Determining the Initial Cluster Centers

After recognizing all clusters in the star-coordinate plane, the possible location of each cluster center in the plane can be roughly identified using the sense of sight. For each cluster, we draw a circle around the middle of the cluster to include a number of points. That is, the points within the circle can be considered as neighbors with the cluster center. Fig. 10 shows the neighbor identification result for the three identified cluster centers in Fig. 9(c) using three blue circles. Hence, the location of a cluster center in the original multi-dimensional space is obtained by averaging the locations of these within-circle objects in the original multi-dimensional space.

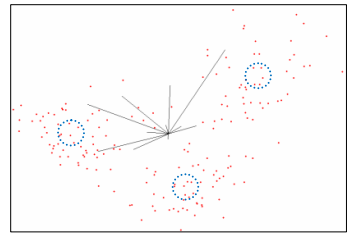


Fig. 10. Identifying the neighbors of the cluster centers for the three identified clusters in Fig. 9(c) using three blue circles

4 Conclusions

The COSCT method is presented to determine the number of clusters and initial cluster centers, which is more reliable than the ones decided through the trial-and-error validation process. No clustering algorithm is employed in the parameterization process, so that it can be a common preprocess for all clustering algorithms. The effectiveness of COSCT is superior to PCA and MDS through our demonstrations.

The reliability of knowledge discovery can be affected by data oriented, knowledge oriented, and algorithm oriented factors [8]. By observing the cluster-displaying result in the star-coordinate plane, two data oriented factors for clustering reliability, including the number of clusters and initial cluster centers, is conducted in this paper. In the future, we will study how to retrieve more useful information from the

cluster-displaying result in the star-coordinate plane for evaluating objectively whether a clustering algorithm is appropriate. It will be useful for users to manage the algorithm oriented factors for achieving more reliable clustering results.

References

1. Pal, N.R., Bezdek, J.C.: On Cluster Validity for Fuzzy C-Means Model. *IEEE Transactions on Fuzzy Systems* 3, 370–379 (1995)
2. Kandogan, E.: Visualizing Multi-dimensional Clusters, Trends, and Outliers Using Star Coordinates. In: *The 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 107–116 (2001)
3. Kennedy, J., Eberhart, R.C.: Particle Swarm Optimization. In: *The 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948 (1995)
4. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society Series B* 39, 1–38 (1977)
5. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.J.: *UCI Repository of Machine Learning Databases* (1998), <http://www.ics.uci.edu/~mllearn/MLSummary.html>
6. Jolliffe, I.T.: *Principal Component Analysis*. Springer, New York (1986)
7. Borg, I., Groenen, P.: *Modern Multidimensional Scaling: Theory and Applications*. Springer, New York (1997)
8. Dai, H.: A Study on Reliability in Graph Discovery. In: *The sixth IEEE International Conference on Data Mining Workshops*, pp. 775–779 (2006)

Constrained Clustering for Gene Expression Data Mining

Vincent S. Tseng, Lien-Chin Chen, and Ching-Pin Kao

Dept. of Computer Science and Information Engineering,
National Cheng Kung University, Taiwan, R.O.C.
{vtseng, lcchen, zeno}@idb.csie.ncku.edu.tw

Abstract. Constrained clustering algorithms have the advantage that domain-dependent constraints can be incorporated in clustering so as to achieve better clustering results. However, the existing constrained clustering algorithms are mostly k -means like methods, which may only deal with distance-based similarity measures. In this paper, we propose a constrained hierarchical clustering method, called *Correlational-Constrained Complete Link (C-CCL)*, for gene expression analysis with the consideration of gene-pair constraints, while using correlation coefficients as the similarity measure. C-CCL was evaluated for the performance with the correlational version of COP- k -Means (C-CKM) method on a real yeast dataset. We evaluate both clustering methods with two validation measures and the results show that C-CCL outperforms C-CKM substantially in clustering quality.

Keywords: Hierarchical Clustering, Constrained Clustering, Gene Expression Mining, Micorarray analysis.

1 Introduction

Clustering analysis has been a widely-used tool for in-silico analysis of microarray or gene expression data [9]. In real applications, however, some constraints on which genes can or cannot reside in the same cluster are often known from background knowledge. COP- k -Means [11] is a constrained version of k -Means, while COP-COBWEB [10] is a constrained version of COBWEB [6]. Constrained- k -Means [1] is a constrained version of Seeded- k -Means, which is a semi-supervised clustering algorithm. Constrained Complete-Link (CCL) [8] utilizes triangle inequality to adjust the proximity matrix according to the pair-wise constraints and then supplies the adjusted matrix to Complete-Link (CL) hierarchical agglomerative clustering. Davidson and Ravi [5] presented a constrained version of k -Means that attempts to minimize the proposed constrained vector quantization error. Although some constrained clustering algorithms have been proposed, they are mostly k -means-like methods. Consequently, they can process data with distance-based similarity measures and can not process data with non-distance-based similarity measures like correlation coefficients, which are popularly used for gene expression analysis.

In this paper, we propose a new constrained clustering method, namely *Correlational-Constrained Complete Link (C-CCL)*, to improve the quality of hierarchical clustering for microarray data analysis with consideration of pair-wise

constraints among genes, while using correlation coefficients as similarity measure. The main idea of C-CCL is based on the Constrained Complete-Link (CCL) algorithm proposed by Klein *et al.* [8]. The proposed method, C-CCL, was evaluated for the performance with the correlational version of CKM (C-CKM) on a real yeast microarray dataset. The evaluations show that C-CCL outperforms C-CKM substantially in terms of clustering quality.

The rest of the paper is organized as follows: In section 2, we demonstrate our proposed method for constrained clustering in detail. Experimental evaluations of the proposed methods are illustrated in section 3. Finally, conclusions and future work are stated in section 4.

2 The Proposed Method

2.1 Imitative Triangle Inequality with Respect to Correlation Coefficient

Given a normalized gene expression matrix, $E = [e_{g,k}]_{m \times n}$, over a set of m genes and n microarray experiments. Each element $e_{g,k}$ means the expression value of gene g at microarray experiment k . Let X and Y denote two gene expression vectors, which are represented by $\langle e_{x,1}, e_{x,2}, \dots, e_{x,n} \rangle$ and $\langle e_{y,1}, e_{y,2}, \dots, e_{y,n} \rangle$, respectively. The Euclidean distance between vectors X and Y is defined as follows:

$$d_{XY} = \sqrt{\sum_{k=1}^n (X_k - Y_k)^2} \quad (1)$$

The Pearson's correlation coefficient between vectors X and Y is defined as follows:

$$r_{XY} = \frac{\sum_{k=1}^n (X_k - \bar{X})(Y_k - \bar{Y})}{\sqrt{\sum_{k=1}^n (X_k - \bar{X})^2} \sqrt{\sum_{k=1}^n (Y_k - \bar{Y})^2}}, \quad (2)$$

where \bar{X} and \bar{Y} denote the sample means of the entries of vectors X and Y , respectively. Pearson's correlation coefficient only measures the degree of linear relationship between two vectors. The correlation takes values ranging from -1 (perfect negative correlation) to 1 (perfect positive correlation). The value 0 of correlation means that there is no linear relationship between two genes.

The uncentered correlation coefficient is similar to the Pearson's correlation coefficient, but is evaluated without centering:

$$r_{XY} = \frac{\sum_{k=1}^n X_k Y_k}{\sqrt{\sum_{k=1}^n X_k^2} \sqrt{\sum_{k=1}^n Y_k^2}} \quad (3)$$

A summary of other types of distance and similarity measure can be found in [7].

The Euclidean distance measure satisfies the triangle inequality:

$$d_{XY} \leq d_{XW} + d_{WY} \quad (4)$$

It means that the distance between two points is the shortest distance along any path. Klein *et al.* utilized this property to adjust the proximity matrix according to the pairwise constraints [8]. However, none of the correlation-based similarity functions satisfy the triangle inequality and hence are known as semi-metric. Therefore, we shall define an imitative triangle inequality with respect to the Pearson’s correlation and the uncentered correlation coefficients in the following.

$$\text{Let } X_w^* = \frac{X_w - \bar{X}}{\sqrt{\sum_{k=1}^n (X_k - \bar{X})^2}} \text{ and } Y_w^* = \frac{Y_w - \bar{Y}}{\sqrt{\sum_{k=1}^n (Y_k - \bar{Y})^2}}, \text{ where } 1 \leq w \leq n. \text{ It}$$

means that $\bar{X}^* = \bar{Y}^* = 0$ and $\sum_{w=1}^n (X_w^*)^2 = \sum_{w=1}^n (Y_w^*)^2 = 1$. Thus, the Pearson’s correlation coefficient (Equation 2) can be rewritten as follows:

$$r_{XY} = r_{X^*Y^*} = \sum_{k=1}^n X_k^* Y_k^* \tag{5}$$

$$\text{Similarly, let } X_w = \frac{X_w}{\sqrt{\sum_{k=1}^n (X_k)^2}} \text{ and } Y_w = \frac{Y_w}{\sqrt{\sum_{k=1}^n (Y_k)^2}}, \text{ where } 1 \leq w \leq n. \text{ Thus,}$$

the uncentered correlation coefficient (Equation 3) can be rewritten as Equation 5. By substituting (5) into (1), we have the form of square of Euclidean distance as follows:

$$\begin{aligned} d_{X^*Y^*}^2 &= \sum_{k=1}^n (X_k^* - Y_k^*)^2 \\ &= \sum_{k=1}^n (X_k^{*2} - 2X_k^*Y_k^* + Y_k^{*2}) \\ &= \sum_{k=1}^n X_k^{*2} + \sum_{k=1}^n Y_k^{*2} - 2\sum_{k=1}^n X_k^*Y_k^* \\ &= 2 - 2\sum_{k=1}^n X_k^*Y_k^* \\ &= 2(1 - r_{X^*Y^*}) \end{aligned} \tag{6}$$

By substituting (4) and (6) into (2), we have the form of Pearson’s correlation and the uncentered correlation coefficients as follows:

$$\begin{aligned} r_{XY} = r_{X^*Y^*} &= 1 - \frac{d_{X^*Y^*}^2}{2} \\ &\geq 1 - \frac{(d_{X^*W^*} + d_{W^*Y^*})^2}{2} \\ &= 1 - \frac{(\sqrt{2(1 - r_{X^*W^*})} + \sqrt{2(1 - r_{W^*Y^*})})^2}{2} \\ &= 1 - (\sqrt{1 - r_{X^*W^*}} + \sqrt{1 - r_{W^*Y^*}})^2 \\ &= 1 - (\sqrt{1 - r_{XW}} + \sqrt{1 - r_{WY}})^2 \end{aligned} \tag{7}$$

Therefore, the imitative triangle inequality with respect to the Pearson’s correlation and the uncentered correlation coefficients is defined as follows:

$$r_{XY} \geq 1 - (\sqrt{1 - r_{XW}} + \sqrt{1 - r_{WY}})^2 \tag{8}$$

2.2 Algorithm of Correlational-Constrained Complete-Link (C-CCL)

The main idea of the proposed C-CCL method is based on Constrained Complete-Link (CCL) algorithm [8], which utilizes triangle inequality with respect to Euclidean distance to adjust the proximity matrix according to the pair-wise constraints. We extend the principle of CCL to meet the requirement for using the Pearson's correlation or the uncentered correlation coefficients as similarity measure. First, we adjust the similarity matrix on the basis of the constraints and their implications. Second, we supply this adjusted matrix to a hierarchical clustering algorithm named Complete-Link (CL) hierarchical agglomerative clustering for obtaining the final clustering results.

There are two steps in the adjusting process. The first step is to *impose* the constraints and the second step is to *propagate* the constraints. In the first step, we specify genes known in the same class as very similar, while two genes in different classes should be very dissimilar. In the second step, we further distort other entries in the similarity matrix to reflect the following two arguments:

1. If genes X and Y are very similar, then genes that are very similar to gene X are similar to gene Y .
2. If genes X and Y are very dissimilar, then genes that are very similar to gene X are dissimilar to gene Y .

The pseudo-code of constraining an input similarity matrix is shown in Figure 1. In case of must-link constrains, we impose the constraints by increasing the correlation between the must-linked pair to 1. This means that the genes in the must-linked pair are equivalent. Hence, the edge directly connecting two genes is a longest path between those genes. Notice that the weight of the edge indicates the correlation between those genes.

After imposing the constraints, we might incur violations on the imitative triangle inequality with respect to the Pearson's correlation coefficient and the uncentered correlation. For example, genes that were previously dissimilar may now become more similar along some path which skips through the constrained pairs. Therefore, we can run an all-pairs-longest-paths algorithm to adjust the imposed matrix. Some simple modifications on the Floyd-Warshall algorithm [4] allow us to do the all-pairs-longest-paths computation. Notice that the path must satisfy the imitative triangle inequality, i.e. equation 8.

For cannot-link constrains, we impose the constraints by setting the correlation between the cannot-linked pair as -1, meaning that the genes in the cannot-linked pair are most dissimilar. However, we will probably violate the imitative triangle inequality, i.e. Equation (8). Therefore, we must adjust the imposed matrix to satisfy the imitative triangle inequality. Unfortunately, it is a NP-complete problem to determine whether a satisfying assignment exists when cannot-links are present [8].

Hence, we choose an algorithm that implicitly produces the same effect during the clustering process. The Complete-link (CL) hierarchical agglomerative clustering algorithm provides a good solution, which merges clusters in order of similarity such that the more similar clusters are merged earlier. After each merging process, CL computes similarities between the new cluster and each of the old clusters. Since CL

Input: an m -by- m similarity matrix S , a constraint set C

Output: the constrained-based similarity matrix S

ConstrainSimilarities (Matrix S , Constraints C)

```

ImposeMustLinks ( $S, C$ )
PropagateMustLinks ( $S, C$ )
ImposeCannotLinks ( $S, C$ )
PropagateCannotLinks ( $S, C$ )

```

ImposeMustLinks (Matrix S , Constraints C)

```

for ( $i, j$ )  $\in C_{\text{must}}$ 
   $S_{ij} = S_{ji} = 1$ 
  for  $k \in \{1 : m\}$ 
     $S_{ik} = S_{ki} = S_{jk} = S_{kj} = \max \{S_{ik}, S_{jk}\}$ 

```

PropagateMustLinks (Matrix S , Constraints C)

```

 $S = \text{FastAllPairsLongestPaths} (S, C)$ 
for ( $i, j$ ) satisfy  $S_{ij} = 1$ 
   $C_{\text{must}} = C_{\text{must}} \cup \{(i, j)\}$ 

```

ImposeCannotLinks (Matrix S , Constraints C)

```

for ( $i, j$ )  $\in C_{\text{cannot}}$ 
   $S_{ij} = S_{ji} = -1$ 
for ( $i, k$ )  $\in C_{\text{must}}$ 
   $S_{ik} = S_{ki} = -1$ 
for ( $j, k$ )  $\in C_{\text{must}}$ 
   $S_{jk} = S_{kj} = -1$ 
for ( $i, k$ )  $\in C_{\text{must}}$ , for ( $j, l$ )  $\in C_{\text{must}}$ 
   $S_{kl} = S_{lk} = -1$ 

```

PropagateCannotLinks (Matrix S , Constraints C)

```

(done implicitly by Complete-Link)

```

FastAllPairsLongestPaths (Matrix S , Constraints C)

```

// find valid intermediates
 $I = \{i : \exists j \neq i, (i, j) \in C_{\text{must}}\}$ 
// modified Floyd-Warshall
for  $k \in I$ , for  $i \in \{1 : m\}$ , for  $j \in \{1 : m\}$ 
   $S_{ij} = S_{ji}$ 
   $= \max \{S_{ij}, 1 - (\sqrt{1 - S_{ik}} + \sqrt{1 - S_{kj}})^2\}$ 

```

Fig. 1. The pseudo-code of constraining an input similarity matrix

defines the similarity between two clusters as the minimum similarity from any member of one cluster to any member of the other clusters, the propagation of the cannot-link constraints can be implied through the merging.

3 Experimental Evaluations

We conducted a series of experiments to evaluate the accuracy of the C-CCL by testing a real gene expression dataset, namely the yeast sporulation data set [3]. The C-CCL is compared with the C-CKM (Correlational COP- K -Means) [11], which was extended from CKM so that it can process correlational data. The major modification is that the Pearson's correlation coefficient is used as similarity measure when assigning each gene to its most similar cluster. For C-CCL, we also use the Pearson's correlation coefficient as similarity measure for all experimental datasets. The quality of clustering results was measured by using *Rand index* and *Jaccard index*.

The tested yeast sporulation data set [3] consists of expression data of 6118 yeast genes, which were sampled at seven different time points during sporulation. According to their sequential induction patterns during sporulation, Chu *et al.* identified and grouped some genes in the data into seven groups, namely Metabolic (52 genes), Early I (61 genes), Early II (45 genes), Early-Mid (95 genes), Middle

(158 genes), Mid-Late (62 genes), and Late (4 genes). We take these 477 genes as the tested data and the seven groups as the standard for the clustering results.

For the C-CCL and the C-CKM, the input parameters k (cluster number) is set as 7. We tested several different mixtures of constraints, including 1) 100% must-links (ML), 2) 100% cannot-links (CL), 3) Equal proportion (Equal Prop.) and 4) Proportion to the relative number of pair types (Data Prop.). We first obtain all must-link and cannot-link constraints for all data pairs, then we generate the tested constraint sets by randomly choosing constraints from all must-link and cannot-link constraints according to the proportion of different mixed types. The same experiment was done for 10 times and the average was taken for a statistical purpose.

Figure 2 and 3 show the results of the C-CCL and the C-CKM for several different constraint mixes, respectively. It is observed that there is no sizable improvement over the unconstrained accuracy in all cases of the C-CKM. It means that C-CKM fails to take advantage of the constraints over a wide range of mix types. For C-CCL, the accuracy assessed by the Rand index rises quickly, as constraints are added in all cases except “Must-Link Only”. And the C-CCL’s accuracy assessed by the Jaccard index rises quickly, as constraints are added in cases of “Equal Proportion” and “Data Proportion”. Especially, the Jaccard index rises sharply in case of “Equal Proportion”.

By comparing all cases of the C-CCL in Figure 2 and 3, we infer that “Equal Proportion” might be the best mix type and “Data Proportion” is the next best mix type for this yeast sporulation data set. “Must-Link Only” and “Cannot-Link Only” may not be appropriate mix types, because the problem of inappropriate merging described previously might take place. It is shown that the C-CCL takes advantage of the constraints more efficiently than C-CKM. Moreover, the correlational propagation of C-CCL allows it to substantially outperform C-CKM.

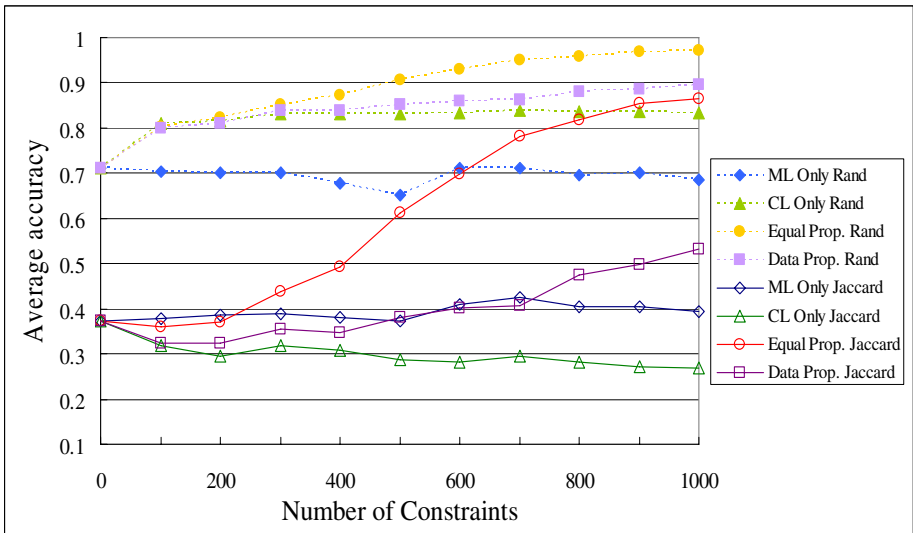


Fig. 2. The accuracy of C-CCL clustering on yeast data set over different constraint mixes

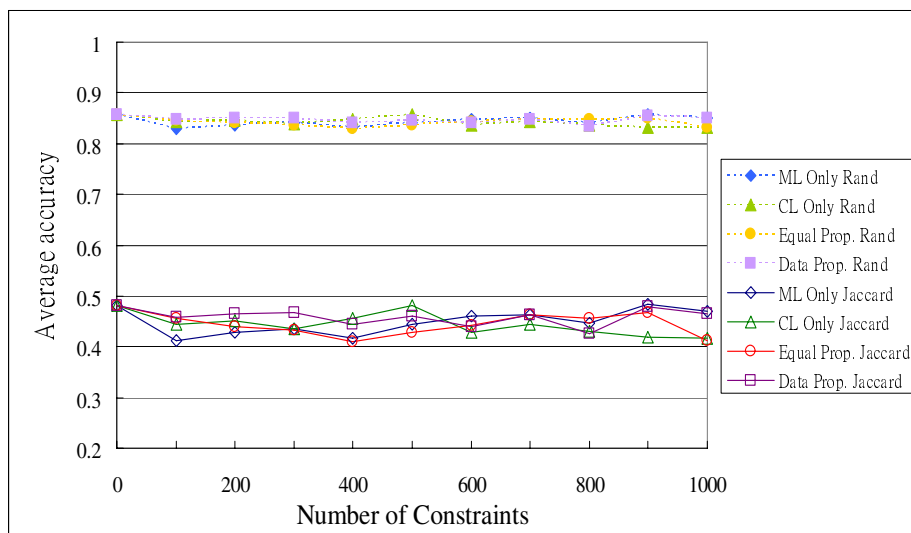


Fig. 3. The accuracy of C-CKM clustering on yeast data set over different constraint mixes

4 Conclusions

In this paper, we propose a new constrained clustering method, namely C-CCL, to improve the quality of hierarchical clustering for microarray data analysis in the presence of pair-wise gene constraints, while using correlation coefficients as similarity measure. Through empirical evaluations on a real yeast dataset, C-CCL was shown to utilize the given gene-pair constraints effectively such that the clustering accuracy is highly enhanced in terms of the Rand and Jaccard indexes.

In the future, we will explore some further issues. First, we will seek an imitative triangle inequality or a method to adjust the proximity with respect to the absolute correlation coefficients. Second, we will extend the C-CCL to capture the pattern structure embedded in the gene expression data sets. This might provide more insights for the functional relationships between genes.

Acknowledgment

This work was supported by the Landmark Project of National Cheng Kung University, Taiwan, R.O.C.

References

1. Basu, S., Banerjee, A., Mooney, R.J.: Semi-supervised Clustering by Seeding. In: Proceedings of the 9th International Conference on Machine Learning, pp. 19–26 (2002)
2. Cho, S.B., Ryu, J.: Classifying Gene Expression Data of Cancer Using Classifier Ensemble with Mutually Exclusive Features. Proceedings of IEEE 90, 1744–1753 (2002)

3. Chu, S., DeRisi, J., Eisen, M., Mulholland, J., Botstein, D., Brown, P.O., Herskowitz, I.: The Transcriptional Program of Sporulation in Budding Yeast. *Science* 282, 699–705 (1998)
4. Cormen, T.H., Leiserson, C.E., Rivest, R.L.: *Introduction to Algorithms*, 2nd edn. MIT Press, Cambridge (2001)
5. Davidson, I., Ravi, S.S.: Clustering With Constraints: Feasibility Issues and the k-Means Algorithm. In: *Proceedings of the SIAM International Conference on Data Mining* (2005)
6. Fisher, D.H.: Knowledge Acquisition via Incremental Conceptual Clustering. *Machine Learning* 2, 139–172 (1987)
7. Gordon, A.D.: *Classification*, 2nd edn. Monographs on Statistics and Applied Probability 82. Chapman and Hall/CRC, NY (1999)
8. Klein, D., Kamvar, S., Manning, C.: From Instance-level Constraints to Space-level Constraints: Making the Most of Prior Knowledge in Data Clustering. In: *Proceedings of the 9th International Conference on Machine Learning*, pp. 307–314 (2002)
9. Tseng, V.S., Kao, C.P.: Efficiently Mining Gene Expression Data via a Novel Parameterless Clustering Method. *IEEE/ACM Transactions on Computational Biology and Bioinformatics* 2, 355–365 (2005)
10. Wagstaff, K., Cardie, C.: Clustering with Instance-level Constraints. In: *17th International Conference on Machine Learning*, pp. 1103–1110 (2000)
11. Wagstaff, K., Cardie, C., Rogers, S., Schroedl, S.: Constrained k-means Clustering with Background Knowledge. In: *Proceedings of the 19th International Conference on Machine Learning*, pp. 577–584 (2001)

Concept Lattice–Based Mutation Control for Reactive Motifs Discovery

Kitsana Waiyamai, Peera Liewlom, Thanapat Kangkachit,
and Thanawin Rakthanmanon

Data Analysis and Knowledge Discovery Laboratory (DAKDL), Computer Engineering
Department, Engineering Faculty, Kasetsart University, Bangkok, Thailand
{kitsana.w, oprll, fengtpk, fengtwr}@ku.ac.th

Abstract. We propose a method for automatically discovering *reactive motifs*, which are motifs discovered from binding and catalytic sites, which incorporate information at binding and catalytic sites with bio-chemical knowledge. We introduce the concept of *mutation control* that uses amino acid substitution groups and conserved regions to generate complete amino acid substitution groups. Mutation control operations are described and formalized using a concept lattice representation. We show that a concept lattice is efficient for both representations of bio-chemical knowledge and computational support for mutation control operations. Experiments using a C4.5 learning algorithm with reactive motifs as features predict enzyme function with 72% accuracy compared with 67% accuracy using expert-constructed motifs. This suggests that automatically generating reactive motifs are a viable alternative to the time-consuming process of expert-based motifs for enzyme function prediction.

Keywords: mutation control, concept lattice, sequence motif, reactive motif, enzyme function prediction, binding site, catalytic site.

1 Introduction

There are many statistic-based motif methods for enzyme function prediction capable of high accuracy; however, most of these methods [2,3,4,5] avoid the direct usage of motifs generated from binding and catalytic sites to predict enzyme function prediction. These methods use other resources from surrounding sites that contain very few sequences of binding and catalytic sites. In certain applications, it is necessary to understand how motifs of binding and catalytic sites are combined in order to perform enzyme function prediction. This is a reason why the statistic-based motifs cannot completely replace expert-identified motifs. In this paper, we develop a method to predict enzyme functions based on direct usage of binding and catalytic sites. Motifs discovered from binding and catalytic sites are called *reactive motifs*. The principal motivation is that different enzymes with the same reaction mechanism at binding and catalytic sites frequently perform the same enzyme function. In previous work [16], we introduced a unique process to discover reactive motifs using *block scan filtering*, *Mutation Control*, and *Reactive Site-Group Definition*. The main step in reactive

motif discovery is mutation control whose objective is to determine a complete substitution group for each position in the sequences, such that the substitution group contains all possible amino acids that can be substituted.

In this paper, we show that the concept lattice provides an efficient representation of various types of bio-chemical background knowledge and efficient computational support for the operations of mutation control. We propose a method to construct an amino-acid property context from background knowledge which is Taylor Physico-Chemistry table [8]. From the amino-acid property context, the concept lattice representing a hierarchy of amino-acid substitution groups sharing the same properties is constructed. Each concept represents a substitution group; lattice operators are applied to obtain complete substitution groups. Reactive motifs generated from the concept-lattice mutation control step are used as input to the C4.5 learning algorithm to obtain the enzyme prediction model. The reactive motifs and PROSITE [1] motifs separately are used as training data for the C4.5 learning model, which is then evaluated using a test dataset containing 19,258 amino acid sequences of 235 known enzyme functions. The learning algorithm using reactive motifs as training data accurately identified 72.6% of the test sequences, compared to 67.25 % accuracy for PROSITE.

The overall process of reactive motif discovery is described in section 2. Section 3 gives details of the concept lattice-based mutation control; experimental results are presented in section 4, and conclusions are given in section 5.

2 Reactive Motifs Discovery with Mutation Control

In this section, we present an overall process of reactive motif discovery, consisting of three steps: *data preparation and block scan filtering*, *mutation control*, and *reactive site-group definition*. More details of reactive motif discovery process can be found in [16].

2.1 Data Preparation and Block Scan Filtering

In the data preparation step, we use an *enzyme sequence dataset* [10,11] that covers 19,258 enzyme sequences of 235 functions. Within this enzyme sequence dataset, we use sequences containing binding or catalytic sites. Designating the binding or catalytic site position as the center, binding or catalytic site sub-sequences are retrieved, each of length 15 amino acids, as shown in Fig. 1. These binding and catalytic site sub-sequences form a *binding and catalytic site* database. Sub-sequences in the binding and catalytic site database are then clustered into subgroups based on their reaction descriptions. There are in total 291 subgroups.

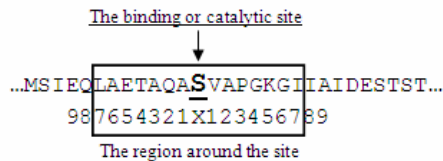


Fig. 1. Sequence with length of 15 amino acids around the binding and catalytic site

The purpose of the *block scan filtering* step is to alter each record of the binding and catalytic site database. For each binding or catalytic site sub-sequence, the dataset is scanned for all other sequences having the same site description, and a sequence similarity score is computed using amino-acid similarity scores such as BLOSUM62 [12]. The sequences are ranked according to similarity scores; then a block member filtering method [13] is applied. A block is designated as high quality when each site in the block has at least 3 positions presenting the same type of amino acids, as shown in Fig. 2.

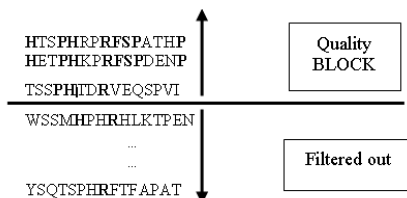


Fig. 2. Block member filtering to obtain a high quality block

2.2 Mutation Control

An enzyme mechanism can be represented by several binding or catalytic site sub-sequences. Therefore specific positions in sequences that control the properties of the enzyme mechanism have common or similar properties. Some positions in all sequences contain the same type of amino acids; these positions are called *conserved regions*. Other positions may have many types of amino acids, but having similar properties. All amino acids in the same position are grouped with respect to the mutation in biological evolution and the resulting group is called a *substitution group*. Therefore, a substitution group is a set of amino acids having common or similar properties that can be substituted at a specific position in a block. There are two kinds of substitution groups, represented by patterns as in the PROSITE motifs:

- (1) A group of amino acids having some common properties; the substitutable amino acids are listed in brackets, for example [HT].
- (2) Amino acids having *prohibited properties* cannot be included at a position in the group. Prohibited amino acids are listed in braces, for example {P}, meaning any amino acid except P.

Mutation control constructs a motif consisting of the complete substitution group or conserved region from each position in the sequence. Using the results of the block scan filter step, all amino acids in the same position are compared and analyzed. Mutation control extends each amino acids substitution group to include all amino acids having common characteristics, identified using the Taylor physico-chemistry table (Table 1), to create a *complete substitution group*. This extension process is described next.

Table 1. Physico-chemistry table representing background knowledge of amino acids properties

	Small	Tiny	Proline	Polar	Charge	Positive	Negative	Hydrophobic	Aromatic	Aliphatic
A	X	X						X		
C	X	X		X				X		
D	X			X	X		X			
E				X	X		X			
F								X	X	
G	X	X						X		
H				X	X	X		X	X	
I								X		X
K				X	X	X		X		
L								X		X
M								X		
N	X			X						
P	X		X							
Q				X						
R				X	X					
S	X	X		X						
T	X			X				X		
V	X							X		X
W								X	X	
Y				X				X	X	

A complete substitution group is constructed by examining both the *common properties* and *boundary properties* at a given position. In some positions, there may be many types of amino acids that yield the same enzyme reaction mechanism. These amino acids have common or similar properties. For example, the amino-acids substitution group [HT] has *Polar* and *Hydrophobic* as common properties, which are necessary for an enzyme mechanism to function.

The *prohibited properties* are all the properties that are not found by any member of the substitution group. For example, the prohibited properties of [HT] are *Tiny*, *Negative*, and *Aliphatic*. The *boundary properties* set is the complement of the prohibited properties. The boundary properties and common properties are used together to identify the complete substitution group.

To be certain that a given substitution group contains all possible amino-acids that can be substituted, the mutation control extends each substitution group to include all amino acids that have *all* the common properties and *only* properties in the boundary set (i.e. no prohibited properties). For example, complete substitution group for [HT] is [HTWYK]. This is the greatest amino acid substitution group that has all common properties and the only properties they have are boundary properties. This complete substitution group is determined at all other positions of the quality block to produce a motif. For the quality block in Fig. 2, we obtain the motif [HTWYK] [CDENQST] [CNST] P H [KNQRT] [DNP] R [FILMV] [DENQS] [ACDGNST] . . .

The source of background information can be used in block scan filtering and mutation control should be the same. For example, if the BLOSUM62 table is used as the similarity score table in *block scan filtering step*, the amino acids properties table transformed from BLOSUM62 should be used in the *mutation control* step. More details about background knowledge transformation can be found in [16].

2.3 Reactive Site – Group Definition

From the previous step, motifs produced from different records of the same binding or catalytic functions are, by definition, redundant. They are grouped together and represented as one *reactive motif* in a grouping process called *reactive site–group definition*. Although motifs are retrieved from the same original binding or catalytic sites in the same subgroup of the binding and catalytic site database, they can have different binding structures to the same substrate. In other words, there are many ways

to “fit and function”. As a result of this step, 1,328 reactive motifs are constructed using the BLOSUM62 data and 1,390 using the Taylor physico-chemistry table.

3 Concept Lattice–Based Mutation Control for Complete Substitution Group Discovery

In this section, we apply concept lattice theory [17,18] to mutation control in order to determine complete substitution groups. From the amino acids context, the concept lattice is generated, where concepts are constructed as amino acids substitution groups sharing common properties. The generated concept lattice represents hierarchy of amino acids substitution groups sharing common properties. From this lattice, mutation control operations are performed to determine complete amino-acid substitution groups. We start by giving some basic definitions of concept lattices as applied to mutation control. Then, concept lattice-based mutation control operations are defined.

3.1 Basic Definitions

Amino acid properties context: An amino acid properties context is a triple (Σ, P, R) , where Σ and P are finite sets of amino acids and properties, and $R \subseteq \Sigma \times P$ is a binary relation. eRp denotes that the amino acid $e \in \Sigma$ is in relation R to the property $p \in P$, if e has the property p (or e verifies property p).

Concept: A concept is a pair $(Extent, Intent)$ where $Extent \subseteq \Sigma$, $Intent \subseteq P$ and $f(Extent) = Intent$ and $g(Intent) = Extent$. Let L be a set of all concepts formed from the context (Σ, P, R) , and let $c \in L$. Hence, c is formed by two parts: an extent representing a subset of Σ (here, amino acids), denoted as $Extent(c)$, and an intent representing the common properties between this subset of amino acids, denoted as $Intent(c)$. For example, $(\{A,C,G\}, \{small, tiny, hydrophobic\})$ is a concept of the context in Table 1. This means that there are no more than three amino acids possessing at least all properties in $\{small, tiny, hydrophobic\}$ and sharing at most these properties in common. The concept’s extent is an amino-acid substitution group sharing similar properties.

Amino acid properties concept lattice: An amino acid properties concept lattice is a concept lattice $L = (L, \leq)$ of an amino acid properties context (Σ, P, R) , is a complete lattice of concepts derived from the amino acid properties context. The lattice structure imposes:

- a partial ordering on concepts such that for concepts $c1, c2 \in L$, $c1 \leq c2$, iff $Extent(c1) \subseteq Extent(c2)$ or, equivalently, $Intent(c2) \subseteq Intent(c1)$.
- any concept subset of L has one greatest subconcept (the Meet element) and one least superconcept (the Join element).

Theorem. Let (Σ, P, R) be a context, let L be a concept lattice of concepts derived from (Σ, P, R) and $S \subseteq L$. The Meet(S) and Join(S) elements are given as follows:

$$Meet(S) = (\bigcap_{c \in S} Extent(c), f(g(\bigcup_{c \in S} Intent(c))) \tag{1}$$

$$Join(S) = (g(f(\bigcup_{c \in S} Extent(c)), \bigcap_{c \in S} Intent(c))) \tag{2}$$

3.2 Complete Amino-Acids Substitution Group Discovery

In this section, we present a method for finding complete amino acid substitution group at a given position of a block of amino acids resulted from the block scan filtering step (section 2.1). Our method works in 4 steps. First, it starts by finding smallest object concept for each amino acid in the amino acid-properties lattice. Then, it uses those concepts to find candidate substitution groups having the greatest common properties and having the greatest boundary properties. Finally, it returns the common amino acids of both substitution groups as the complete amino-acid substitution group.

3.2.1 Finding Amino Acid Concepts

Each amino acid in the same position of a block is used for finding its introduction concept in the amino acid-properties lattice called *amino-acid concept* [19]. Considering Fig. 3, ($\{H\}$, $\{aro,hyd,cha,pol,pos\}$) and ($\{T,C\}$, $\{hyd,sma,pol\}$) are introduction concepts of amino-acids H and T.



Fig. 3. Shows two candidate-substitution groups of amino acids {H, T} which are represented by gray nodes. The black node represents candidate substitution group having the greatest common properties, derived from the gray edges, while candidate substitution group having the boundary properties, represented by dotted node, can be derived from the dotted edges.

3.2.2 Finding Candidate Substitution Group Having Common Properties

According to an important characteristic of a substitution group (described in section 2.2), complete substitution group should have common properties. In order to determine the substitution group having common properties at most or *greatest set of common properties*, the lattice operator $Join(S)$ is applied where S is a set of amino-acid concepts derived from the previous section. $Join(S)$ returns a concept whose intent contains greatest common properties of S and whose extent is a candidate substitution group.

In the following, we show how the greatest common properties of amino acids $\{H,T\}$ and its candidate substitution group can be determined. From the previous step, we obtained $(\{H\},\{aro,hyd,cha,pos\})$ and $(\{T,C\},\{hyd,sma,pos\})$ as amino-acid concepts represented as gray nodes in the Fig. 3. Then, we use them as input to the $Join(S)$ operator. $(\{W,H,Y,K,T,C\},\{hyd,pos\})$ is the result of $Join(S)$ whose extent represents candidate substitution group of amino acids $\{H,T\}$.

3.2.3 Finding Candidate Substitution Group Having Boundary Properties

According to the definition of a substitution group (described in Section 2.2), a complete substitution group should exclude any amino acid having the prohibited properties that prevent the enzyme mechanism function. The substitution group having the greatest set of boundary properties is the result of the union of the extent of all super-concept of the lattice operator $Meet(S)$, where S is a set of amino-acid concepts as described in Section 3.2.1. In the case that $Meet(S)$ produces a concept whose intent contains any prohibited properties, a virtual boundary concept will be used instead. The intent of the virtual boundary concept includes only the greatest boundary properties and its extent is an empty set. A virtual boundary concept can be formally defined as follows:

Definition: Let $(\sum P,R)$ be a context, L be a concept lattice derived from $(\sum P,R)$, and $S \subseteq L$. A concept $(\emptyset, \bigcup_{c \in S} Intent(c))$ is called a virtual boundary concept if $Meet(S) = (\emptyset, I)$ and $I \not\subseteq \bigcup_{c \in S} Intent(c)$.

In the following, we show how the greatest set of boundary properties of amino acids $\{H,T\}$ and their candidate substitution group can be determined. From Section 3.2.1, we obtain $S = \{(\{H\},\{aro,hyd,cha,pos\}), (\{T,C\},\{hyd,sma,pos\})\}$ as a set of amino-acid concepts represented by gray nodes in the Fig. 3. Then, $Meet(S)$ results the bottom concept $(\{\},\{sma,tin,aro,neg,ali,hyd,cha,pos\})$. In this case, the intent of result concept contains prohibit properties such as $\{tin, pros, neg, ali\}$. Thus, a virtual boundary concept $(\{\},\{aro,hyd,cha,pos,sma\})$ is created. We then link it as the immediate predecessor concept of the bottom concept. Then, we determine its immediate predecessor concepts by choosing the immediate predecessor concepts of the bottom concept having no prohibited properties, which is the set of concepts $\{(\{H\},\{aro,hyd,cha,pos\}), (\{T,C\},\{hyd,cha,pos\})\}$, represented by a dashed node in Fig. 3. Finally, from the set of super-concepts of the virtual boundary concept, we select only object concepts. Then, the union of the extent of those object concepts is the substitution group having boundary properties $\{H,T,W,Y,F\}$.

3.2.4 Complete Amino Acid Substitution Group

Once both candidate substitution groups are extracted from the previous step, a complete amino acid substitution group can be determined by finding the common amino acids appearing in both substitution groups. From Fig. 3, amino acids having common properties are {W,H,Y,K,T,C}; while amino acids having the boundary properties are {H,T,W,Y,F}. Thus, the amino acids that appear in both substitution groups form the complete substitution group {H,T,W,Y} of amino acids {H, T}, as required.

4 Experimental Results

We performed experiments using a dataset containing 19,258 protein sequences that covers 235 enzyme functions, using the C4.5 learning algorithm with a 5-fold cross validation.

The accuracy of the enzyme function prediction models is shown in Table 2. Each prediction model is constructed using reactive motifs generated from different background knowledge. The model constructed with reactive motifs generated using BLOSUM62 is called *BLOSUM – reactive motif*. The model constructed with reactive motifs generated using Taylor’s physico-chemistry table is called *physico-chemistry – reactive motif*. The reference model, called *conserved amino acid – reactive motif*, is constructed using reactive motifs without a substitution group. These reactive motifs are generated from conserved regions using BLOSUM62. In case the *conserved region-group definition* step is not applied, the *BLOSUM – reactive motifs* model gives the best results with 68.69% accuracy. The prediction model using physico-chemistry – reactive motifs with application of conserved region-group definition gives the best accuracy, 72.58%; however, the accuracies of all models are very close.

Table 3 shows the prediction accuracy of enzyme function prediction model, with respect to different class members using PROSITE motifs. The accuracy of the prediction model retrieved from PROSITE motifs gives the best accuracy of 67.25%.

Table 2. Accuracy comparison among function prediction models using reactive motifs

Reactive site– group definition	Reactive motifs					
	Conserved amino acid		BLOSUM		Physicochemistry	
	# motif	C4.5 (%)	# motif	C4.5 (%)	# motif	C4.5 (%)
From Binding and Catalytic Site Database	291	60.84	291	68.69	291	64.38
Conserved region – group definition	1324	70.57	1328	71.66	1390	72.58

Table 3. Accuracy of function prediction models using PROSITE motifs

#Members	# Functions	# Motifs	# Sequences	C4.5 (%)
Between 10 and 1000	42	36	2579	37.15
Between 5 and 1000	76	65	2815	67.25

5 Conclusions and Discussion

In this paper, we show that concept lattice is an efficient representation of biochemistry background knowledge and an efficient computational support for mutation control operations. To obtain an enzyme prediction model, reactive motifs generated from the concept lattice based mutation control step are used as the input to C4.5 learning algorithm. Our enzyme prediction model yields good results (~70% accuracy of enzyme function prediction) and can overcome problems such as lack of protein or enzyme functional information; only about ~5.8% in our dataset contain information about binding and catalytic sites. The reactive motifs using physico-chemistry background knowledge give the best results; although the coverage value is not satisfied, the number of reactive motifs found per enzyme sequence is very good. It indicates the motifs are very specific.

The limited improvement in accuracy observed when using the conserved region group definition indicates that the details of the mechanism descriptions need further improvement. Improving the quality of the descriptions of binding and catalytic sites would, in the authors' view, further increase the accuracy of enzyme function prediction using reactive motifs.

Acknowledgement. Thanks to J. E. Brucker for his reading and comments of this paper.

References

1. Bairoch, A.: PROSITE: a dictionary of sites and patterns in proteins. *Nucleic Acids Res.* 19, 2241–2245 (1991)
2. Sander, C., Schneider, R.: Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins Struct. Funct. Genet.* 9, 56–68 (1991)
3. Huang, J.Y., Brutlag, D.L.: The EMOTIF database. *Nucleic Acids Res.* 29, 202–204 (2001)
4. Eidhammer, I., Jonassen, I., Taylor, W.R.: Protein structure comparison and structure patterns. *Journal of Computational Biology* 7(5), 685–716 (2000)
5. Bennett, S.P., Lu, L., Brutlag, D.L.: 3MATRIX and 3MOTIF: a protein structure visualization system for conserved sequence. *Nucleic Acids Res.* 31, 3328–3332 (2003)
6. Henikoff, S., Henikoff, J.G.: Automated assembly of protein blocks for database searching. *Nucleic Acids Res.* 19, 6565–6572 (1991)
7. Barton, G.J.: Protein multiple sequence alignment and flexible pattern matching. *Methods Enzymol.* (183), 403–428 (1990)
8. Taylor, W.R.: The classification of amino acid conservation. *J. Theor. Biol.* 119(2), 205–218 (1986)
9. Wu, T.D., Brutlag, D.L.: Discovering Empirically Conserved Amino Acid Substitution Groups in Databases of Protein Families. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* (4), 230–240 (1996)
10. Bairoch, A., Apweiler, R.: The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.* 28, 45–48 (2000)

11. Nomenclature Committee of the International Union of Biochemistry and Molecular Biology (NC-IUBMB). *Enzyme Nomenclature. Recommendations 1992*. Academic Press (1992)
12. Henikoff, S., Henikoff, J.G.: Amino acid substitution matrices from protein blocks. *Proc. Natl. Acad. Sci. USA* (89), 10915–10919 (1992)
13. Smith, H.O., Annau, T.M., Chandrasegaran, S.: Finding sequence motifs in groups of functionally related proteins. *Proc. Natl. Acad. Sci. U S A* 87(2), 826–830 (1990)
14. Diplaris, S., Tsoumakas, G., Mitkas, P.A., Vlahavas, I.: Protein Classification with Multiple Algorithms. In: *Proc. of 10th Panhellenic Conference in Informatics, Volos, Greece, November 21-23*. LNCS. Springer, Heidelberg (2005)
15. Frank, E., Hall, M., Trigg, L., Holmes, G., Witten, I.H.: Data mining in bioinformatics using Weka. *Bioinformatics* 20(15), 2479–2481 (2004)
16. Liewlom, P., Rakthanmanon, P., Waiyamai, K.: Prediction of Enzyme Class using Reactive Motifs generated from Binding and Catalytic Sites. In: Alhajj, R., Gao, H., Li, X., Li, J., Zaiane, O.R. (eds.) *ADMA 2007*. LNCS (LNAI), vol. 4632. Springer, Heidelberg (2007)
17. Wille, R.: Restructuring lattice theory: an approach based on hierarchies of concepts. In: Rival, I. (ed.) *Ordered sets*, Dordrecht–Boston, pp. 445–470 (1982)
18. Waiyamai, K., Taouil, R., Lakhil, L.: Towards an object database approach for managing concept lattices. In: Embley, D.W. (ed.) *ER 1997*. LNCS, vol. 1331, pp. 299–312. Springer, Heidelberg (1997)
19. Arévalo, G., Berry, A., Huchard, M., Perrot, G., Sigayret, A.: Performances of Galois Sub-hierarchy-building Algorithms. In: Kuznetsov, S.O., Schmidt, S. (eds.) *ICFCA 2007*. LNCS (LNAI), vol. 4390, pp. 166–180. Springer, Heidelberg (2007)

Mining a Complete Set of Both Positive and Negative Association Rules from Large Databases

Hao Wang, Xing Zhang, and Guoqing Chen

Department of Management Science and Engineering, Tsinghua University,
Beijing 100084, China
w-hao02@mails.tsinghua.edu.cn

Abstract. Association rule mining is one of the key issues in knowledge discovery. In recent years, negative association rule mining has attracted remarkable attention. This paper presents a notion of validity for both positive and negative association rules, which is considered intuitive and necessary. Then, a mining algorithm to find all rules in light of completeness is proposed. In doing so, several pruning strategies based on the upward closure property are developed and incorporated into the algorithm so as to guarantee the computational efficiency.

Keywords: Negative association rules, upward closure, Apriori, data mining.

1 Introduction

As one of the promising areas of research for knowledge discovery, association rule mining (ARM) attempts at finding the relationships between the different items in databases [1-3]. Researchers have extended the association rule (AR) concept — originally specific to binary data tables— to a multitude of domains, involving quantitative, hierarchical, fuzzy, and many other kinds of databases. The main characteristic of the efforts is to predict the presence of some data items (itemsets) from the presence of other data items. In other words, the focal point of interest is the positive association of itemsets, namely, a presence-to-presence relationship. On the other hand, in many real applications, negative associations (i.e., the relationship between the presence and the absence of itemsets, or the absence and the absence of itemsets) are meaningful and therefore attracting more and more attention nowadays (e.g., [4-13]). For example, “office workers who did NOT buy cars turned to rent homes near subway stations”, and “customers who were NOT interested in big screen mobile phones would NOT buy other value-added services (e.g., games, web connections, etc.)”. These kinds of ARs reflect certain negative patterns of data items and are usually referred to as negative ARs.

Mining negative ARs, however, raises a number of critical issues [13]. First, the density of data in databases becomes much higher. Second, the computational cost may skyrocket when each item in the database and its corresponding negated item (indicating absence of the original item) are considered, since the mining complexity may increase significantly in terms of the number of data items. Moreover, negative

ARs may invalidate some important pruning strategies used to restrict the search space and guarantee efficiency in classical ARM algorithms.

In order to address these issues and explore efficient algorithms, a number of efforts have been made to develop improvements and extensions. Savasere et al. [5] and Yuan et al. [8] incorporate domain knowledge (taxonomy structure) into the algorithms. These approaches compare expected support or confidence of an itemset (based on itemsets' positions in the taxonomy structure) with the actual value of these measures. The limitations of these approaches are: first, negative ARs are mainly restricted to relative negative ARs compared with other sibling itemsets; and second, the domain knowledge (taxonomy structure) needed may often not be readily available. Wu et al. [12] and Antonie et al. [9] focus on certain notions of negative ARs (such as $\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$, $\neg X \Rightarrow \neg Y$) and present approaches to mine them. However, it is found that their approaches can hardly guarantee to generate a complete set of valid rules, that is, some valid negative ARs defined may not be obtained using their algorithms [13]. Furthermore, Brin et al. [4], and Cornelis et al. [13] concentrate on similar notions of negative ARs and provide algorithms to generate all rules of concern. However, their notions are, though meaningful, restrictive in semantics and deemed necessary to extend.

In this paper, another notion of validity for both positive and negative ARs is presented, which reflects semantics in a broader sense and appears to be intuitive. Then, a mining algorithm is proposed, which is sound and complete in terms of generating all rules of interest. Pruning strategies based on the upward closure property are developed and incorporated into the algorithm so as to guarantee the computational efficiency.

2 Valid Association Rules

In association rule mining, two measures, namely the Degree of Support (*supp*) and the Degree of Confidence (*conf*), are used to define a rule [2-3]. According to [9, 12-13], In the case of the negation of a set of items (itemset) X , denoted by $\neg X$, the degree of support is: $supp(\neg X) = 1 - supp(X)$. A rule of the form $X \Rightarrow Y$ is called a positive rule, whereas rules of the other forms ($\neg X \Rightarrow Y$, $X \Rightarrow \neg Y$, $\neg X \Rightarrow \neg Y$) are negative rules. Specifically, the degrees of support and the degrees of confident are defined as follows:

$$\begin{aligned} supp(X \Rightarrow Y) &= supp(X \cup Y) \\ supp(\neg X \Rightarrow Y) &= supp(Y) - supp(X \cup Y) \\ supp(X \Rightarrow \neg Y) &= supp(X) - supp(X \cup Y) \\ supp(\neg X \Rightarrow \neg Y) &= 1 - supp(X) - supp(Y) + supp(X \cup Y) \\ conf(C_1 \Rightarrow C_2) &= \frac{supp(C_1 \Rightarrow C_2)}{supp(C_1)} \end{aligned}$$

where $X \cap Y = \emptyset$, $C_1 \in \{X, \neg X\}$, $C_2 \in \{Y, \neg Y\}$. Subsequently, we have definition 1 for valid association rules.

Definition 1 (valid association rule). Let X and Y be itemsets. A valid association rule (AR) is an expression $C_1 \Rightarrow C_2, C_1 \in \{X, \neg X\}, C_2 \in \{Y, \neg Y\}, X \cap Y = \emptyset$, such that $posbound(C_1 \Rightarrow C_2) = 1$ and $negbound(C_1 \Rightarrow C_2) = 1$, where $posbound$ and $negbound$ are mappings from the set of possible ARs to $\{0, 1\}$ with:

$$posbound(C_1 \Rightarrow C_2) = \begin{cases} 0 & \text{if } Supp(C_1 \cup C_2) < ms \text{ or } Supp(X) < ms \\ & \text{or } Supp(Y) < ms \text{ or } conf(C_1 \Rightarrow C_2) < mc \\ 1 & \text{otherwise} \end{cases}$$

$$negbound(C_1 \Rightarrow C_2) = \begin{cases} 0 & \text{if } C_2 = \neg Y \text{ and } \exists Y' \subset Y, \text{ s.t. } posbound(C_1 \Rightarrow \neg Y') = 1 \\ 1 & \text{otherwise} \end{cases}$$

It is worth mentioning that, according to the definition, if $C_1 \Rightarrow \neg Y$ is valid, then there should not exist $Y' \subset Y$ such that $C_1 \Rightarrow \neg Y'$ is also valid. This is based on the fact that, if $posbound(C_1 \Rightarrow \neg Y') = 1$, then $posbound(C_1 \Rightarrow \neg Y) = 1$ is always true, for all $Y' \subset Y$. Moreover, it is important to note that several useful properties hold as follows, which can be used in pruning strategies for efficiency and to guarantee the completeness of the proposed AR mining algorithm that will be discussed in the next section.

Property 1

(1.1) $supp(X) \geq supp(X')$, for all $X \subseteq X'$ (downward closure)

(1.2) $supp(\neg X') \geq supp(\neg X)$, for all $X \subseteq X'$ (upward closure)

Property 2

(2.1) $supp(C_1 \Rightarrow Y) \geq supp(C_1 \Rightarrow Y')$, for all $Y \subseteq Y'$.

(2.2) $supp(\neg X' \Rightarrow C_2) \geq supp(\neg X \Rightarrow C_2)$, for all $X \subseteq X'$.

(2.3) $supp(C_1 \Rightarrow \neg Y') \geq supp(C_1 \Rightarrow \neg Y)$, for all $Y \subseteq Y'$.

Property 3

(3.1) $conf(C_1 \Rightarrow Y) \geq conf(C_1 \Rightarrow Y')$, for all $Y \subseteq Y'$.

(3.2) $conf(C_1 \Rightarrow \neg Y') \geq conf(C_1 \Rightarrow \neg Y)$, for all $Y \subseteq Y'$.

Property 4

Let $\overline{conf}(\neg X \Rightarrow Y) = \frac{supp(Y)}{1 - supp(X)}$ then,

(4.1) $\overline{conf}(\neg X \Rightarrow Y) \geq \overline{conf}(\neg X \Rightarrow Y)$.

(4.2) $\overline{conf}(\neg X \Rightarrow Y) \geq \overline{conf}(\neg X \Rightarrow Y')$, for $Y \subseteq Y'$.

3 Generating Valid Association Rules

As can be seen in previous discussions, all valid association rules of concern, namely the AR space, are composed of four types of ARs. In other words, the AR space could

be partitioned into four parts: Part I: positive valid ARs, in forms of $X \Rightarrow Y$; Part II: negative valid ARs, in forms of $\neg X \Rightarrow Y$; Part III: negative valid ARs, in forms of $X \Rightarrow \neg Y$; and Part IV: negative valid ARs, in forms of $\neg X \Rightarrow \neg Y$.

The mining process, therefore, constitutes four major steps to generate all frequent itemsets and all valid ARs in part I, all negative valid ARs in part II, III and IV respectively. For part I, all frequent itemsets and valid ARs could be generated using the Apriori-type approaches etc. While the effective Apriori's pruning strategy based on downward closure property (Property 1.1) still pertains in generating part I, it does not suit in generating parts II, III, and IV. Hence, new pruning strategies, say, based on upward closure property (Property 1.2) and other properties mentioned in section 2, need to be developed so as to enable an effective and efficient generation of all negative valid ARs.

3.1 Pruning Strategies

In addition to property 1.1 used as a pruning strategy to generate positive valid ARs, other pruning strategies are needed in generating negative valid ARs. According to properties 1, 2 and 3, an important property (Property 5) can be derived, which is downward-closure-like and could be incorporated in the mining process for part II as a pruning strategy. Furthermore, in discovering valid ARs in forms of $C_1 \Rightarrow \neg Y$ (i.e., valid ARs in parts III and IV), another proven property (Property 6) is important as well.

Property 5. If $\neg X \Rightarrow Y'$ is valid, then $\neg X \Rightarrow Y$ ($Y \subset Y'$) is valid.

Property 6. If $C_1 \Rightarrow \neg Y$ is valid, then $C_1 \Rightarrow \neg Y'$ ($Y' \subset Y$) is not valid.

Property 5 enables us to generate valid ARs of the form $\neg X \Rightarrow Y$ by extending the consequents of already obtained valid ARs and prune candidate ARs by examining their consequents' $(k-1)$ -length sub-itemsets in valid ARs (k is the length of their consequents). Property 6 enables us to use potential ARs (not valid ARs, but having potential to generate valid rules by extending their consequents) to generate valid ARs by extending the consequents of them and prune candidate ARs by examining their consequents' $(k-1)$ -length sub-itemsets in potential ARs.

Notably, the difference between the Apriori-type approach and the proposed approach for parts II, III and IV is that the former uses frequent itemsets to generate and prune candidate itemsets, whereas the latter uses valid ARs to generate and prune candidate ARs for part II, and uses potential ARs for parts III and IV. More details are presented in the following subsection.

3.2 Algorithmic Details

The following notations are used in discussing algorithmic ideas.

X, Y : positive itemsets;

$|X|, |Y|$: the number of items in X, Y

$L(P_1)$: frequent itemsets in part I;

$L(P_1)_k$: k -length frequent itemsets in part I;

$VR(P_i)$: valid ARs in part i ;

$VR(P_i)_{k,p}$: valid ARs with k -length antecedent and p -length consequent in part i ;

$NR(P_i)$: potential ARs; not valid ARs, but having potential to generate valid rules by extending its consequent in part i . It is used to generate $CR(P_i)$ for parts III and IV.

$CR(P_i)$: candidate ARs in part i , including $VR(P_i)$ and $NR(P_i)$.

$S(P_i)$: positive itemsets whose support needs to be calculated via DB scan in part i ;
(analogously: $NR(P_i)_{k,p}$, $CR(P_i)_{k,p}$, $S(P_i)_{k,p}$).

Procedure 1. Generate all Negative Valid ARs in Parts II, III and IV (with minimal confidence mc)

1: $VR(P_i) = \emptyset$

2: $CR(P_i)_{i,1} = \begin{cases} \{\neg X \Rightarrow Y \mid X, Y \in L(P_1)_1, X \cap Y = \emptyset, \overline{\text{conf}}(\neg X \Rightarrow Y) \geq mc\} & \text{for } i=\text{II} \\ \{X \Rightarrow \neg Y \mid X, Y \in L(P_1)_1, X \cap Y = \emptyset\} & \text{for } i=\text{III} \\ \{\neg X \Rightarrow \neg Y \mid X, Y \in L(P_1)_1, X \cap Y = \emptyset\} & \text{for } i=\text{IV} \end{cases}$

3: **for** $\{k = 1; CR(P_i)_{k,1} \neq \emptyset; k++\}$ **do**

4: **for** $\{p = 1; CR(P_i)_{k,p} \neq \emptyset; p++\}$ **do**

5: generate $S(P_i)_{k,p}$

6: compute support of all itemsets in $S(P_i)_{k,p}$

7: generate $VR(P_i)_{k,p}$ and $NR(P_i)_{k,p}$

8: $VR(P_i) = VR(P_i) \cup VR(P_i)_{k,p}$

9: generate $CR(P_i)_{k,p+1}$

10: delete $NR(P_i)_{k,p}$ for $i \neq \text{II}$

11: **end for**

12:

$CR(P_i)_{k+1,1} = \begin{cases} \{\neg X \Rightarrow Y \mid X \in L(P_1)_{k+1}, Y \in L(P_1)_1, X \cap Y = \emptyset, \overline{\text{conf}}(\neg X \Rightarrow Y) \geq mc\} & \text{for } i=\text{II} \\ \{X \Rightarrow \neg Y \mid X \in L(P_1)_{k+1}, Y \in L(P_1)_1, X \cap Y = \emptyset\} & \text{for } i=\text{III} \\ \{\neg X \Rightarrow \neg Y \mid X \in L(P_1)_{k+1}, Y \in L(P_1)_1, X \cap Y = \emptyset\} & \text{for } i=\text{IV} \end{cases}$

13: **end for**

Procedure 1 first generates candidate, valid and potential ARs with k -length antecedents and 1-length consequents, then generates candidate, valid and potential ARs with k -length antecedents and $p+1$ -length consequents from valid ARs (for part II) or from potential ARs (for parts III and IV) with k -length antecedents and p -length

consequents. More concretely, let us consider certain lines of algorithmic treatments in Procedure 1 as follows, whereas corresponding (sub-) procedural codes are omitted due to the limitation of space.

For line 5, $S(P_i)_{k,p}$ is positive itemsets, the support of any element in it is unknown. It is needed in the computation of *pospound* for generating $VR(P_i)_{k,p}$ and $NR(P_i)_{k,p}$. Line 6 computes itemsets in $S(P_i)_{k,p}$ via database scan. For line 9, $CR(P_i)_{k,p+1}$ from $VR(P_i)_{k,p}$ for part II and from $NR(P_i)_{k,p}$ for parts III and IV are generated, in that pruning strategies discussed in subsection 3.1 are used for $CR(P_i)_{k,p+1}$.

Note that the generation of $CR(P_i)_{k,p+1}$ is only related to $VR(P_i)_{k,p}$ (for part II) or $NR(P_i)_{k,p}$ (for parts III and IV), the generation of $CR(P_i)_{k,p+1}$ can start after the generation of $VR(P_i)_{k,p}$ (for part II) or $NR(P_i)_{k,p}$ (for parts III and IV), and do not have to wait generations of other ARs with \hat{k} -length antecedents ($\hat{k} \neq k$). This is a very good feature that parallel computing may be possible, where dynamically specified cores (or processors) could be executed for Procedure 1 from lines 4 to 12 with certain parallel computing algorithms.

Importantly, it can be proven that the above-mentioned algorithm will generate a complete set of all positive and negative valid ARs. That is, the proposed approach (and the corresponding algorithm) is both sound and complete. It is also easy to show that the proposed approach in this paper is considered advantageous over existing ones (e.g., [4-5], [8-9], [12-13]) in terms of meaningfulness in rule validity and completeness in rule generation.

4 Experiment Results

To study the effectiveness of our approach, we have performed data experiments based on synthetic databases generated by IBM Synthetic Data Generator for Associations and Sequential Patterns (http://www.cse.cuhk.edu.hk/~kdd/data_collection.html). In the experiments, we used C++ on a Lenovo PC with 3G of CPU and 4GB memory. The main parameters of the databases are as follows. The total number of attributes is 1000; the average number of attributes per row is 10; the number of rows is 98358, approximately 100K; the average size of maximal frequent sets is 4.

The experiments were to illustrate that, though data-dependent, there are much more negative ARs than positive ones due to the nature of negation of data items semantically, and that the proposed approach (namely Algorithm VAR) is effective in generating negative ARs in terms of throughput rate (number of rules per time unit), which is higher than that of Apriori algorithm [3] (namely Apriori) for generating positive ARs. Table 1 shows the results.

Table 1. Running time (seconds) and numbers of positive and negative ARs

<i>Ms</i>	<i>Mc</i>	Positive rules (generated by Apriori)			Negative rules (generated by VAR)		
		Time	Number	Number /Time	Time	Number	Number /Time
0.001	0.6	41	91045	2221	1868	17345516	9286
0.001	0.7	40	86004	2150	1863	17346435	9311
0.001	0.8	40	68654	1716	1890	17357488	9184
0.001	0.9	40	37438	936	1913	17367665	9079
0.015	0.6	24	6211	259	307	5573850	18156
0.015	0.7	24	5947	248	309	5572996	18036
0.015	0.8	24	5488	229	309	5569366	18024
0.015	0.9	24	4092	171	313	5554373	17746

Moreover the advantage became larger with the increase in minimal support ms . The fact that Number/Time in VAR decreased with the increase in minimal confidence mc is because larger mc made the negative ARs' negative consequents to become longer to satisfy it. For example, if $supp(C_1 \Rightarrow \neg Y) \geq ms$ and $conf(C_1 \Rightarrow \neg Y) < mc$, in our algorithm, we may generate some Y' , $Y \subseteq Y'$ such that $supp(C_1 \Rightarrow \neg Y') \geq ms$ and $conf(C_1 \Rightarrow \neg Y') \geq mc$. This process costs a little more time when mc increases, however, the minimal Number/Time of VAR is still advantageous over the maximal of Apriori.

5 Conclusion

Negative association rules are considered useful in many real world applications. This paper has proposed a notion of valid association rules and developed an effective approach, along with the corresponding algorithm, to mining all positive and negative ones in a sound and complete manner. Several rule properties have been investigated and incorporated into the mining process as pruning strategies in order to gain algorithmic efficiency. The main advantage of the proposed approach over others could be characterized in terms of meaningfulness in rule validity and completeness in rule generation.

Acknowledgements

The work was partly supported by the National Natural Science Foundation of China (70621061/70231010) and Research Center for Contemporary Management at Tsinghua University.

References

1. Piatetsky-Shapiro, G.: Discovery, Analysis and Presentation of Strong Rules. In: Knowledge Discovery in Databases, pp. 229–248. AAAI/MIT, Menlo Park (1991)
2. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In: Proc. ACM-SIGMOD Intl. Conf. on Management of Data, pp. 207–216 (1993)

3. Srikant, R., Agrawal, R.: Fast Algorithms for Mining Association Rules. In: Proc. VLDB Conference, pp. 487–499 (1994)
4. Brin, S., Motwani, R., Silverstein, C.: Beyond Market Baskets: Generalizing Association Rules to Correlations. In: Proc. ACM SIGMOD on Management of Data, pp. 265–276 (1997)
5. Savasere, A., Omiecinski, E., Navathe, S.: Mining for Strong Negative Associations in a Large Database of Customer Transactions. In: Proc. Intl. Conf. on Data Engineering, pp. 494–502 (1998)
6. Aggarwal, C.C., Yu, P.S.: A New Framework for Itemset Generation. In: Proc. ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems, pp. 18–24 (1998)
7. Wei, Q., Chen, G.: Association Rules with Opposite Items in Large Categorical Database. In: Proc. Intl. Conf. on Flexible Query Answering Systems, pp. 507–514 (2000)
8. Yuan, X., Buckles, B.P., Yuan, Z., Zhang, J.: Mining Negative Association Rules. In: Proc. Seventh Intl. Symposium on Computers and Communication, Italy, pp. 623–629 (2002)
9. Antonie, M.L., Zaïane, O.R.: Mining Positive and Negative Association Rules: an Approach for Confined Rules. In: Proc. Intl. Conf. on Principles and Practice of Knowledge Discovery in Databases, pp. 27–38 (2004)
10. Daly, O., Taniar, D.: Exception Rules Mining Based On Negative Association Rules. In: Laganá, A., Gavrilova, M.L., Kumar, V., Mun, Y., Tan, C.J.K., Gervasi, O. (eds.) ICCSA 2004. LNCS, vol. 3046, pp. 543–552. Springer, Heidelberg (2004)
11. Thiruvady, D.R., Webb, G.I.: Mining Negative Association Rules Using GRD. In: Proc. Pacific-Asia Conf. on Advances in Knowledge Discovery and Data Mining, pp. 161–165 (2004)
12. Wu, X., Zhang, C., Zhang, S.: Efficient Mining of Both Positive and Negative Association Rules. *ACM Trans. on Information Systems* 22(3), 381–405 (2004)
13. Cornelis, C., Yan, P., Zhang, X., Chen, G.: Mining Positive and Negative Association Rules from Large Databases. In: Wang, Y., Cheung, Y.-m., Liu, H. (eds.) CIS 2006. LNCS (LNAI), vol. 4456, Springer, Heidelberg (2007)

Designing a System for a Process Parameter Determined through Modified PSO and Fuzzy Neural Network

Jui-Tsung Wong¹, Kuei-Hsien Chen^{2,*}, and Chwen-Tzeng Su³

¹ Department of International Business, Shih Chien University Kaohsiung Campus, Neimen Shiang, Kaohsiung, No. 200, University Rd, Neimen Shiang, Kaohsiung, Taiwan, 845
g9221802@yuntech.edu.tw

² Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, 123 University Rd. Sec. 3 Douliou, Yunlin 640, Taiwan, R.O.C.,
Tel.: +886 5 5342601 x5194; Fax: +886 5 5312073
g9321803@yuntech.edu.tw

³ Department of Industrial Engineering and Management, National Yunlin University of Science and Technology, 123 University Rd. Sec. 3 Douliou, Yunlin 640, Taiwan, R.O.C.
suct@yuntech.edu.tw

Abstract. In the manufacturing industry, the key to retaining a competitive advantage lies in increased yield and reduced a number of reworks. Determining the optimal parameters for the process so that the quality characteristics can meet the target is an important strategy. Traditional statistical techniques such as response surface methodology and analysis of variance, whose basic assumptions must be met, are generally used in this regard. In recent years, artificial intelligence has reached a sufficient level of maturity and is extensively being used in various domains. This paper proposes a system based on the modified particle swarm optimizer (PSO) and the adaptive network-based fuzzy inference system (ANFIS) to determine the process parameters. A perturbed strategy is incorporated into the modified PSO. The application of this system is then demonstrated with the determining of parameters in the wire bonding process in the IC packaging industry. Moreover, the performance of the modified PSO is evaluated with testing functions. The results show that the modified PSO yielded a superior performance to traditional PSO. In the optimization of the process parameter, the modified PSO is able to find the optimal solution in the ANFIS model.

Keyword: wire bonding process, determination of process parameters, modified particle swarm optimizer, adaptive network-based fuzzy inference system.

1 Introduction

Businesses in the high-tech industry are faced with increasing competition. Given the high cost of raw materials, the key to survival in this industry lies in increased yield. As far as optimizing the process parameters is concerned, the engineers' priority has become using efficient and convenient methods to adjust controllable parameters in

* Corresponding author.

order to bring quality characteristics close to the desired target. Traditional statistical techniques such as analysis of variance and response surface methodology are generally used to determine the process parameters ([1], [2], [3], [4], [5], [6], [7]). However, if such techniques are to be applicable, their basic assumptions must be met. This paper proposes an artificial intelligence-based system in determining the process parameters.

The particle swarm optimizer (PSO) was an evolutionary computation first proposed by [8]. Like bird flocking, fish schooling, and swarm theory, PSO was inspired by the social behavior of animals. In executing the PSO, every individual particle moves in accordance with a randomized velocity in the flying experience of itself and others in the same swarm. Unlike traditional genetic algorithms, PSO possesses memory, so the optimal solution for the swarm in execution will be memorized. Individual particles will also memorize the personal best solution. The velocity of every particle will be updated accordingly. PSO, when used in the optimization of process parameters, is deemed a very useful approach ([9], [10]). Trelea [11] analyzed how the selection of parameters in PSO affected convergence and the performance of finding the solution through dynamic system theory.

The adaptive network-based fuzzy inference system (ANFIS) is basically a fuzzy neural network. First proposed by [12], ANFIS systematically generate fuzzy rules from the training data of input and output. This is a supervised neural network based on fuzzy theory, which has been in extensive use in the prediction and control domain. Cai et al. [13] used ANFIS to predict the state-of-charge of high power in a rechargeable

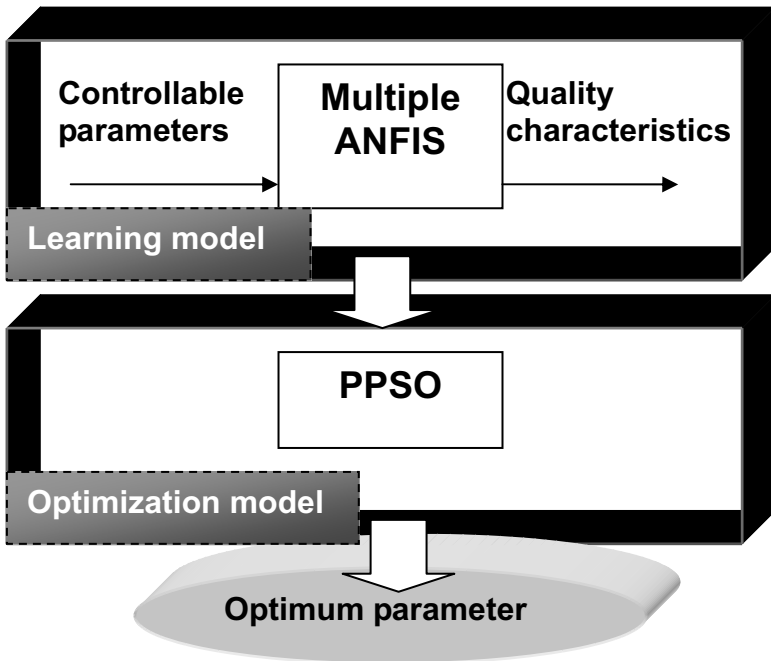


Fig. 1. The framework of the proposed method

battery, whose performance was then compared with the back-propagation artificial neural network (BPN). In this simple testing, it is found that ANFIS outperformed BPN. Mar and Lin [14] used ANFIS to formulate rules in controlling the speed of cars to avoid collisions. ANFIS were used in other areas by [15], [16], [17], [18].

This paper, therefore, proposes a system for determining the process parameters by using ANFIS as the simulation model. A modified PSO is then used to determine the optimal process parameters. A perturbed strategy is also incorporated into the modified PSO to better avoid caving into the local optimum. This method is therefore called PPSO. The application of the method is then demonstrated and tested with the finding of process parameters for the second bonding process, which is an important step in the IC packaging industry. Figure 1 is the framework of the proposed method.

2 The Architecture of the Proposed Approach

2.1 The Integrated System

Figure 1 shows that the learning process of the proposed method is multiple ANFIS, its trained input being the controllable parameters of the process, and the output being the quality characteristics. Once the input/output is established by multiple ANFIS, PPSO algorithm is then used to find the optimal process parameters.

2.2 PPSO Algorithm

The procedure for the algorithm is shown in figure 2. Since the quality characteristic of this example is the larger-the-better case, PPSO is basically a maximum problem. The PPSO algorithm, unlike traditional PSO algorithms, includes the perturbed strategy. Its implementation is as follows:

Step 1. (Initial solution): Randomly generate L initial solutions.

Step 2. (Update the velocity): The calculation of every particle in the PPSO algorithm is moved by two sets of information, which are the current optimal solution and the optimal solution for individual particles. PPSO algorithm moves the whole group of particles toward the optimal solution through the global optimum ($gbest$). Individual particles perform the calculation in accordance with their personal memory. The particles update their velocity as follows:

$$v_{ij}^{k+1} = wv_{ij}^k + c_1 rand_1 (pbest_{ij} - s_{ij}^k) + c_2 rand_2 (gbest_j - s_{ij}^k) \tag{1}$$

where v_{ij}^k is the velocity of particle i at controllable parameter j at iteration k . w is the inertia weight within the range $[0, 1]$. c_1 and c_2 are two constants; $rand_1$ and $rand_2$ represent the uniformly random value between 0 to 1. s_{ij}^k is the position (solution) of particle i at controllable parameter j at iteration k . $pbest_{ij}$ is the value of the optimal solution of particle i at controllable parameter j . $gbest_j$ is the value of the global optimum at controllable parameter j . The initial velocity is generated randomly.

Step 3. (Update position): The update of the solution for every particle is as follows:

$$s_{ij}^{k+1} = s_{ij}^k + v_{ij}^{k+1} \tag{2}$$

Step 4. (Obtain quality characteristic): The quality characteristics are obtained through a model learned by ANFIS.

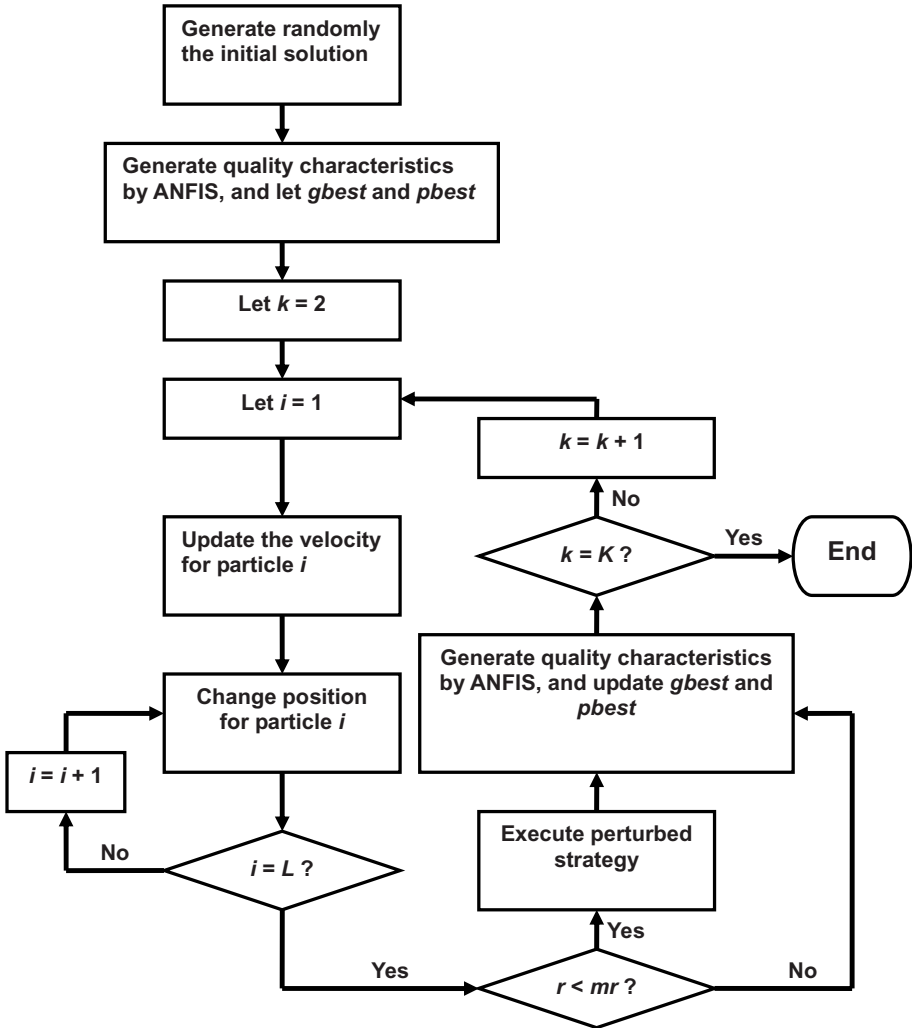


Fig. 2. Flow chart of the PPSO

Step 5. (Perturbed strategy): This is the key step in the modified PSO. When a random value between [0, 1] $r <$ the perturbed rate pr , the perturbed strategy is executed. Its procedure is as follows:

```

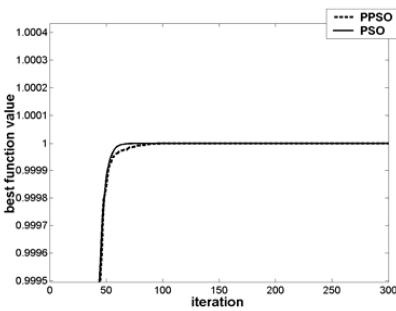
begin
  if  $r < pr$ 
    for  $i = 1$  to  $L$ 
       $\{v_{ij}^k \mid j \in E_i^k\} \leftarrow$  a random value between  $[a, b]$ ;
    end for
  end if
end

```

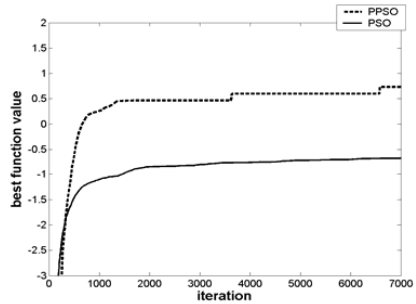
where a and b are adjustable parameters. A controllable parameter of particle i , E_i^k , is chosen randomly at iteration k .

3 Proposed Algorithm Test

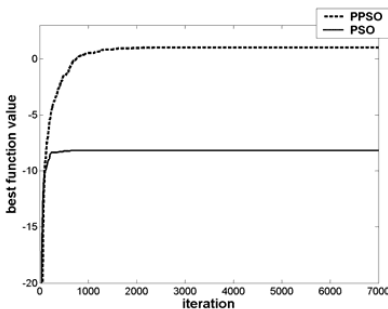
This section compares the performance of the PPSO proposed by this paper and the PSO through the testing functions, the objective being maximum equations (3)-(6). The algorithm is run on every function 100 times; the initial solution is generated with the uniform random number between $[-5, 5]$. The parameter of the algorithm is $w = 0.7$, the number of particles = 200, and the number of iterations = 7000. Moreover, parameters $c_1 = 1.2$ and $c_2 = 1.2$ are used in equations (3)-(5). Parameters $c_1 = 1.2$ and $c_2 = 0.7$ are used in equation (6).



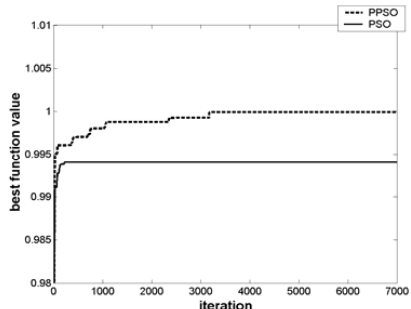
(a)



(b)



(c)



(d)

Fig. 3. Convergence process of PPSO and PSO in the (a) f_1 , (b) f_2 , (c) f_3 , and (d) f_4

$$f_1 = 1 - \sum_{i=1}^N z_i^2 \tag{3}$$

$$f_2 = 1 - \sum_{i=1}^{N-1} \left[100 \times (z_i^2 - z_{i+1})^2 + (1 - z_i)^2 \right] \tag{4}$$

$$f_3 = 1 - 10 \times N + \sum_{i=1}^N \left[z_i^2 - 10 \cos(2\pi z_i) \right] \tag{5}$$

Table 1. Result of the algorithm test

1.	Tested function Equation (3)				Optimal solution $z_i=0$ $f_1(z_i) = 1$			
	PSO				PPSO			
	Min.	Avg.	Max.	∇^a	Min.	Avg.	Max.	∇
$N = 3$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 6$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 9$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 12$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
2.	Tested function Equation (4)				Optimal solution $z_i=1$ $f_2(z_i) = 1$			
	PSO				PPSO			
	Min.	Avg.	Max.	∇	Min.	Avg.	Max.	∇
$N = 3$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 6$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 9$	-2.986	0.798	1.000	92	1.000	1.000	1.000	100
$N = 12$	-3.320	-0.589	1.000	16	-2.987	0.920	1.000	98
3.	Tested function Equation (5)				Optimal solution $z_i=0$ $f_3(z_i) = 1$			
	PSO				PPSO			
	Min.	Avg.	Max.	∇	Min.	Avg.	Max.	∇
$N = 3$	0.005	0.970	1.000	97	1.000	1.000	1.000	100
$N = 6$	-4.970	-0.264	1.000	28	1.000	1.000	1.000	100
$N = 9$	-10.940	-3.447	1.000	3	1.000	1.000	1.000	100
$N = 12$	-17.904	-9.029	-0.990	0	1.000	1.000	1.000	100
4.	Tested function Equation (6)				Optimal solution $z_i=0$ $f_4(z_i) = 1$			
	PSO				PPSO			
	Min.	Avg.	Max.	∇	Min.	Avg.	Max.	∇
$N = 3$	1.000	1.000	1.000	100	1.000	1.000	1.000	100
$N = 6$	0.988	0.999	1.000	85	1.000	1.000	1.000	100
$N = 9$	0.988	0.998	1.000	73	1.000	1.000	1.000	100
$N = 12$	0.975	0.994	1.000	48	0.993	1.000	1.000	99

a: ∇ is the number of optimal solutions obtained

$$f_4 = 1 - \sum_{i=1}^N \left(\frac{z_i^2}{4000} \right) - \prod_{i=1}^N \cos \left(\frac{z_i}{\sqrt{i}} \right) \tag{6}$$

Table 1 shows that in all 100 runs, except for when $N = 12$, PPSO is able to find the global optimum for every testing functions, and yielded superior performance to the PSO. Figure 3 shows the improvement of the average best solution for equations (3)-(6). The results indicate that the perturbed strategy can effectively prevent this algorithm from caving into the local optimum.

4 Example Application of the Approach

This section demonstrates the applicability of the proposed method through the optimization of parameters for the second bonding process.

4.1 The Wire Bonding Process

In semi-conductor manufacturing, the wire bonding process is the key technology in the packaging industry. The goal is to connect the chip with the inner lead in the lead frame with a fine gold wire so that the electronic signals of the IC chips can be transmitted. The bonding point should be firmly secured, or the IC chip will not function. Therefore, the wire bonding process plays a pivotal role in the whole IC packaging industry, the key point being the finding of the optimal parameter for the wire bonding process. During the bonding process, the tip of the gold wire is first molten into a small ball, and then pressed onto the first bonding point. The gold wire is then placed in the designated path, and pressed onto the second bonding point, as shown in figure 4.

The data used in this paper is the actual process output [19]. The main controllable parameters in the second bonding process include: bonding force, bonding time, the intensity of ultrasonic power. Its quality characteristic is wire pull.

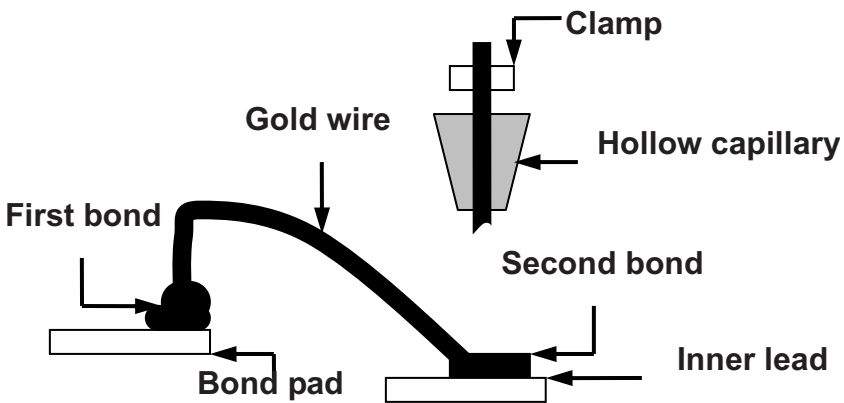


Fig. 4. A typical wire bonding process

4.2 The Learning Result of ANFIS

In order to enhance the learning of ANFIS, the controllable parameters and quality characteristics are first pre-processed. This means normalizing the original data so as to avoid overlooking the importance of variables of a smaller range if the range of the variables in the trained data became wider. This will prevent the entire network learning being dominated by variables with a greater range, and also affect the entire learning result of ANFIS. Therefore, this experiment normalized the quality characteristics and the parameters between [0, 1].

Table 2. The learning result of ANFIS

Number of membership	Membership function					
	Triangle		Trapezoid		Bell-shaped	
	Training error	Testing error	Training error	Testing error	Training error	Testing error
3-3-3	0.083153	0.15436	0.083241	0.16392	0.083241	0.12689
4-6-4	0.038244	0.21061	0.061158	0.09495	0.049367	0.11436
6-4-6	0.037719	0.34165	0.061158	0.44553	0.049267	0.42281

The parameters of ANFIS learning include: membership function and the number of memberships among the variables. In terms of membership function, triangle, trapezoid and bell-shape are chosen for testing. There are also three sets of numbers chosen as the number of membership among the variables. Table 2 shows that when the membership function is a trapezoid and the numbers of memberships are [4-6-4], the root mean square error (RMSE) is the smallest. Therefore, this paper adopts this model. Figure 5 shows the response output for quality characteristic in ANFIS.

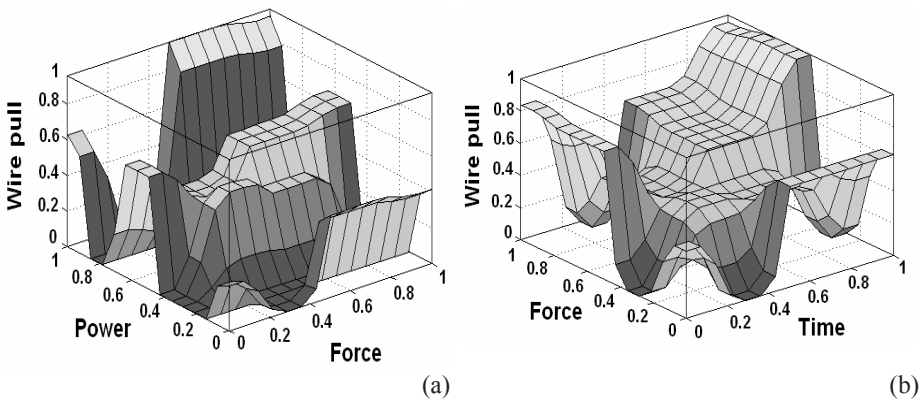


Fig. 5. The response surface showing the effect of (a) bonding force and the intensity of ultrasonic power, (b) bonding force and bonding time on the wire pull

4.3 The Proposed Algorithm Implementation

In the packaging industry, the larger the quality characteristic of the second bonding process, the better. This paper uses the PPSO algorithm to find the largest wire pull for the gold wire.

Table 3 shows the result of PSO and PPSO algorithm in 30 runs at the second bonding process. The result indicates that the PPSO algorithm is able to generate the near-optimal parameters for the manufacturing process under the ANFIS-trained model in all 30 runs.

Table 3. Result of the optimization algorithm for the example

Algorithm	Max.	Min.	Avg.	Standard deviation
PSO	1.0000	0.9583	0.9917	0.0167
PPSO	1.0000	1.0000	1.0000	0.0000

5 Conclusion

Manufacturing in the high-tech industry is a complex undertaking. Designing an efficient and easy decision-making system to determine the parameters for processing is, therefore, of paramount importance. Traditional statistical techniques may be restrained by basic assumptions. Therefore, this paper proposes a system for determining the optimal parameters for the process based on artificial intelligence.

This paper uses an adaptive network-based fuzzy inference system to construct the simulation model for the process. A modified particle swarm optimizer algorithm is then used to determine the optimal parameter for the process. This paper then tested the performance of the PPSO algorithm with testing functions, the result of which shows that the perturbed strategy used by PPSO is effective at avoiding caving into the local optimum. This paper further demonstrated the application of the proposed approach with the second bonding process in the IC packaging industry. As far as the optimal parameter in this example is concerned, figure 5 shows that a local optimum exists in the relationship model of controllable parameters and quality characteristics. The result of the testing also shows that this PPSO algorithm is able to find the global optimum under the ANFIS model.

References

1. Kalil, S.J., Mauger, F., Rodrigues, M.I.: Response surface analysis and simulation as a tool for bioprocess design and optimization. *Process Biochem* 35(6), 539–550 (2000)
2. Kincl, M., Turk, S., Vrečer, F.: Application of experimental design methodology in development and optimization of drug release method. *Int. J. Pharm.* 291(1-2), 39–49 (2005)

3. Pech-Canul, M.I., Katz, R.N., Makhlof, M.M.: Optimum conditions for pressureless infiltration of SiCp performs by aluminum alloys. *J. Mater. Process. Tech.* 108(1), 68–77 (2000)
4. Kwak, J.S.: Application of Taguchi and response surface methodologies for geometric error in surface grinding process. *Int. J. Mach. Tool. Manu.* 45(3), 327–334 (2005)
5. Hewidy, M.S., El-Taweel, T.A., El-Safty, M.F.: Modelling the machining parameters of wire electrical discharge machining of Inconel 601 using RSM. *J. Mater. Process. Tech.* 169(2), 328–336 (2005)
6. Gao, Y.L., Jiang, H.H.: Optimization of process conditions to inactivate *Bacillus subtilis* by high hydrostatic pressure and mild heat using response surface methodology. *Biochem. Eng. J.* 24(1), 43–48 (2005)
7. Kansal, H.K., Singh, S., Kumar, P.: Parametric optimization of powder mixed electrical discharge machining by response surface methodology. *J. Mater. Process. Tech.* 169(3), 427–436 (2005)
8. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948. IEEE Press, New York (1995)
9. Costa Jr., E.F., Lage, P.L.C., Biscaia Jr., E.C.: On the numerical solution and optimization of styrene polymerization in tubular reactors. *Comput. Chem. Eng.* 27(11), 1591–1604 (2003)
10. Ourique, C.O., Biscaia Jr., E.C., Pinto, J.C.: The use of particle swarm optimization for dynamical analysis in chemical processes. *Comput. Chem. Eng.* 26(12), 1783–1793 (2002)
11. Trelea, I.C.: The particle swarm optimization algorithm: convergence analysis and parameter selection. *Inform. Process. Lett.* 85(6), 317–325 (2003)
12. Jang, J.-S.R.: ANFIS: adaptive-network-based fuzzy inference system. *IEEE T. Syst. Man. Cy.* 23(3), 665–685 (1993)
13. Cai, C.H., Du, D., Liu, Z.Y.: Battery State-of-Charge (SOC) estimation using adaptive neuro-fuzzy inference system (ANFIS). In: *The IEEE International Conference on Fuzzy Systems*, vol. 2, pp. 1068–1073. IEEE Press, New York (2003)
14. Mar, J., Lin, F.J.: An ANFIS controller for the car-following collision prevention system. *IEEE T. Veh. Technol.* 50(4), 1106–1113 (2001)
15. Ho, S.Y., Lee, K.C., Chen, S.S., Ho, S.J.: Accurate modeling and prediction of surface roughness by computer vision in turning operations using an adaptive neuro-fuzzy inference system. *Int. J. Mach. Tool. Manu.* 42(13), 1441–1446 (2002)
16. Lo, S.P.: An adaptive-network based fuzzy inference system for prediction of workpiece surface roughness in end milling. *J. Mater. Process. Tech.* 142(3), 665–675 (2003)
17. Yeh, F.H., Tsay, H.S., Liang, S.H.: Applied CAD and ANFIS to the Chinese Braille display optimization. *Displays* 24(4-5), 213–222 (2003)
18. Ryoo, J., Dragojlovic, Z., Kaminski, D.A.: Control of convergence in a computational fluid dynamics simulation using ANFIS. *IEEE T. Fuzzy Syst.* 13(1), 42–47 (2005)
19. Huang, K.M.: Applying Taguchi's method to study the optimum parameter for the 50mm fine pitch on the pad in the wire bonding process of the IC assembly. Master Thesis of Engineering, Yunlin University of Science & Technology (2002)

Data-Aware Clustering Hierarchy for Wireless Sensor Networks*

Xiaochen Wu, Peng Wang, Wei Wang, and Baile Shi

Fudan University, Shanghai, China
{052021120, pengwang5, weiwang1, blshi}@fudan.edu.cn

Abstract. In recent years, the wireless sensor network (WSN) is employed a wide range of applications. But existing communication protocols for WSN ignore the characteristics of collected data and set routes only according to the mutual distance and residual energy of sensors. In this paper we propose a Data-Aware Clustering Hierarchy (DACH), which organizes the sensors based on both distance information and data distribution in the network. Furthermore, we also present a multi-granularity query processing method based on DACH, which can estimate the query result more efficiently. Our empirical study shows that DACH has higher energy efficiency than Low-Energy Adaptive Clustering Hierarchy (LEACH), and the multi-granularity query processing method based on DACH brings more accurate results than a random access system using same cost of energy.

Keywords: wireless sensor network, communication protocol, data distribution, multi-granularity query.

1 Introduction

In recent years, the wireless sensor network (WSN) [1, 2] is employed a wide range of applications in military security, environmental monitoring, and many other fields. Except some high accuracy required applications (for example, applications in military), most applications of WSN are cost driven. Users want to acquire more information with less energy cost. In order to minimize the energy consumption and maximize the life span of the network, clustering techniques based on data fusion [3] such as LEACH [4], LEACH-C [5], BCDCP [6] etc. have been proposed.

All above cluster-based protocols try to find the shorter routes for data transmission and spread energy consumption around all the sensors more evenly. The sensors are organized into clusters according to the mutual distance and residual energy of them. In such scheme, the process of data collection is independent with the characteristics of collected data. But in many applications of WSN, data collected from some adjacent sensors are similar. The sensing field can be divided into regions with different characteristics. For example, in a building site, the temperature data col-

* This research is supported in part by the National High-Tech Research and Development Plan of China under Grant 2006AA01Z234 and the National Basic Research Program of China under grant 2005CB321905.

lected from indoor sensors and outdoor sensors may be similar respectively. However, during the running of the wireless sensor network, we can estimate data distribution in the network using some data mining methods. And based on this information, the clusters can be organized not only according to the mutual distance, but also the characteristics of collected data. We can build the clusters so that data collected from sensors in a same cluster are similar. This method can compress data volume more efficiently after data fusion and prolong the network's life span further. Users can acquire more information from the network with less energy cost.

In this paper, we propose a Data-Aware Clustering Hierarchy (DACH), which is not only energy-efficient, but also capable of obtaining data distribution from the network. In DACH, data distribution is estimated by a data mining process based on collected data and the sensors are distributed into a clustering hierarchy according to the discriminations between the collected data. Furthermore, we introduce a multi-granularity query processing method based on DACH to estimate the query results using a few sensors' data instead of all of them.

Our Contributions

1. We propose a data mining method to estimate the data distribution in wireless sensor network, and based on it we introduce a new clustering structure and also a new communication protocol for WSN.
2. We propose a multi-granularity query processing method based on DACH to estimate the query results using a few sensors' data instead of all of them.

2 Data-Aware Clustering Hierarchy

2.1 Data Distribution in Wireless Sensor Network

In many applications of wireless sensor network, the sensing field can be divided into a series of regions with different characteristics. In the example mentioned in section 1, the whole building site contains the indoor regions and the outdoor regions, and the space inside a building still can be divided into different building stories, rooms and areas. It is possible that the temperature is very different between some regions (for example, the indoor regions and the outdoor regions). And on the other hand, it is similar in some regions (for example, the areas in a same room). We refer to this property as the "Regional Property".

As the regional property of the sensing field, data collected from sensors deployed in the field also have the regional property. In above example, data collected from outdoor sensors may be very different from data collected from indoor sensors, as the difference of the physical conditions between outdoor and indoor regions. And in the mean time, data collected from sensors in a same room may be similar.

Based on the regional property, we can estimate the data distribution by the discrimination between data collected from different sensors or sensor sets (discrimination of sensors or discrimination of sensor sets for short). We can consider data collected from each sensor as a time series and define the discrimination of sensors by the discrimination of the time series. In this paper, the time series, denoted by TS, with length n is: $TS=TS_1, TS_2, \dots, TS_n$.

Def 1: Discrimination of sensors: Data collected from a sensor in a time interval compose a time series. We use Euclidean distance to define the discrimination between sensors. Furthermore, the value of the discrimination is divided by $n^{1/2}$ to eliminate the influence of the length of time series:

$$disc(i, j) = disc(TS^i, TS^j) = \sqrt{\frac{1}{n} \sum_{k=1}^n (TS_k^i - TS_k^j)^2} / n \quad (1)$$

Moreover, we can estimate the discrimination between two sensor sets by the discrimination between centroids of corresponding time series of sensors in each set.

2.2 Data-Aware Clustering Hierarchy

As an example to illustrate our motivation, we use a sensor network to monitor the temperatures of a building site as shown in figure 1a. The gray region indicates outdoor regions, and the white part indicates indoor ones. Using traditional clustering methods, the clustering structure in a certain round may be organized as Figure 1b [4]. It shows that there are 17 nodes in the cluster A, nine of which are in the gray region and other eight are in the white one. Data collected from sensors in this cluster may be very different between each other.

An ideal clustering structure is shown in figure 1c. It still contains 5 clusters. The difference from figure 1b is that each cluster represents a section of outdoor or indoor regions. So this structure is more data-aware. Data collected from sensors in same clusters are similar. Based on this property, we can acquire more information from WSN using less energy. We can compress data volume more efficiently after data fusion and prolong the network's life span further. Moreover, in some applications of approximate queries, we can only query a few sensors' data instead of all of them.

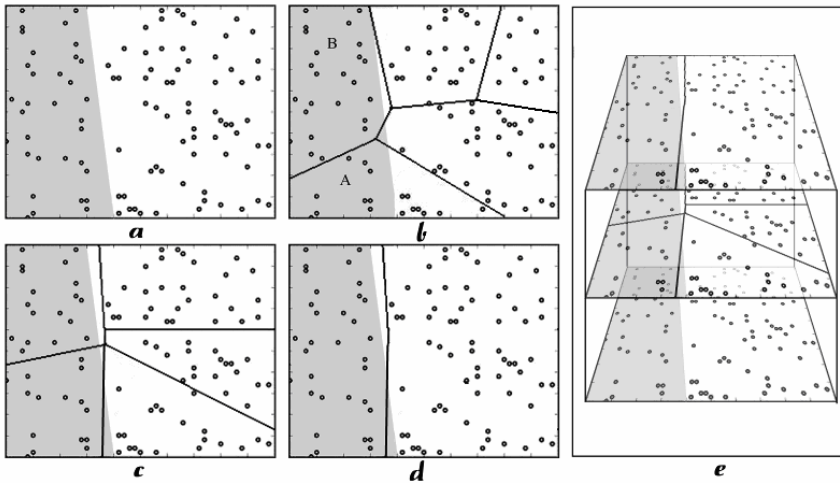


Fig. 1. (a) A region where we deployed wireless sensors to monitor temperature; (b) The clustering structure in a certain round using traditional cluster method; (c) The clustering structure based on data distribution; (d) The clustering structure on the higher level; (e) A data-aware clustering hierarchy.

In the figure 1d, similar clusters in figure 1c are merged to larger clusters respectively. The left cluster contains all the sensors in the outdoor regions, and the right one contains sensors in the indoor regions. The cluster structures in these two figures constitute a clustering hierarchy showed in figure 1e.

2.3 Algorithm for Building Data-Aware Clustering Hierarchy

In this section we propose an algorithm for building the data-aware clustering hierarchy based on topological structure of network and data distribution. To facilitate our discussion, we first give some general definitions:

Def.2 Relay: The indirectly transmission from node A to node B through node C is called Relay. The node C is called Relay node.

Def.3 Relay Region: Given a node s and a relay node r , the relay region of s with respect to r is defined as follows:

$$R_{\alpha,c}(s, r) = \{x \mid \|sx\|^\alpha > \|sr\|^\alpha + \|rx\|^\alpha + c\} . \tag{2}$$

where $\|x\|$ denotes the distance between node x and node y , α and c are two constant parameters which equal to 4 and E_{TX}/ϵ_{amp} respectively according to the above radio model. Obviously, the nodes in the relay region of s with respect to r can be reached with least energy by relaying r .

Def.4 Neighbor: The node not in any relay region of s is called neighbor of s . Formally, we define it as follows:

$$N_{\alpha,c}(s) = \{u \mid \forall r, u \notin R_{\alpha,c}(s, r)\} . \tag{3}$$

Furthermore, two sensor sets A and B are called neighborhood sets if there are two neighborhood sensors a and b , where sensor a is in the set A and b is in set B.

Before building a k -level clustering hierarchy, we define a series of thresholds, $\delta_0, \delta_1, \dots, \delta_{k-2}$ satisfying $\delta_{k-2} > \delta_{k-1} > \dots > \delta_1 > \delta_0 = 0$. The thresholds can be specified according to the sensing scenarios. For example, in a temperature monitoring system, the thresholds can be specified as 0, 0.5, 1, 2, 4 (°C).

Assume that the base station save data collected from every sensors as a time series. In the lowest level (level 0), we initialize a set for each sensor in the network and compute the discrimination between each pair of sets using following equation:

$$d(S_A, S_B) = \begin{cases} \infty & \text{A and B are not neighbors} \\ disc(S_A, S_B) & \text{A and B are neighbors} \end{cases} . \tag{4}$$

Then, we build clusters on higher level in a bottom-up way by following steps:

1. Find the pair of sets with minimum mutual discrimination d_{min} ;
2. If d_{min} is larger than the threshold δ_i , output current sets as the clustering structure of level- i ;
3. Combine these two sets into a new set, compute the centroid of all the time series in the new set and update its discrimination with other sets
4. Repeat steps 1-3 until the minimum discrimination is larger than the maximum threshold δ_{k-2}

5. Combine all the remaining sets into an only set as the cluster on the highest level, which contains all the sensors in the network.

After that, we obtained a clustering hierarchy. Every set on every level contains a series of subset on the lower level which is called as “descendent sets”. And in the mean time, it is also a part of a set on the higher level which is called as “ancestor set”.

3 A Communication Protocol Based on DACH

Based on the DACH proposed in last section, we introduce a novel communication protocol for WSN. We also call this communication protocol as DACH for short in the context of not leading to any ambiguity. DACH operates in three phases: initialization, setup and data transmission.

Initialization Phase: When the sensors are deployed on the field or the topological structure of the network is changed (e.g. when the energy of some sensors is exhausted or the properties of the circumstance are changed), the network enters the initialization phase.

During the initialization phase, the base station receives data from all sensors in a given period and generates a time series for every sensor. Based on these time series, the base station computes the discriminations between each pair of sensors and builds the clustering hierarchy using the method proposed in section 2.3.

Setup Phase: When the system is initialized, the network enters the setup phase. The main task in this phase is to generate routing path and schedule for each node. The base station receives information of the current energy status from all the nodes in the network. Based on the feedback and the clustering hierarchy, base station generates the routing path in a bottom-up way.

For each cluster, the algorithm selects one node as the cluster-head. For level 0, since each cluster only contains one node, each node is a cluster-head. For each cluster C on level- i ($i > 0$), the cluster-head must satisfy following two conditions:

- 1) It is a cluster-head of one of its children clusters;
- 2) Its residual energy is the highest among cluster-heads of all its children clusters.

Now the routing path of each sensor node can be obtained easily. It first transmits data to the corresponding cluster-head which subsequently transmits data to the cluster-head of its parent cluster. This process continues until the data is transmitted to the only cluster-head on the highest level. And the data is sent to the base station finally.

The cluster-head of a level- i cluster and all the cluster-heads in its children cluster compose a sub-network. To improve the energy efficiency, in this sub-network the sensors transmit data using a multi-hop method.

Data Transmission Phase: The data transmission phase consists of three major activities: data gathering, data aggregation and data routing. Using the scheme described above, each sensor node transmits sensed data to its corresponding cluster-head. For each cluster-head, once receiving data from all contained nodes, it aggregates the collected data into a data of smaller volume and sends it to cluster-head on higher level. The cluster-head on highest level transmits the aggregated data to the base station.

For spreading energy consumption between sensors more evenly, after a period of data transmission phase, the network will enter the setup phase again, reselect the cluster-head and regenerate the routing path for every sensor.

4 Multi-granularity Query Processing Method Based on DACH

In most performance-driven application, WSN may have less stringent performance requirements and can be implemented at much lower cost. J. Frolik proposed random access techniques to help facilitate such requirements [9]. In [9] the quality of service (QoS) measures application reliability with a goal of energy efficiency. According to the user-defined quality of service, the random access system selects a proportion of sensors for data gathering. These sensors are called “active sensors”. But as the active sensors are selected randomly, the data collected by a random access system may not be able to simulate the whole data set appropriately.

In this section, we discuss our multi-granularity query processing method based on DACH for cost-driven applications. Since in each cluster, the data of all nodes are similar, we can execute the query on the cluster-heads on certain level instead of all sensor nodes.

The multi-granularity queries have the following basic structure:

```
SELECT expr1, expr2...
FROM network
WHERE pred1 [and|or] pred2
LEVEL ON levelNum
```

The SELECT, FROM and WHERE clauses are defined as the standard SQL. The LEVEL ON clause specifies the level of the query. From the definition of clustering hierarchy, it can be seen that the data of all nodes are similar in a certain cluster, and the lower the level, the number of the cluster-heads is larger and the data are more similar. In other words, the user can specify the number of active sensors and the approximate estimating error using the LEVEL ON clause.

Assume the levelNum be k , the cluster-heads on level k estimate the data of the sensors in the same cluster based on its own data. That is, the query is processed on the cluster-heads on level- k .

Because of the similarity of data collected from sensors in the same cluster, our method can estimate the query result accurately. Data gathered by the random access system may omit data of sensors in some small and special region because the probability of sampling data in it is relatively low. On the contrary, our method will not ignore these regions.

5 Performance Evaluations

To test the performance of DACH and the multi-granularity query processing method, we simulate an environment temperature monitoring system. Using this simulated system, we compare the energy efficiency of DACH with LEACH and the estimating accuracy of the multi-granularity query processing method with the random access system presented in [9]. All the algorithms are implemented in JAVA. The test

environment is PC with AMD Athlon processors 3000+, 1GB of RAM, and running Windows XP Professional.

We use the network model and radio model discussed in [4, 5, 6] and simulate the temperature information using a bitmap. The system generates the coordinates of the sensors randomly and set the parameters of temperature based on RGB colors of the corresponding pixels. The value of $R/10$ represents the average temperature in one day. The value of $G/10$ represents the maximum temperature in one day. And the value of $B/10$ represents the time of the maximum temperature. We simulate the intraday temperature by sinusoid. The temperature on point (x, y) at time t is denoted by:

$$T_{(x,y,t)} = R_{(x,y)} / 10 + (G_{(x,y)} - R_{(x,y)}) / 10 \cdot \sin((x - B_{(x,y)}) / 10 + 6) \cdot \pi / 12) . \quad (5)$$

We assume that the base station locates at point $(-20, -20)$. Each node is assigned an initial energy of 2J. The number of data frames transmitted for each round is set at 10; the data message size for all simulations is fixed at 500 bytes, of which 25 bytes is the length of the packet header.

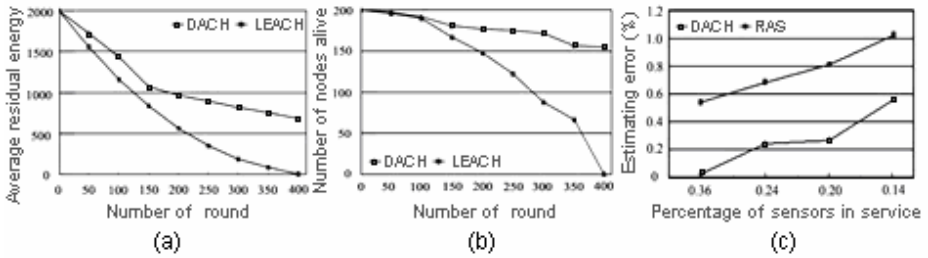


Fig. 2. (a) The average residual energy of sensors of DACH and LEACH at different number of operation rounds. (b) The number of nodes that remain alive at different number of rounds. (c) The estimating errors of the multi-granularity query processing method and that of the random access system corresponding to percentages of sensors in service.

In the first experiment we compare the energy efficiency of DACH and LEACH. We simulate a $50m \times 50m$ network with 200 sensors. Figure 2a shows the average residual energy of sensors of DACH and LEACH at different number of operation rounds. It can be seen that DACH has more desirable energy efficiency than LEACH.

Figure 2b shows the number of nodes that remain alive at different number of rounds. After 100 rounds, the sensors in the LEACH die more quickly than DACH. And when all sensors are dead in the LEACH, more than 150 sensors in the DACH remain alive. It indicates that DACH distributes the energy load among the sensors in the network more efficiently.

In the second experiment, we compare the accuracy of the query results between the multi-granularity query processing method based on DACH and the random access system. We simulate a $50m \times 50m$ network with 100 sensors and compute the average temperature in the network. The figure 2c shows that our method always

processes the query more accurately than the random access system. On average, our method reduces the estimating error by 0.5 degrees.

6 Conclusions

In this paper we proposed a data-aware clustering hierarchy for wireless sensors network (DACH) and a multi-granularity query processing method based on DACH. DACH divides the sensors into clusters according to the data distribution as well as mutual distance between sensors. Using the similarity of data collected from sensors in same clusters, the multi-granularity query processing method estimates the query result only by the data from cluster-heads on certain level. The simulation results show that DACH has much higher energy efficiency than LEACH. And the estimating errors of the multi-granularity query processing method are less than random access approach.

References

- [1] Akyildiz, I.F., Su, W., Sankarasubramaniam, Y., Cayirci, E.: Wireless Sensor Network: A Survey. In: IEEE Communications Magazine (August 2002)
- [2] Dong, M., Yung, K., Kaiser, W.: Low Power Signal Processing Architectures for Network Microsensors. In: Proceedings 1997 International Symposium on Low Power Electronics and Design, August 1997, pp. 173–177 (1997)
- [3] Hall, D.: Mathematical Techniques in Multisensor Data Fusion. Artech House, Boston (1992)
- [4] Heinzelman, W.R., Chandrakasan, A.P., Balakrishnan, H.: Energy Efficient Communication Protocol for Wireless Microsensor Networks. In: 33rd Hawaii International Conference on System Sciences (January 2000)
- [5] Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: An Application-Specific Protocol Architecture for Wireless Microsensor Networks. IEEE Transactions on wireless communications 1(4) (2002)
- [6] Murganathan, S.D., Ma, D.C.F., Bhasin, R.I., Fapojuwo, A.A.O.: A Centralized Energy-Efficient Routing Protocol for Wireless Sensor Networks. In: IEEE Radio Communications (March 2005)
- [7] Lindsey, S., Raghavendra, C., Sivalingam, K.M.: Data Gathering Algorithms in Sensor Networks using Energy Metrics. IEEE Transactions on Parallel and Distributed Systems 13(9) (September 2002)
- [8] Ding, P., Holliday, J., Celik, A.: Distributed Energy-Efficient Hierarchical Clustering for Wireless Sensor Networks. In: Prasanna, V.K., Iyengar, S.S., Spirakis, P.G., Welsh, M. (eds.) DCOSS 2005. LNCS, vol. 3560, Springer, Heidelberg (2005)
- [9] Frolik, J.: QoS Control for Random Access Wireless Sensor Networks. In: IEEE Wireless Communications and Networking Conference (2004)

A More Topologically Stable Locally Linear Embedding Algorithm Based on R*-Tree*

Tian Xia^{1,2}, Jintao Li¹, Yongdong Zhang¹, and Sheng Tang¹

¹ Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, China

² Graduate University of the Chinese Academy of Sciences, Beijing 100039, China
{txia, jtli, zhyd, ts}@ict.ac.cn

Abstract. Locally linear embedding is a popular manifold learning algorithm for nonlinear dimensionality reduction. However, the success of LLE depends greatly on an input parameter - neighborhood size, and it is still an open problem how to find the optimal value for it. This paper focuses on this parameter, proposes that it should be self-tuning according to local density not a uniform value for all the data as LLE does, and presents a new variant algorithm of LLE, which can effectively prune “short circuit” edges by performing spatial search on the R*-Tree built on the dataset. This pruning leads the original fixed neighborhood size to be a self-tuning value, thus makes our algorithm have more topologically stableness than LLE does. The experiments prove that our idea and method are correct.

Keywords: LLE, Manifold Learning, Nonlinear Dimensionality Reduction, R*-Tree, Neighborhood size.

1 Introduction

Dimensionality reduction is introduced as a way to overcome the curse of dimensionality when dealing with high-dimensional data and as a modeling tool for such data. There are usually two kinds of methods for dimensionality reduction: linear and nonlinear methods. Linear subspace methods are the most popular linear methods, including PCA (Principle Component Analysis), FLA (Fisher Linear Analysis), and ICA (Independent Component Analysis). However, we will only concentrate on the nonlinear methods in this paper because these methods pay more attention to nonlinearity in the dataset, and nonlinearity is more universal than linearity in the real world.

Recently, some manifold learning methods have been proposed to perform nonlinear dimensionality reduction, including LLE (Locally Linear Embedding) [1][8], ISOMAP

* This work was supported in part by the National Basic Research Program of China (973 Program, 2007CB311100), the National High Technology and Research Development Program of China (863 Program, 2007AA01Z416), the Knowledge Innovation Project of The Institute of Computing Technology, Chinese Academy of Sciences (20076031).

[2], and Eigenmaps [3]. All the above nonlinear algorithms share the same framework, consisting of three steps:

1. Constructing a K nearest neighborhood graph over the dataset.
2. Constructing a “normalized” matrix M.
3. Calculating spectral embedding based on the eigenvectors of M.

The neighborhood size K which has to be specified manually in the first step plays an important role in constructing a reasonable neighborhood graph for nonlinear dimensionality reduction. A large value of K tends to introduce “short circuit” edges [4] into the neighborhood graph, while a too small one may lead to an unconnected graph, both cases will distort the results of nonlinear dimensionality reduction. [5] and [6] tried to select the optimal value for K automatically based on a predefined cost function. Two main issues arise in this approach. First, it is an enumeration method in fact, and very time consuming. Second, the cost function is very difficult to define, and we do not think the cost functions used in [5] and [6] are reasonable and effective. We will give our explanation and proof in Section 3 and Section 5.

In this paper, we focus on the selection of neighborhood size in manifold learning methods for nonlinear dimensionality reduction, and concentrate on LLE without loss of generality. We propose that K should be self-tuning according to local density not a uniform value for all the data, and present a new variant algorithm of LLE, which can effectively prune “short circuit” edges by performing spatial search on the R*-Tree [7] built on the dataset. This pruning leads the original fixed neighborhood size to be a self-tuning value, thus makes our algorithm have more topologically stableness than LLE does.

2 Background

2.1 LLE Algorithm

Locally Linear Embedding [1][8] tries to find meaningful low-dimensional structure hidden in high-dimensional data. It maps a dataset $X = \{\bar{X}_1, \dots, \bar{X}_N\}$, $\bar{X}_i \in \mathbb{R}^D$, to a data set $Y = \{\bar{Y}_1, \dots, \bar{Y}_N\}$, $\bar{Y}_i \in \mathbb{R}^d$, where $d \ll D$. Formally the algorithm consists of three steps:

Step 1. For each data point \bar{X}_i , find its K nearest neighbors set $NE_i = \{\bar{X}_j \mid j \in J_i, |J_i| = K\}$.

Step 2. Compute the weights W_{ij} that best reconstruct each data point \bar{X}_i from its K nearest neighbors by solving a least squares problem as follows.

$$\min \varepsilon(W_{N \times N}) = \sum_{i=1}^N \left| \bar{X}_i - \sum_{j \in J_i} W_{ij} \bar{X}_j \right|^2, \text{ s.t. } \sum_{j \in J_i} W_{ij} = 1, W_{ij} = 0 \text{ if } j \notin J_i. \tag{1}$$

Step 3. For each data point \bar{X}_i , compute the vector \bar{Y}_i which best fits the reconstruction weights by solving the optimization problem as follows.

$$\min \Phi(Y_{d \times N}) = \sum_{i=1}^N \left| \bar{Y}_i - \sum_{j \in J_i} W_{ij} \bar{Y}_j \right|^2, \text{ s.t. } \sum_{i=1}^N \bar{Y}_i = \bar{0}, \frac{1}{N} \sum_{i=1}^N \bar{Y}_i \bar{Y}_i^T = I. \quad (2)$$

This optimization problem can be solved by calculating the non-zero bottom d eigenvectors of matrix $M = (I - W)^T (I - W)$, these eigenvectors form rows of matrix Y . We also list two properties the matrix M satisfies as follows, and these properties will help us to understand some results of LLE in Section 3.

1. M is symmetric and positive semi-definite.

2. M has N non-negative, real-valued eigenvalues $\lambda_N \geq \dots \geq \lambda_2 \geq \lambda_1 = 0$, and the corresponding eigenvector to 0 is the constant one vector $\bar{1}$.

2.2 R*-Tree

Since 1984 when Antonin Guttman first proposed R-Tree, it has become one of the most popular spatial index mechanisms which can help retrieve spatial data more efficiently according to their spatial locations. During these years, researchers and practitioners have applied R-Tree everywhere, from CAD and GIS to Multimedia Information Retrieval, and have made many variations including R+-Tree, R*-Tree, TV-Tree, X-Tree, Pyramid-Tree. Details about R-Tree and its variations can be found in [9].

R*-Tree [7] which is used in our topologically stable LLE algorithm, is also a variant of R-Tree. It deals with the problem as follows. Given a spatial dataset: $S = \{s_1, \dots, s_n\}$, $s_i \in \mathbb{R}^m$, and a m -dimensional bounding box represented by $I = (I_1, I_2, \dots, I_m)$, I_i is a closed bounded interval $[a_i, b_i]$, using what spatial index mechanism can we retrieve the data $R = \{r_i \mid r_i \in S, r_i \text{ locates in } I\}$ quickly and precisely? Based on the R*-Tree built on S , we can do the retrieval very easily. Particularly, we construct I in our algorithm as follows. Given any location $l_k = (l_{k1}, \dots, l_{km}) \in \mathbb{R}^m$ and a positive real-valued range ε , I_i is a closed bounded interval $[l_{ki} - \varepsilon, l_{ki} + \varepsilon]$.

3 Problem Formation and Related Work

As we can see, LLE has one free parameter K - the number of neighbors used in Step 1. K controls the range of neighbors based on which we reconstruct a data point, and we think the optimal K should satisfy two requirements at the same time:

1. The local linearity is preserved in K nearest neighbors.
2. K is as large as possible.

The first requirement is easy to understand, as for requirement 2, we can see that the larger K is, the more information the neighbors can provide for reconstruction in Step 2. But it is very difficult to select the optimal K satisfying the two requirements manually. A small K can divide a continuous manifold into unconnected sub-manifolds. Fig. 1a-c illustrate the result of LLE performing on modified S-Curve dataset with a small K .

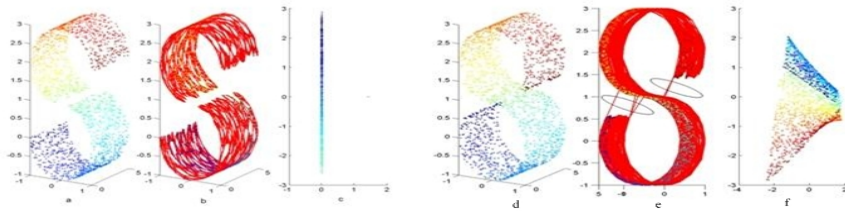


Fig. 1. (a) A broken S-Curve dataset with 2000 points. (b) The corresponding neighborhood graph on the broken S-Curve where $K=7$. (c) The result of LLE over the broken S-Curve where $K=7$. (d) A S-Curve dataset with 2000 points. (e) The corresponding neighborhood graph on S-Curve where $K=20$, the edges circled by a ring are “short circuit” edges. (f) The result of LLE over S-Curve where $K=20$.

Fig. 1a-c show that LLE performs very badly when K is not large enough to make the entire graph a connected one. We can interpret this situation when considering the properties of matrix M which are mentioned in Section 2.1. Considering M consists of C connected components ($C=2$ in Fig. 1), without loss of generality, we assume that the data points are ordered according to the connected components they belong to. In this case, M has a block diagonal form: $M = \text{diag}(M_1, \dots, M_C)$. Note that each block M_i has the same properties as M . Thus, we know that M has eigenvalue 0 with multiplicity C , and the corresponding eigenvectors are the indicator vectors of the connected components with 1 at the positions of one block and 0 at the positions of the other blocks. LLE assumes the multiplicity of eigenvalue 0 of M is 1, so it performs badly when the multiplicity is larger than 1.

In contrast, a large K tends to violate the requirement 1 through introducing “short circuit” edges [4] into the neighborhood graph. Fig. 1 d-f show the failure caused by “short circuit” edges.

Some work [5][6] focused on choosing the optimal K automatically. They usually choose an interval of possible values of K firstly, then determine the optimal K by calculating the predefined cost function for each candidate in the interval. These methods are computationally demanding as a result of enumerating every candidates of K . And the cost function measuring the quality of input-output mapping is hard to define. [2] proposed to use the residual variance to evaluate the mapping. The residual variance in [2] is defined as $1 - \rho_{\hat{D}_M D_Y}^2$, where D_Y is the matrix of Euclidean distances in the low-dimensional embedding, \hat{D}_M is a best estimate of the intrinsic manifold distances D_M , and ρ is the standard linear correlation coefficient. But how to compute \hat{D}_M which is a good estimate of real manifold distances D_M , especially for data in real world, is a very hard problem. [5] and [6] used $K_{opt} = \arg \min_K (1 - \rho_{D_X D_Y}^2)$ to specify the optimal K . where D_X is the matrix of Euclidean distances in the high-dimensional input space. It is obviously wrong to take D_X as an estimate of D_M , it even doesn't work on S-Curve and Swiss-Roll datasets, we will demonstrate this by experimental results in Section 5.

In fact, we can calculate the exact D_M on some artificial datasets, such as S-Curve and Swiss-Roll. We ignore the dimensionality of height, and only discuss the computation of arc length on two-dimensional S-Curve and Swiss-Roll. The two dimensional S-Curve consists of connected two pieces of circular curves with opposite rotation orientations, the arc length between two points with angles θ_1 and θ_2 (measured in radians) is simply: $s = r * |\theta_1 - \theta_2|$, where r is the radius.

In polar coordinates, two-dimensional Swiss-Roll is defined as $r = \theta$. So the arc length is:

$$s = \int_{\theta_1}^{\theta_2} \sqrt{r^2 + \left(\frac{dr}{d\theta}\right)^2} d\theta = \frac{\theta}{2} \sqrt{\theta^2 + 1} + \frac{1}{2} \ln \left| \theta + \sqrt{\theta^2 + 1} \right| \Big|_{\theta_1}^{\theta_2} \tag{3}$$

We use $1 - \rho_{D_M, D_y}^2$ to evaluate LLE over artificial datasets in the following sections.

4 Our Solution Based on R*-Tree

As we can see in Section 3, it is not only difficult to specify the optimal K automatically, but also unreasonable to assume that each data point share the same number of neighbors. In fact, LLE ignores the effect of local scaling [11], when the dataset includes subsets with different statistics there may not be a single value of K that works well for all the data.

Actually, every data point should have its own optimal K , not a uniform value. The individual optimal K should subject to the two constraints proposed in Section 3, it means that for each data point, K should be as large as possible without introducing the “short circuit” edges into the neighborhood graph. Investigating into the “short circuit” edge, we can find that it usually passes by a low-density area in which very sparse data points are located. We can detect and prune the “short circuit” edges based on this property. [10] proposed the pruned-ISOMAP algorithm, which first constructs an neighborhood graph with a relative large K , then prunes “short circuit” edges existed possibly in the graph based on kernel density estimation. But the pruning based on kernel density estimation in [10] is computationally demanding, and needs to specify manually two elaborate parameters which depict the kernel function. In fact, if we just want to prune “short circuit” edge, we need not to know the exact local density at the edge, the computation of which is very time consuming. Based on a spatial index built on the dataset, we can do the pruning more efficiently. In this paper we use R*-Tree to do the pruning.

4.1 “Short Circuit” Edge Pruning Based on R*-Tree

As we discussed in Section 2.2, given a data point and a range, R*-Tree helps us retrieve its neighbors efficiently. So we can detect “short circuit” edge by counting the number of its neighbors based on R*-Tree. As for an edge $E(X_i, X_j)$ connecting two data points X_i and X_j , and $X_i, X_j \in \mathbb{R}^n$. We sample the bisecting point on the edge for the consideration of computing efficiency: $X_{ij} = (X_i + X_j) / 2$. For the sample point

$X_{ij} = (x_1, \dots, x_n)$, we perform a R*-Tree search for the neighbors of X_{ij} in a n-dimensional bounding box represented by $I = (I_1, I_2, \dots, I_n)$, where I_m is a closed bounded interval $[x_m - \varepsilon_{ij}, x_m + \varepsilon_{ij}]$. ε_{ij} is an important value that will be discussed later in this section. Now we denote the number of neighbors retrieved by the R*-Tree search as C_{ij} , $C_{ij} \in \{0, 1, 2, \dots\}$, and we define a metric $E_{ij} = C_{ij}$ on the edge $E(X_i, X_j)$. A small E_{ij} means there are sparse data points located in the neighborhood of the edge, and it is very likely to be a “short circuit” edge, while a large E_{ij} means the opposite. E_{ij} is a simple and effective metric which helps us prune the “short circuit” edges simply by setting a threshold on it. We set 0 as the threshold in our algorithm, it means edge $E(X_i, X_j)$ is recognized as a “short circuit” edge if E_{ij} equals 0, a normal edge if E_{ij} is larger than 0. Fig. 2 illustrates the histograms of E_{ij} on two neighborhood graphs, and shows that it is suitable for us to set the threshold to be 0.

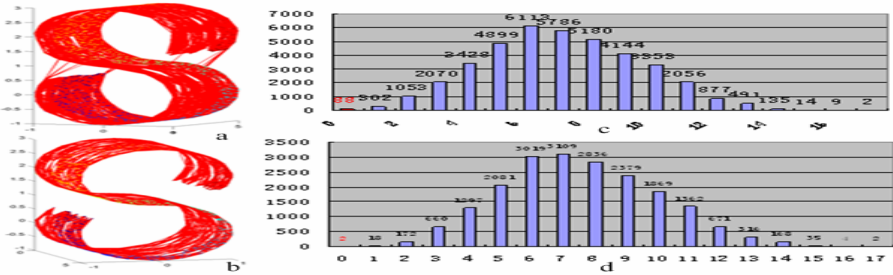


Fig. 2. (a) Neighborhood graph on S-Curve with N=2000, K=20. (b) Neighborhood graph on S-Curve with N=2000, K=10. (c) Histogram of E_{ij} on neighborhood graph in a. (d) Histogram of E_{ij} on neighborhood graph in b.

As for ε_{ij} which confines the searching range, it should deal with the effect of local scaling [11], otherwise it tends to recognize the edges in large scale as “short circuit” edges. It means that ε_{ij} should be self-tuning according to the local scales, dense distribution fits to a small ε_{ij} , while sparse distribution fits to a large one. We define ε_{ij} for edge $E(X_i, X_j)$ as follows.

$$\varepsilon_{ij} = \min(S(X_i), S(X_j)), \quad S(X_i) = \frac{1}{n} \sum_{X_k \in NE_n(X_i)} \|X_i - X_k\| \quad (4)$$

Where $NE_n(X_i)$ is the n nearest neighbors set of X_i , and n=2 in our algorithm.

4.2 The More Topologically Stable LLE Algorithm

In summary, we propose a more topologically stable LLE algorithm as follows.

1. Choose a relative large K with which the neighborhood graph is connected at least (1%-5% of total number of data points is recommended). Then construct a K nearest neighborhood graph on the dataset based on Euclidean metric.
2. Prune the “short circuit” edges.
 - 2.1. Create an R*-Tree on the dataset.
 - 2.2. For every edge $E(X_i, X_j)$ in the K nearest neighborhood graph.
 - 2.2.1 Specify the bisecting point X_{ij} .
 - 2.2.2 Calculate the searching range ε_{ij} in Eq. (4).
 - 2.2.3 Compute E_{ij} by performing a R*-Tree search for the neighbors of X_{ij} within ε_{ij} .
 - 2.2.4 Prune the edge if its E_{ij} equals 0.
3. Run LLE on the pruned neighborhood graph.

5 Experimental Results

In this section, we present several examples to illustrate the performance of our algorithm that we name as R*-Tree LLE for brevity. We give both subjective and objective results: visualization of output data and residual variance metric which is discussed in Section 3. The test datasets include S-Curve and Swiss-Roll [8].

First, we compare R*-Tree LLE to LLE on uniformly sampled S-Curve and Swiss-Roll under different neighborhood sizes to illustrate the topologically stableness of our algorithm. Fig. 3 and Fig. 6a illustrate the comparison on 2000-point uniformly sampled S-Curve. Fig. 4 and Fig. 6b illustrate the comparison on 2000-point uniformly sampled Swiss-Roll. The residual variance in Fig. 6 is obtained by Eq. (5).

$$RV_Y1M = 1 - \rho_{D_M D_{Y1}}^2 \quad RV_Y2M = 1 - \rho_{D_M D_{Y2}}^2 \quad RV_Y1X = 1 - \rho_{D_X D_{Y1}}^2 \quad (5)$$

Where D_M is the matrix of manifold distances which is discussed in Section 3, D_{Y1} and D_{Y2} are the matrices of Euclidean distances in the output low-dimensional embeddings of LLE and R*-Tree LLE separately. D_X is the matrix of Euclidean distances in the input high-dimensional space. From these figures, we can see R*-Tree LLE has more reasonable visual results and lower residual variances than LLE does, especially when the neighborhood size is large. But we also find that the performances of the two algorithms are similar when the neighborhood size is small. That is because the 2000-point S-Curve and Swiss-Roll are sampled uniformly from two smooth manifolds, thus each data point has nearly the same optimal neighborhood size as it has nearly the same local linearity in its neighborhood. Meanwhile, it is not likely to introduce the “short circuit” edges into the neighborhood graph when the neighborhood size is small. In Fig. 6a, we also show the residual variance RV_Y1X used in [5] and [6], it is obvious to see RV_Y1X is an improper evaluation metric for LLE as we know in Section 3.

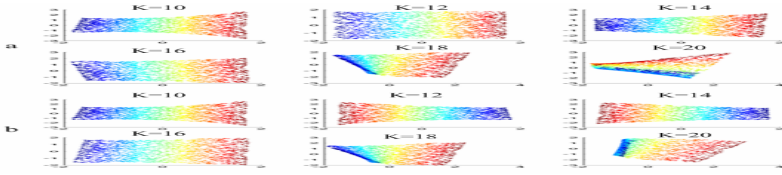


Fig. 3. (a) Results of LLE on 2000-point uniformly sampled S-Curve under different neighborhood sizes. (b) Results of R*-Tree LLE on 2000-point uniformly sampled S-Curve under different neighborhood sizes.

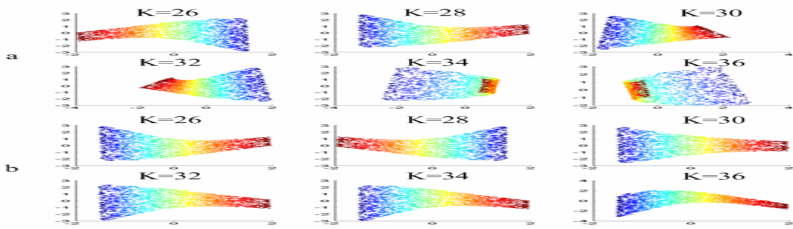


Fig. 4. (a) Results of LLE on 2000-point uniformly sampled Swiss-Roll under different neighborhood sizes. (b) Results of R*-Tree LLE on 2000-point uniformly sampled Swiss-Roll under different neighborhood sizes.

Then, we give the following example to illustrate a self-tuning neighborhood size used in R*-Tree LLE is more reasonable than a uniform neighborhood size used in LLE. We generate the data points from part of Swiss-Roll with a missing rectangle strip, as Fig. 5a illustrates. So the resulting modified Swiss-Roll is not sampled uniformly, and has different densities at different locations. In this case, a self-tuning neighborhood size has more superiorities over a uniform one. From Fig. 5 and Fig. 6c, we can see R*-Tree LLE performs much better than LLE at all the candidate neighborhood sizes, including at the optimal value for LLE.

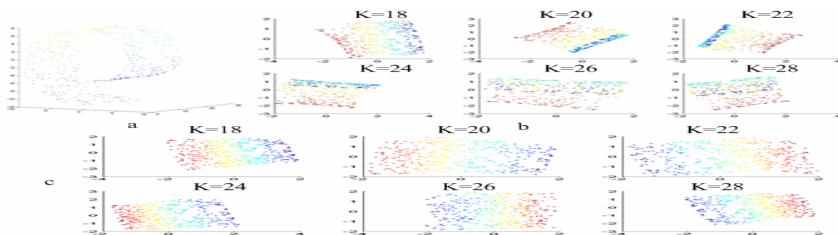


Fig. 5. (a) 450-point non-uniformly sampled Swiss-Roll. (b) Results of LLE on 450-point non-uniformly sampled Swiss-Roll under different neighborhood sizes. (c) Results of R*-Tree LLE on 450-point non-uniformly sampled Swiss-Roll under different neighborhood sizes.

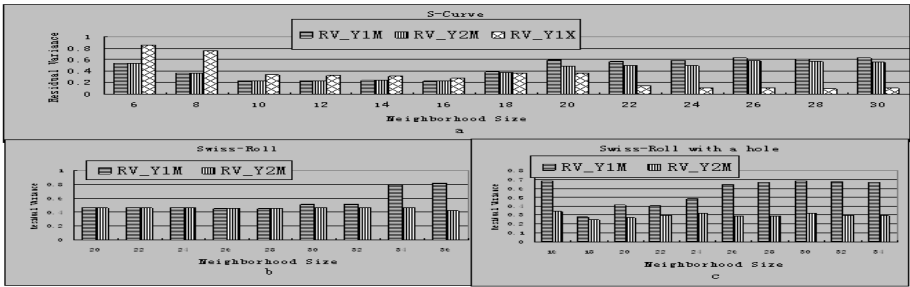


Fig. 6. (a) Residual variances on 2000-point uniformly sampled S-Curve under different neighborhood sizes. (b) Residual variances on 2000-point uniformly sampled Swiss-Roll under different neighborhood sizes. (c) Residual variances on 450-point non-uniformly sampled Swiss-Roll under different neighborhood sizes.

Finally, we give the execution time of the R*-Tree operations in our algorithm which is the necessary time price we pay for higher performance compared to LLE. Implemented in C++ on a personal computer with Pentium-4 3.40 GHz CPU, 1 GB Memory, and MS Windows XP Professional SP2, it takes around 250 ms to create an R*-Tree over 2000 data points, and 0.4 ms to perform a search operation over 2000-point R*-Tree.

6 Conclusion

In this paper, we explore the selection of neighborhood size in LLE, and propose that the neighborhood size should be self tuning according to the local density. Based on this idea, we propose a new variant of LLE which use self tuning K through pruning “short circuit” edges based on R*-Tree.

There are, however, some open problems. In our algorithm, we test every edge while do pruning, in future work, we plan to accelerate the pruning process by a two-step pruning, the first step pruning is based on global information, and the second one depends on local density. The first step is very time efficient, and largely reduces the edges for the second step pruning. We also plan to use other spatial indices, such as TV-Tree and X-Tree, to substitute for R*-Tree, these indices outperforms R*-Tree in indexing high dimensional data. Finally we plan to extend our method to spectral clustering which also needs to construct a neighborhood graph in its algorithm.

References

1. Roweis, S.T., Saul, L.K.: Nonlinear dimensionality reduction by locally linear embedding. *Science* 290, 2323–2326 (2000)
2. Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. *Science* 290, 2319–2323 (2000)

3. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* 15(6), 1373–1396 (2003)
4. Balasubramanian, M., Schwartz, E.L., Tenenbaum, J.B., de Silva, V., Langford, J.C.: The ISOMAP algorithm and topological stability. *Science* 295, 5552.7a (2002)
5. Samko, O., Marshall, A.D., Rosin, P.L.: Selection of the optimal parameter value for the ISOMAP algorithm. *Pattern Recognition Letters* 27, 968–979 (2006)
6. Kouropteva, O., Okun, O., Pietikainen, M.: Selection of the optimal parameter value for the Locally Linear Embedding algorithm. In: *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery*, pp. 359–363 (2002)
7. Beckmann, N., Kriegel, H.P., Schneider, R., Seeger, B.: The R*-Tree: an efficient and robust access method for points and rectangles. In: *Proceeding ACM SIGMOD International Conference on Management of Data*, NJ, USA, pp. 322–331 (1990)
8. <http://www.cs.toronto.edu/~roweis/lle/>
9. <http://www.rtreeportal.org/>
10. Chao, S., Hou-kun, H., Lian-wei, Z.: A more topologically stable ISOMAP algorithm. *Journal of Software* 18(4), 869–877 (2007)
11. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. *Advances in Neural Information Processing Systems* 17, 1601–1608 (2005)

Sparse Kernel-Based Feature Weighting^{*}

Shuang-Hong Yang^{1,2}, Yu-Jiu Yang^{1,2}, and Bao-Gang Hu^{1,2}

¹ National Lab of Pattern Recognition(NLPR) & Sino-French IT Lab(LIAMA)

Institute of Automation, Chinese Academy of Sciences

² Graduate School, Chinese Academy of Sciences

P.O. Box 2728, Beijing, 100080 China

{shyang,yjyang,hubg}@nlpr.ia.ac.cn

Abstract. The success of many learning algorithms hinges on the reliable selection or construction of a set of highly predictive features. Kernel-based feature weighting bridges the gap between feature extraction and subset selection. This paper presents a rigorous derivation of the Kernel-Relief algorithm and assesses its effectiveness in comparison with other state-of-art techniques. For practical considerations, an online sparsification procedure is incorporated into the basis construction process by assuming that the kernel bases form a causal series. The proposed sparse Kernel-Relief algorithm not only produces nonlinear features with extremely sparse kernel expressions but also reduces the computational complexity significantly.

1 Introduction

Reducing the dimensionality of the raw data is an important preprocessing step in data mining. It plays a fundamental role in practices for a variety of reasons [4].

In the literature, there are two major types of methods: while

(FS) methods identify a subset of useful features and discard others,

(FE) approaches construct new features out of the original ones.

Traditional feature selection algorithms are conducted in the original input space. Therefore they cannot satisfactorily capture the inherent nonlinear structures in the data. Quite recently, Cao et al [2] proposed a kernel-based feature weighting algorithm based on the Relief algorithm [5,7]. By conducting feature selection in the kernel space, Kernel-Relief (K-Relief) [2] actually bridges the gap between FS and FE, i.e., it achieves the purpose of FE through FS in a nonlinear space. Although the idea is quite interesting, the problems occurred in their mathematic derivations and numerical comparisons strongly weakened the reliability of their study. In addition, an practical shortcomings of K-Relief is that the nonlinear features constructed by this algorithm has a non-sparse kernel expression, which could become prohibitive in practice especially when large scale problems are concerned (see Section 4 for details).

In this paper, we first revisit the K-Relief algorithm proposed by [2] both theoretically and empirically. In particular, detailed rigorous derivations are provided to produce reliable formulations. And numerical evaluations are carried out

^{*} This work is supported in part by NSFC (#60275025, #60121302).

under more reliable configurations. Furthermore, an online greedy sparsification method was proposed to achieve very sparse expressions of the features while preserving their effectiveness as much as possible.

The organization of the rest parts of the paper is as follows. Section 2 briefly introduce the key ideas of K-Relief algorithm. Section 3 revisits the algorithm and provides a detailed rigorous derivation. Section 4 proposes a sparsification procedure to build sparse K-Relief. Section 5 presents the empirical studies, and Section 6 concludes the whole paper.

2 Kernel-Based Relief

Suppose we are given a set of input vectors $\{\mathbf{x}_n\}_{n=1}^N$ along with corresponding targets $\{y_n\}_{n=1}^N$, where $\mathbf{x}_n \in \mathbf{X} \subset \mathcal{R}^D$ is a training instance and $y_n \in \mathbf{Y} = \{0, 1, \dots, C-1\}$ is its label, N, D, C denote the training set size, the input space dimensionality and the total number of categories respectively, and the d -th feature of \mathbf{x} is denoted as $x^{(d)}$, $d=1, 2, \dots, D$. The Relief algorithm [5] ranks the features according to the weights w_d 's obtained from the following equation [7]:

$$\begin{aligned} \mathbf{w} &= \arg \max \sum_{n=1}^N \mathbf{w}^T \mathbf{m}_n \\ \text{s.t. : } & \|\mathbf{w}\| = 1, w_d \geq 0, d = 1, 2, \dots, D \end{aligned} \quad (1)$$

where $\mathbf{w} = (w_1, w_2, \dots, w_D)^T$, $\mathbf{m}_n = |\mathbf{x}_n - M(\mathbf{x}_n)| - |\mathbf{x}_n - H(\mathbf{x}_n)|$ is called the margin for the pattern \mathbf{x}_n , $H(\mathbf{x}_n)$ and $M(\mathbf{x}_n)$ denote the nearest-hit (the nearest neighbor from the same class) and nearest-miss (the nearest neighbor from different class) of \mathbf{x}_n respectively.

Traditional linear feature selection methods are not appropriate for non-linearly separable classification problems. Cao et al [2] established an algorithm that allows Relief to be approximately carried out in the kernel space. Suppose the data have been mapped from the original input space \mathcal{R}^D to a high (usually infinite) dimensional feature space \mathcal{F} through an implicit mapping function $\phi: \mathcal{R}^D \rightarrow \mathcal{F}$, which is induced by the kernel function $k(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}), \phi(\mathbf{x}') \rangle$, Cao et al [2] proposed to implement the kernel-based Relief (K-Relief) by the following procedures:

- (a) Construct an orthogonal basis set $\{\mathbf{v}^{(l)} = \sum_{j=1}^N \alpha_{lj} \phi(\mathbf{x}_j) | l = 1, 2, \dots, L\}$ of the feature space by kernel Gram-Schmidt process, where $L \leq \text{rank}(K)$;
- (b) Apply Relief to the space spanned by the constructed basis set and select features based on the feature weights.

Though the idea is very interesting, some problems occurred in their study: (i) They failed to provide a rigorous theoretical derivation of the algorithm. As a result, several key formulas in both procedure (a) and (b) are incorrect, making their algorithm unreliable. (ii) The configuration of their comparison study is problematic, which makes the empirical performance of K-Relief unclear. Firstly, if one wishes to compare different DR methods using the classification accuracy as the evaluation metric, it is necessary to fix the classifier. However, in their study, this discipline is defied. Secondly, to fairly assess K-Relief, one needs to compare it with some state-of-art feature extraction techniques.

3 Kernel-Relief Revisiting

In this section, we provide a rigorous derivation of K-Relief algorithms as follows:

(a) **Kernel Gram-Schmidt Process (GP)**. Suppose the l -th orthogonal basis can be expressed as $\mathbf{v}^{(l)} = \sum_{n=1}^N \alpha_{ln} \phi(\mathbf{x}_n)$. Denote the design matrix in the feature space as $\Phi = (\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_n))^T$, let $\alpha^{(l)} = (\alpha_{l1}, \dots, \alpha_{lN})^T$, and $\mathbf{e}^{(l)}$ be the l -th standard basis of the Euclidean space (the vector with l -th component be 1 and all others be zero). Applying the Gram-Schmidt orthogonalization process we have:

$$\begin{aligned} \mathbf{v}^{(l)} &= \phi(\mathbf{x}_l) - \sum_{k=1}^{l-1} \langle \phi(\mathbf{x}_l), \mathbf{v}^{(k)} \rangle \mathbf{v}^{(k)} \\ &= \langle \mathbf{e}^{(l)}, \Phi \rangle - \left\langle \left(\sum_{k=1}^{l-1} \langle \phi(\mathbf{x}_l), \mathbf{v}^{(k)} \rangle \alpha^{(k)} \right), \Phi \right\rangle \\ &= \left\langle \left(\mathbf{e}^{(l)} - \sum_{k=1}^{l-1} \alpha^{(k)} \sum_{j=1}^N \alpha_{kj} k(\mathbf{x}_l, \mathbf{x}_j) \right), \Phi \right\rangle \end{aligned} \tag{2}$$

To get an orthogonal basis set, we need to normalize each basis.

$$\| \mathbf{v}^{(l)} \|^2 = \sum_{j=1}^N \sum_{k=1}^N \alpha_{lj} \alpha_{lk} k(\mathbf{x}_j, \mathbf{x}_k) \tag{3}$$

Therefore, we have the l -th basis: $\mathbf{v}^{(l)} = \langle (\alpha^{(l)}), \Phi \rangle = \sum_{j=1}^N \alpha_{lj} \phi(\mathbf{x}_j)$, where:

$$\alpha^{(l)} = \frac{\mathbf{e}^{(l)} - \sum_{k=1}^{l-1} \alpha^{(k)} \sum_{j=1}^N \alpha_{kj} k(\mathbf{x}_l, \mathbf{x}_j)}{\sqrt{\alpha^{(l)T} K \alpha^{(l)}}} \tag{4}$$

(b) **Applying Relief in Kernel Space**. We are now ready to project the data into the space spanned by the basis set that we have constructed.

$$\phi(\mathbf{x}_n) \approx \sum_{l=1}^L \langle \phi(\mathbf{x}_n), \mathbf{v}^{(l)} \rangle \mathbf{v}^{(l)} \tag{5}$$

or equivalently:

$$\begin{aligned} \psi(\mathbf{x}_n) &= \begin{pmatrix} \psi_1(\mathbf{x}_n) \\ \dots \\ \psi_L(\mathbf{x}_n) \end{pmatrix} = \begin{pmatrix} \langle \phi(\mathbf{x}_n), \mathbf{v}^{(1)} \rangle \\ \dots \\ \langle \phi(\mathbf{x}_n), \mathbf{v}^{(L)} \rangle \end{pmatrix} = \begin{pmatrix} \sum_{j=1}^N \alpha_{1j} \langle \phi(\mathbf{x}_n), \phi(\mathbf{x}_j) \rangle \\ \dots \\ \sum_{j=1}^N \alpha_{Lj} \langle \phi(\mathbf{x}_n), \phi(\mathbf{x}_j) \rangle \end{pmatrix} \\ &= A \times \mathbf{k}_n \end{aligned} \tag{6}$$

That is, the design matrix in \mathfrak{F} can be expressed as:

$$\Psi = (\psi(\mathbf{x}_1), \dots, \psi(\mathbf{x}_n))^T = A^T \times K \tag{7}$$

where $\psi : \mathcal{R}^D \rightarrow \mathfrak{F}$, $\mathfrak{F} = \{ \sum_{l=1}^L \lambda_l \mathbf{v}^{(l)} | \lambda_l \in \mathcal{R} \}$ is the subspace spanned by the basis set, $A = (\alpha_{ln})_{L \times N}$, and \mathbf{k}_n denotes the n -th column of K . This approximate

Algorithm 1. Kernel-Relief based on Gram-Schmidt basis (K-Relief)

Input: Gram matrix K , label vector \mathbf{y}
Output: A set of M nonlinear features $\{\psi_m(\mathbf{x})|m = l_1, l_2, \dots, l_M\}$
for $l=1$ **to** L
 calculate $\boldsymbol{\alpha}^{(l)}$ by Eq.(4)
end for
Calculate feature weights by Eq.(9)
Select l_1, l_2, \dots, l_M based on the rank of weights
Project the data set by $\{\psi_m(\mathbf{x})|m = l_1, l_2, \dots, l_M\}$, where ψ_m is defined by Eq.(6)

expression of Φ makes it possible to apply Relief algorithm in the kernel space. In particular, Relief in \mathfrak{F} solves the following optimization problem:

$$\begin{aligned} \max \quad & \sum_{n=1}^N \mathbf{w}^T \mathbf{m}_n \\ \text{s.t.} \quad & \|\mathbf{w}\| = 1, w_l \geq 0, l = 1, 2, \dots, L \end{aligned} \tag{8}$$

where $\mathbf{w}=(w_1, \dots, w_L)^T$, $\mathbf{m}_n = |\psi(\mathbf{x}_n) - M(\psi(\mathbf{x}_n))| - |\psi(\mathbf{x}_n) - H(\psi(\mathbf{x}_n))|$, and $H(\psi(\mathbf{x}_n))$ and $M(\psi(\mathbf{x}_n))$ denote the nearest-hit and nearest-miss of \mathbf{x}_n in \mathcal{F} respectively, which can be found based on the distance $d_{\mathcal{F}}(\mathbf{x}, \mathbf{x}') = \langle \phi(\mathbf{x}) - \phi(\mathbf{x}'), \phi(\mathbf{x}) - \phi(\mathbf{x}') \rangle = k(\mathbf{x}, \mathbf{x}) + k(\mathbf{x}', \mathbf{x}') - 2k(\mathbf{x}, \mathbf{x}')$.

Eq.(2) has an explicit solution:

$$\mathbf{w} = (\mathbf{z})^+ / \|\mathbf{z}\|^+ \tag{9}$$

$$\mathbf{z} = \sum_{n=1}^N (|\psi(\mathbf{x}_n) - M(\psi(\mathbf{x}_n))| - |\psi(\mathbf{x}_n) - H(\psi(\mathbf{x}_n))|). \tag{10}$$

The K-Relief algorithm is described as Algorithm 1. Note that several key formulas in [2], such as Eq.(4), (9) and (10), are incorrect.

Once the feature weights are obtained from Eq.(9), one can select a preferable subset of M ($M \ll L$) features $\{\psi_m(\mathbf{x})|m = l_1, l_2, \dots, l_M\}$ according to the ranking of the weights. Then one either applies the selected feature mapping $\{\psi_m(\mathbf{x})|m = l_1, l_2, \dots, l_M\}$ according to Eq.(6) and builds a classifier directly based on these features, or else, one can compute a kernel matrix K_w and build a kernel machine based on K_w , where $K_w(\mathbf{x}, \mathbf{x}') = \sum_{m=1}^M \psi_{l_m}(\mathbf{x})\psi_{l_m}(\mathbf{x}')$.

4 Sparse Kernel-Relief

A fundamental problem with the K-Relief algorithm is that the nonlinear feature has a non-sparse kernel expression, which may not be eliminated even when sparse kernel machines, such as SVM and RVM, are applied to the tasks because each feature is expressed as a linear combination of kernel functions centered at all the training data points (see Eq.(6)). In practice, this could cause severe over-fitting and would also become a crucial computation prohibition especially when we are facing large scale problems or online learning requirements, because

both the memory for storing the kernel matrix and the training & testing time are typically proportional to the number of support patterns¹ (denote as N_{sp}).

The non-sparseness stems from the basis construction process. Therefore, if we can construct a set of orthogonal bases which has a sparse kernel expression, i.e., which can be expressed as a linear combination of a small set of the kernel vectors, then we can easily achieve a sparse K-Relief algorithm. Clearly, the sparse kernel PCA (SKPCA, [8]) satisfies this requirement. Here however, we shall establish an online sparsification procedure for the kernel Gram-Schmidt process.

Our goal is to recursively select L ($M \ll L \ll N$) out of N kernel vectors to construct a set of L orthogonal bases. Denote $\{\mathbf{k}_j : j = i_1, i_2, \dots, i_L\}$ the selected kernel vector subset (dictionary), the samples in $\{\mathbf{x}_j : j = i_1, i_2, \dots, i_L\}$ are the support patterns. Clearly, if we have obtained all of the L support patterns, it would be straightforward to apply Gram-Schmidt process on the sub-matrix $K_L = (k(\mathbf{x}_j, \mathbf{x}_k)|_{j, k \in \{i_1, \dots, i_L\}})_{L \times L}$. However, compared to this offline approach, an online construction would be more preferable.

The proposed sparse Gram-Schmidt process starts with a randomly chosen kernel vector \mathbf{k}_{i_1} . Denote $ind^{(l)} = \{i_1, i_2, \dots, i_{l-1}\}$ the index set of support patterns at l -th step, $\{1, 2, \dots, N\}$ the full index set, and $(:, ind^{(l)})$ the matrix comprised by the $ind^{(l)}$ -indexed columns of the Gram matrix K . Suppose at l -th step, we have collected a dictionary $K_{ind} = (:, ind^{(l)})$, the major operating procedures from l -th to $(l+1)$ -th step can be summarized as follows.

(1) **Choose a new pattern.** Randomly choose i_l from $\{1, 2, \dots, N\} - ind^{(l)}$.

(2) **Approximate linear dependence (ALD) test [3].** The new candidate basis \mathbf{k}_{i_l} can be viewed to be approximately expressed by linear combination of K_{ind} , if the ALD condition is satisfied, i.e.:

$$\varepsilon^{(l)} = \min \left\| \sum_{j \in ind^{(l)}} u_j \mathbf{k}_j - \mathbf{k}_{i_l} \right\|^2 \leq \xi \tag{11}$$

There exists a close form solution to Eq.(6).

$$\varepsilon^{(l)} = \|\mathbf{k}_{i_l} - \mathbf{u}^T K_{ind}\|^2, \text{ where } \mathbf{u} = (K_{ind}^T K_{ind})^{-1} K_{ind}^T \mathbf{k}_{i_l} \tag{12}$$

and ξ is a small constant which determines the sparsity degree. If ALD condition satisfies, \mathbf{k}_{i_l} will not be added to the dictionary; else, we need to add i_l to $ind^{(l)}$ and construct a new basis vector based on K_{ind} .

(3) **Construct the l -th orthogonal basis.** Suppose the bases form a $(l-1)$ -dimensional subspace, i.e., l -th basis can be expressed as the linear combination of the previously selected $(l-1)$ support patterns, i.e.: $\mathbf{v}^{(l)} = \sum_{k \in ind^{(l)}} \beta_{lk} \phi(\mathbf{x}_k)$. Applying the Gram-Schmidt process, we have:

$$\mathbf{v}^{(l)} = \frac{\phi(\mathbf{x}_{i_l}) - \sum_{k=1}^{l-1} \langle \phi(\mathbf{x}_{i_l}), \mathbf{v}^{(k)} \rangle \mathbf{v}^{(k)}}{\|\mathbf{v}^{(l)}\|} \tag{13}$$

¹ With a slight abuse of terminology, here the patterns that appear in the expression of the basis set are called support patterns.

Algorithm 2. K-Relief based on Sparse Gram-Schmidt basis (SK-Relief)

Input: Gram matrix K , label vector \mathbf{y} , ξ , $ind^{(0)} = \emptyset$, $Ind = \{1, \dots, N\}$
Output: A set of M nonlinear features with sparse kernel expressions
for $l=1$ **to** L
 Randomly choose i_l from $Ind-ind^{(k)}$.
 ALD test: Eq. (11) (12)
 if (Not ALD)
 Add to Dictionary: $ind^{(l)} = ind^{(l)} \cup \{i_l\}$
 Construct a new basis $\mathbf{v}^{(l)}$: Eq. (13)
 end if
end for
Calculate the linear combination coefficients matrix: Eq. (15)
Calculate feature weights by Eq. (9)
Select M features based on the rank of weights
Project the data set by $\{\psi_m(\mathbf{x}) | m = 1, 2, \dots, l_M\}$, where ψ_m is defined by Eq. (6)

Solving β_{ik} from Eq. (13), we have:

$$\boldsymbol{\beta}^{(l)} = \frac{\mathbf{e}^{(l)} - \sum_{k=1}^{l-1} \boldsymbol{\beta}^{(k)} \sum_{j=1}^{k-1} \beta_{kj} k(\mathbf{x}_{i_l}, \mathbf{x}_j)}{\sum_{j \in ind^{(l)}} \sum_{k \in ind^{(l)}} \beta_{lj} \beta_{lk} k(\mathbf{x}_j, \mathbf{x}_k)} \quad (14)$$

where $\mathbf{e}^{(l)}$ is an l -by-1 sized vector with the last element be 1 and all others equal to zero, $\boldsymbol{\beta}^{(l)} = (\beta_{l1}, \dots, \beta_{ll})^T$ is also an l -by-1 sized vector. This means that the linear combination coefficients defined in Eq. (6) form a lower triangular matrix:

$$A_l = (\alpha_{ij})_{l \times l} = \begin{pmatrix} \beta_{11} & 0 & 0 & \dots & 0 \\ \beta_{21} & \beta_{22} & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \beta_{l-1,1} & \beta_{l-1,2} & \dots & \beta_{l-1,l-1} & 0 \\ \beta_{l1} & \beta_{l2} & \dots & \beta_{l,l-1} & \beta_{ll} \end{pmatrix} \quad (15)$$

The derived algorithm is described as Algorithm.2. Note that the computation complexity has been reduced from $O(N^2)$ to $O(N_{sp}L)$.

5 Experiments

In this section, we conduct extensive experiments to evaluate the effectiveness of the proposed methods in comparison with several state-of-art kernel-based FE techniques. Six benchmark machine learning data sets from UCI collection are selected because of their diversity in the numbers of features, instances and classes. The information of each data set is summarized in Table 1. To eliminate statistical deviation, all the results are averaged over 20 random runs. The k NN classifier is tested and the testing accuracy is used to evaluate the performances of different FE methods. In all experiments, Gaussian RBF kernels are used. The hyper-parameters, i.e., the number of nearest neighbors and the kernel width, are both determined by five-fold cross validation.

Table 1. Characteristics of six UCI data sets

Data Set	Train Size	Test Size	#Feature	#Class
Breast	400	283	9	2
Ringnorm	1400	6000	20	2
Pima	400	368	8	2
LRS	380	151	93	48
Ecoli	200	136	7	8
Glass	120	94	9	6

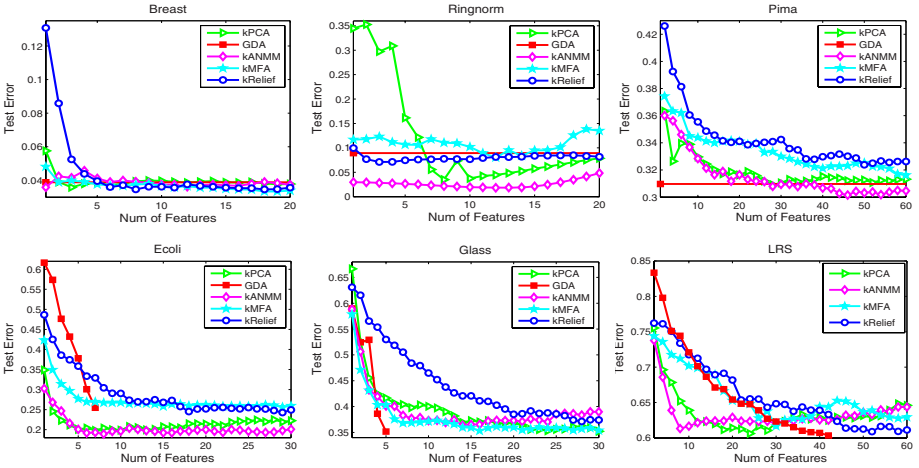


Fig. 1. Comparison of kernel-based dimensionality reduction methods

Baseline methods in our comparison include the famous classical techniques such as KPCA [6] and GDA [11], and recently developed ones like KANMM [10], and KMFA [11]. The average testing error of KNN for each FE method, as a function of the number of projected dimensions, is plotted in Fig. 1. As a reference, the best results of each algorithm, along with the corresponding number of features (the value in bracket) and the number of support patterns (the value in square bracket) used in K-Relief and SK-Relief, are reported in Table 2. From these experimental results, we arrive at the following observations:

1. K-Relief is a competitive feature extraction method. It has much lower computation complexity [2] yet performs comparably with other state-of-art methods in most cases.
2. The performance of SK-Relief is very similar to K-Relief. However, the number of support patterns has been significantly reduced. Compared with K-Relief, only around 10% of samples are used in SK-Relief, which means that most of the training patterns can be discarded after the classifier being trained. This is clearly an advantage of SK-Relief over K-Relief, especially when a large data set or an online mining task is faced.

Table 2. Comparison of kernel-based feature extraction methods. The value in each entry is the lowest average testing error, the number in () is the corresponding number of features, the value in [] denotes N_{sp} . The best results are highlighted in bold.

DATA	KPCA	GDA	KANMM	KMFA	KRELIEF	SKRELIEF
BREA	0.036(3)	0.039(1)	0.037(12)	0.034 (18)	0.034 (8)[400]	0.035(16)[48]
RING	0.035(8)	0.089(1)	0.018 (12)	0.080(12)	0.071(4)[1400]	0.079(9)[85]
PIMA	0.309(28)	0.310(1)	0.301 (46)	0.316(60)	0.324(48)[400]	0.323(58)[35]
ECOL	0.198(7)	0.255(7)	0.189 (8)	0.254(29)	0.231(18)[200]	0.231(15)[18]
GLAS	0.353(30)	0.351 (5)	0.373(15)	0.351 (16)	0.371(28)[120]	0.384(29)[21]
LRS	0.598 (22)	0.598 (46)	0.614(8)	0.613(30)	0.613(48)[380]	0.598 (60)[32]

6 Conclusion

In this paper, we provide a rigorous derivation for the K-Relief algorithm. To achieve sparse kernel-based nonlinear features, we assume the kernel bases form a causal series and incorporate a greedy online sparsification procedure into the basis construction process, leading to an SK-Relief algorithm. Gram-Schmidt process is in no way the only method to construct orthogonal bases in the kernel space. An obvious alternative is KPCA. However, besides such unsupervised approaches, there are various supervised approaches. Therefore, an interesting investigation would be to explore how these different basis construction methods affect the performance of K-Relief.

References

1. Baudat, G., Anouar, F.: Generalized Discriminant Analysis Using a Kernel Approach. *Neural Computation* 12, 2385–2404 (2000)
2. Cao, B., Shen, D., Sun, J.T., Yang, Q., Chen, Z.: Feature Selection in a Kernel Space. In: *Proceeding of 24th ICML* (2007)
3. Engel, Y., Mannor, S., Meir, R.: The Kernel Recursive Least Squares Algorithm. *IEEE Trans. Signal Processing* 52(8), 2275–2285 (2004)
4. Guyon, I., Elissee, A.: An Introduction to Variable and Feature Selection. *Journal of Machine Learning Research* 3, 1157–1182 (2003)
5. Kira, K., Rendell, L.A.: A Practical Approach to Feature Selection. In: *Proceeding of the 9th ICML*, pp. 249–256 (1992)
6. Scholköpf, B., Smola, A., Müller, K.R.: Nonlinear Component Analysis as a Kernel Eigenvalue Problem. *Neural Computation* 10(5), 1299–1319 (1998)
7. Sun, Y.J.: Iterative Relief for Feature Weighting: Algorithms, Theories, and Applications. *IEEE Trans. on PAMI* 29(6), 1035–1051 (2007)
8. Tipping, M.E.: Sparse Kernel Principal Component Analysis. In: *NIPS* (2001)
9. Torkkola, K.: Feature Extraction by Nonparametric Mutual Information Maximization. *Journal of Machine Learning Research* 3, 1415–1438 (2003)
10. Wang, F., Zhang, C.S.: Feature Extraction by Maximizing the Average Neighborhood Margin. In: *Proceeding of IEEE CVPR* (2007)
11. Yan, S.C., Xu, D., Zhang, B.Y., Zhang, H.J., Yang, Q., Lin, S.: Graph Embedding and Extensions: A General Framework for Dimensionality Reduction. *IEEE Trans. on PAMI* 29(1), 40–51 (2007)

Term Committee Based Event Identification within News Topics

Kuo Zhang, JuanZi Li, Gang Wu, and KeHong Wang

Tsinghua University
Beijing, 100084, China
zkuo99@mails.tsinghua.edu.cn,
{ljz,wug03,wkh}@keg.cs.tsinghua.edu.cn

Abstract. Most previous research focus on organizing news set into flat collections of stories. However, a topic in news is more than a mere collection of stories: it is characterized by a definite structure of inter-related events. Stories within a topic usually share some terms which are related to the topic other than a specific event, so stories of different events are usually very similar to each other within a topic. To deal with this problem, we propose a new event identification method based on the term committee. We first capture some tight term clusters as term committees of potential events, and then use them to re-weight the key terms in a story. The experimental results on two Linguistic Data Consortium (LDC) datasets show that the proposed method for event identification outperforms previous methods significantly.

1 Introduction

Topic Detection and Tracking (TDT) [1] has given the definitions of news Topic and news Event. A Topic is defined as “a seminal event or activity, along with directly related events and activities” [2]. An Event is defined as “something (non-trivial) happening in a certain place at a certain time” [3]. For instance, when an oil tanker sinks in an ocean, it is the seminal event which triggers the topic. Other events within the topic may include salvaging efforts, environmental damage and so on.

Nallapati et al. first presented the concepts of event identification within news topics [4]. In their work, cosine formula was used to compute news story similarity. Finally agglomerative clustering was employed to identify news events. However, the methods widely used in Topic Detection (e.g. agglomerative clustering) can hardly achieve satisfying accuracy in event identification. Based on analysis of data, we have two observations: (1) within the same topic, even two stories describing different events may have a good portion of overlapping terms (topic related); (2) each event usually only has a small number of key terms which are strongly related to the event other than the whole topic. Therefore document similarity contributed by event key terms is usually drowned out by the similarity contributed by topic related terms.

This paper aims at resolving the problems described above, and has the following contributions: (1) We define term committee to represent event key terms. (2) We propose a clustering based method to discover term committees of difference events

within a topic. We compute the similarity between terms according to the set of stories containing the terms at first. Then we discover a set of tight term clusters (high intra-group similarity), that are well scattered in the similarity space (low inter-group similarity), as term committees of potential events. (3) We propose to use the term committees to adjust story representation and similarity computing in event identification. The experimental results show that our proposed event identification method improves 16.7% in accuracy compared to the method used in paper [4].

2 Related Work

Yang [5] employed an agglomerative clustering algorithm named Group Average Clustering to identify events. Li [6] believed that news stories are always aroused by events; therefore, they proposed a probabilistic model to incorporate both content and time information in a unified framework. Gabriel [7] used some probabilistic models to identify burst features within a time window at the first step, then group the burst features and use them to determine the hot periods of the bursty events. Although these methods are called “event identification”, their concept of “event” is more like the concept of “topic” in TDT, bigger than the concept of “event” in TDT. For example, the methods can detect a topic “Winter Olympic Game 1998”, but they are not good at identifying and differentiating two events within a topic: “the open ceremony of Olympic Game” and “a hockey match in Olympic Game”.

Another related work is a new clustering algorithm named CBC (Clustering By Committee) [8]. Our method is different from CBC: the committee elements are terms (features) in our algorithm, while the committee elements are documents (samples) in CBC. And we use term committee to re-weight the representation of stories.

3 Problem Definition and Analysis

We use the same definition of event identification as paper [4]:

Event Identification (EI): event identification detects events within a news topic. Let $D(T)=\{d_1, d_2, \dots, d_n\}$ be the entire story set of a certain topic T , where n is the number of stories in topic T . Each news story d is represented by a 2-tuple (v, t) , where v is the document vector and t is the publication time. The results of event identification is $E(T)=\{e_1, e_2, \dots, e_m\}$, where each element is a set of stories describing the same event in topic T . And the elements have the following constraints:

- i $\forall i \quad e_i \subseteq D(T), e_i \neq \emptyset$
- ii $\forall i, j \quad i \neq j \rightarrow e_i \cap e_j = \emptyset$
- iii $\forall d_i \quad \exists e_j \in E(T) \text{ s.t. } d_i \in e_j$

Event identification is more challenging than traditional TDT tasks, because stories of different events are usually too similar to each other within a topic. We use the corpus of TDT2 from LDC to make an investigation about the similarities between stories from the same events or different events.

In table 1, we use *S-event* to represent the average similarity of all pairs of stories in the same events and use *D-event* to represent the average similarity of all pairs of

stories belonging to different stories. $S\text{-event}(de)$ and $D\text{-event}(de)$ denote the similarity obtained by using time decay according to the difference between two story's publication time. The time decay method is defined as follows:

$$sim_D(d, d') = sim(d, d')e^{-\frac{\alpha(d.t-d'.t)}{T}} \quad (1)$$

where $sim(d, d')$ is the cosine similarity of documents d and d' , $d.t$ means the publication time of story d , and T is the time difference between the earliest and the latest story in the given topic. α is the time decay factor and set to 1 here. From table 1, we can see that the difference between $S\text{-event}$ and $D\text{-event}$ is not significant in most topics. Even when time decay is used, the difference between $S\text{-event}(de)$ and $D\text{-event}(de)$ is still not significant.

Table 1. Average story similarities in the same and different events for some topics in TDT2

TopicID	$S\text{-event}$	$D\text{-event}$	$D\text{-event}/S\text{-event}$	$S\text{-event}(de)$	$D\text{-event}(de)$	$D\text{-event}(de)/S\text{-event}(de)$
20012	0.250	0.240	96.00%	0.220	0.179	81.36%
20022	0.345	0.310	89.86%	0.283	0.236	83.39%
20026	0.355	0.315	88.73%	0.294	0.249	84.69%
20033	0.183	0.125	68.31%	0.143	0.094	65.73%
20056	0.233	0.171	73.39%	0.206	0.137	66.50%
20077	0.268	0.195	72.76%	0.243	0.147	60.49%
20087	0.238	0.188	78.99%	0.202	0.123	60.89%
Average	0.267	0.220	82.48%	0.227	0.166	73.22%

In Table 2, we give the average story similarities of all pairs of stories in the same topics and different topics. From the statistics, we can see that, stories in different topics tend to have low similarities. Obviously, traditional method for topic detection is not suitable for event identification.

Table 2. Average story similarities in the same and different topics in TDT2 and TDT3 corpus

	$S\text{-topic}$	$D\text{-topic}$	$D\text{-topic}/S\text{-topic}$	Topic-number
TDT2	0.1927	0.0685	35.55%	28
TDT3	0.2082	0.0793	38.09%	25

By analyzing the data, we have classified the terms into three classes:

- i. *Term class A (non-key terms)*: terms that are not strongly related to the topic nor a specific event. Terms of this class should be given low weights.
- ii. *Term class B (topic key terms)*: terms that occur frequently in the whole topic. They are strongly related to the topic other than a specific event. Obviously, terms of this class should be assigned low weights too.
- iii. *Term class C (event key terms)*: terms that occur frequently in an event and infrequently in other events. They are strongly related to a specific event. Through data analysis, we found an event usually only have a small number of event key terms.

To decrease the similarity contributed by term class *A* and *B*, we have to find event key terms and increase their weights. A term committee is defined as a set of key terms (term class *C*) of the corresponding event. Term committees are captured at first and then used for later event identification steps.

4 Our Approach

In this section, we describe our approach to event identification. Our event identification method consists of three phases. At the first phase, we preprocess the news stories and generate the vector representation for each news story. At the second phase, we discover a set of tight term clusters (high intra-group similarity), called term committees, that are well scattered in the similarity space (low inter-group similarity). At the third phase, we use the term committees to help re-weight key terms in stories, and agglomerative clustering is used for events identification. The details of this approach are given by subsequent subsections.

4.1 News Story Representation

Preprocessing is needed before generating story representation. For preprocessing, we tokenize words, recognize abbreviations, normalize abbreviations, and remove stop-words, replace words with their stems using K-stem algorithm [9], and then generate word vector for each news story.

Thus, each story *d* is represented as follows:

$$d \rightarrow (v, t) \rightarrow (\{weight(d, w_1), weight(d, w_2), \dots, weight(d, w_n)\}, t)$$

where *t* is the publication time of news story *d*, and *n* means the number of distinct terms in story *d*. And *weight(d, w)* means the weight of term *w* in story *d*:

$$weight(d, w) = \frac{\log(tf(d, w) + 1) * \log((N + 1) / (df(w) + 0.5))}{\sum_{w' \in d} \log(tf(d, w') + 1) * \log((N + 1) / (df(w') + 0.5))} \tag{2}$$

where *N* means the total number of news stories, and *tf(d, w)* means how many times term *w* occurs in news story *d*. And *df(w)* is the number of stories containing term *w*.

4.2 Term Committee Discovery

For each term *w*, we create a story set at first:

$$F(w) = \{d \mid d \text{ contains term } w\}$$

The similarity of two terms *w_i* and *w_j* is defined as follows:

$$sim(w_i, w_j) = \frac{|F(w_i) \cap F(w_j)|}{\sqrt{|F(w_i)| \times |F(w_j)|}} \tag{3}$$

The details of term committee discovery algorithm are presented in Figure 1.

Input: Term set $W = \{w \mid F(w) > 1\}$, threshold θ_1

Step 1: Put all terms in W into a list of residues R .

Step 2: Let C be a list of committees, and set $C = \emptyset$.

Step 3: Cluster the terms in R using average-link clustering.
 For each term cluster c created in the clustering process
 Compute the following score: $\log(|c|) \times \text{avgsim}(c)$, where $|c|$ is the number of terms in c and $\text{avgsim}(c)$ is the average pairwise similarity between terms in c .

Step 4: If cluster c has the highest score, and c 's similarity to each committee previously added to C is below the threshold θ_1 , then add c to C .

Step 5: Remove all the terms in c from residues R .
 Remove all the terms in R whose similarity with c is above threshold θ_1 .

Step 6: If R is empty, then we are done and return C . Otherwise, go to step 3.

Output: A list of term committees C

Fig. 1. Term committee discovery algorithm

Because the number of terms that only appear in a single news story is very large, and these terms can not provide useful information for event identification, we just use the terms appear in at least two stories as the input of this phase. At step 1, we put all the input terms into a term set R . And then initiate a term committee set C as an empty set. At step 3, we use agglomerative clustering algorithm to cluster the terms in R until all the terms belong to a single cluster, and we compute the score $\log(|c|) \times \text{avgsim}(c)$ for each temporary cluster c created during the clustering procedure. The factor $\log(|c|)$ reflects a preference for bigger term sets, while the factor $\text{avgsim}(c)$ reflects a preference for tighter term sets. Step 4 selects the cluster c with the highest score at first. If the similarity between cluster c and each previous term committee in C is smaller than threshold θ_1 , then add c to C . Step 5 remove all the terms in c from residues R , and remove all the terms in R whose similarity with c is above threshold θ_1 . If R is empty then return term committee set C as result, otherwise the algorithm jumps to step 3.

4.3 Event Identification

To reduce the similarity contributed by term class A and B , we use term committees for potential events to re-weight terms in similarity calculation. For two stories d and d' , their similarity is defined as follows:

$$\text{sim}(d, d') = \max_{c \in C} (\text{sim}(d, d', c)) \quad (4)$$

where C is the term committee set of the current topic obtained in the last phase, and

$$\text{sim}(d, d', c) = \frac{\sum_{w \in d, d'} \text{weight}(d, w, c) * \text{weight}(d', w, c)}{|d_c| \cdot |d'_c|},$$

$$d_c \rightarrow (v, t) \rightarrow (\{\text{weight}(d, w_1, c), \text{weight}(d, w_2, c), \dots, \text{weight}(d, w_n, c)\}, t),$$

$$\text{weight}(d, w, c) = \begin{cases} \text{weight}(d, w) * \gamma, & \text{when } w \in c \\ \text{weight}(d, w), & \text{when } w \notin c \end{cases}$$

Here γ is an empirical parameter (bigger than 1). Terms in a committee c are the key terms of the corresponding potential event. Therefore, the more overlapping terms two stories have in the same committee c , the more probable that the two stories belong to the same event.

At last, we also use agglomerative clustering method for event identification. Similarity between two clusters cl_1 and cl_2 is computed using average link:

$$sim(cl_1, cl_2) = \max_{c \in C} \left(\frac{\sum_{d \in cl_1} \sum_{d' \in cl_2} sim(d, d', c)}{|cl_1| \cdot |cl_2|} \right) \quad (5)$$

where $|cl|$ is the number of stories in cluster cl .

5 Experiments

5.1 Datasets

The datasets include 28 topics selected from TDT2 corpus, and 25 topics selected from TDT3 corpus [10]. Nallapati et al. annotated event membership for each selected story, and then created a training set of 26 topics and a test set of 27 topics by merging the 28 topics from TDT2 and 25 from TDT3 and then splitting them randomly. Table 3 shows some statistics for the training and test datasets. There are more details about the annotation spec in paper [4].

Table 3. Statistics of the training and test datasets

<i>Features</i>	<i>Training set</i>	<i>Test set</i>
Number of Topics	26	27
Average Number of Stories/Topic	28.69	26.74
Average Story Length	64.6	64.04
Average Number of Stories/Event	5.65	6.22
Average Number of Events/Topic	5.07	4.29

5.2 Evaluation Metric

We use the same evaluation metrics as paper [4]. For an automatically generated event model $M'=(\mathbf{E}',\Psi')$ and a true event model (annotated by human) $M=(\mathbf{E},\Psi)$, we examine a pair of stories at a time and verify whether the generated model and the true model agree on their event membership. The related metrics and definitions are given in detail as follows:

- **Event pairs $C(M)$:** This set includes all the unordered pairs (d_i, d_j) of stories d_i and d_j that are belong to the same event given a model M . Formally,

$$C(M) = \{(d_i, d_j) \mid d_i, d_j \in \mathbf{D} \wedge \exists e \in \mathbf{E} \text{ s.t. } (d_i \in e \wedge d_j \in e)\} \quad (6)$$

- **Event Precision EP :** this is the probability that a pair of two randomly selected stories belongs to set $C(M)$ given that it belongs to $C(M')$.

$$EP = |C(M) \cap C(M')| / C(M') \quad (7)$$

- **Event Recall ER:** this is the probability that a pair of two randomly selected stories belongs to set $C(M')$ given that it belongs to $C(M)$.

$$ER = |C(M) \cap C(M')| / C(M) \quad (8)$$

And the well known F1-measure is used to combine the above measures:

$$EF = 2 \cdot EP \cdot ER / (EP + ER) \quad (9)$$

5.3 Experimental Results

We implemented and tested four systems. SYSTEM 1 and SYSTEM 2 are two previous systems with which we want to compare. SYSTEM-3 is based on topic-specific stopword removal and SYSTEM-4 is based on our approach.

SYSTEM-1: This system uses cosine distance as the similarity of stories, and employs agglomerative clustering based on average-link to identify events. This system is used as baseline system. [4]

SYSTEM-2: This system is the same as SYSTEM-1, except that it uses formula (1) to adjust similarities according to time difference between news stories. [4]

SYSTEM-3: this system is based on topic-specific stopword removal. The idea was firstly presented in paper [11]. Since topical common terms (term class A) cause events in the same topic to be mutually confusing, a natural choice is to remove those terms. We obtained a stopword list for each topic by thresholding on the training set document frequency of a term t in T_i :

$$\frac{n(t, T_i)}{n(T_i)} > \beta \quad (10)$$

where $n(T_i)$ is the number of documents on topic T_i ; $n(t, T_i)$ is the number of documents containing term t and on topic T_i ; and parameter β is empirically chosen on training set. The story similarities are also adjusted by time difference the same as SYSTEM-2.

SYSTEM-4: This system is implemented based on our approach. It has three phases: story preprocessing, term committee discovery, event identification.

The results of the four systems on training set and test set are shown in Table 4 and Table 5 respectively. Each value in the tables is the average score over all topics. P-value that is marked by * means that it is a statistical significant improvement over the compared system (95% confidence level, one tailed T-test). The results of SYSTEM-1 and SYSTEM-2 listed in these tables are obtained from paper [4].

Table 4. Comparison of event detection algorithms (on training set)

<i>System</i>	<i>EP</i>	<i>ER</i>	<i>EF</i>	<i>P-value/Improvement</i>
SYSTEM-1[4]	0.39	0.67	0.46	-
SYSTEM-2[4]	0.45	0.70	0.53	2.9e-4*/+15.2%(to SYSTEM-1)
SYSTEM-3	0.44	0.73	0.54	-
SYSTEM-4	0.48	0.78	0.62	1.5e-3*/+17.0%(to SYSTEM-2)

Table 5. Comparison of event detection algorithms (on test set)

<i>System</i>	<i>EP</i>	<i>ER</i>	<i>EF</i>	<i>P-value/Improvement</i>
SYSTEM-1[4]	0.44	0.67	0.50	-
SYSTEM-2[4]	0.48	0.70	0.54	0.014*/+8.0% (to SYSTEM-1)
SYSTEM-3	0.47	0.71	0.52	-
SYSTEM-4	0.56	0.77	0.63	4.1e-3*/+16.7% (to SYSTEM-2)

For SYSTEM-3, we tested it on the test set with the optimal parameter β obtained from training set. For SYSTEM-4, we tested our method on the test set with the two parameters θ_1 and γ fixed at their optimal values learned from training set.

When tested on training set, SYSTEM-3 is slightly better than SYSTEM-2. However, it is even worse than SYSTEM-2 on test set. By analyzing the cases of SYSTEM-3, we found that there are two main reasons: (1) the parameter β is hard to determine. If β is set to a relative small value, some terms of class *C* (related to a big event) may be removed falsely. Otherwise, if β is set to a relative great value, most of the terms of class *B* cannot be removed. (2) when some terms of class *B* and *C* are removed, terms of class *A* will get heavier weights and make more noise.

The *EF* result of SYSTEM-4 on training set is 0.62 which is 34.8% higher than SYSTEM-1, and 17.0% higher than SYSTEM-2. On the test set, SYSTEM-4's *EF* value is 26.0% higher than SYSTEM-1 and 16.7% higher than SYSTEM-2. On both training and test sets, SYSTEM-4 shows statistically significant improvement compared to SYSTEM-2 which performs the best in paper [4].

6 Conclusions

Most of the previous work, such as TDT, organizes news stories by topics which are flat collections of news stories. However, this paper aims to model events within a news topic. Due to the high similarity between stories of different events within the same topic (usually stories within a topic share lots of terms about the topic), we proposed an event identification method based on term committee. We first capture some tight term clusters as term committees of potential events, and then use them to re-weight key terms in a story. The experimental results show that our approach for event identification has significant improvement over previous methods.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant No. 90604025 and the National Basic Research Program of China (973 Program) under Grant No. 2007CB310803.

References

- [1] <http://www.nist.gov/speech/tests/tdt/index.htm>
- [2] In: Topic Detection and Tracking. Event-based Information Organization. Kluwer Academic Publishers, Dordrecht (2002)

- [3] Yang, Y., Carbonell, J., Brown, R., Pierce, T., Archibald, B.T., Liu, X.: Learning Approaches for Detecting and Tracking News Events. *IEEE Intelligent Systems Special Issue on Applications of Intelligent Information Retrieval* 14(4), 32–43 (1999)
- [4] Nallapati, R., Feng, A., Peng, F., Allan, J.: Event Threading within News Topics. In: *CIKM 2004*, Washington, DC, USA, November 8–13, pp. 446–453 (2004)
- [5] Yang, Y., Pierce, T., Carbonell, J.: A Study on Retrospective and On-line Event Detection. In: *Proceedings of SIGIR 1998*, Melbourne, Australia, pp. 28–36 (1998)
- [6] Li, Z., Wang, B., Li, M., Ma, W.: A Probabilistic Model for Retrospective News Event Detection. In: *Proceedings of ACM SIGIR 2005*, pp. 61–81 (2005)
- [7] Fung, G., Yu, J., Yu, P., Lu, H.: Parameter Free Bursty Events Detection in Text Streams. In: *Proceedings of the 31st VLDB Conference*, Trondheim, Norway, pp. 181–192 (2005)
- [8] Pantel, P., Lin, D.: Document Clustering with Committees. In: *Proceedings of the 25th Annual International ACM SIGIR Conference*, Tampere, Finland, pp. 199–206 (2002)
- [9] Krovetz, R.: Viewing Morphology as An Inference Process. In: *Proceedings of ACM SIGIR 1993*, pp. 61–81 (1993)
- [10] The linguistic data consortium, <http://www.ldc.upenn.edu/>
- [11] Yang, Y., Zhang, J., Carbonell, J., Jin, C.: Topic-conditioned Novelty Detection. In: *Proceedings of the 8th ACM SIGKDD International Conference*, pp. 688–693. ACM Press, New York (2002)

Locally Linear Online Mapping for Mining Low-Dimensional Data Manifolds

Huicheng Zheng¹, Wei Shen¹, Qionghai Dai^{1,2}, and Sanqing Hu³

¹ Media Processing and Communication Lab, Department of Electronics and Communication Engineering, Sun Yat-sen University, 510275 Guangzhou, China
{zhenghch, issshenw}@mail.sysu.edu.cn

² Department of Automation, Tsinghua University, 100084 Beijing, China
qhdai@tsinghua.edu.cn

³ Department of Neurology, Division of Epilepsy and Electroencephalography, Mayo Clinic, 200 First Street SW, Rochester, MN 55905, USA
Hu.Sanqing@mayo.edu

Abstract. This paper proposes an online-learning neural model which maps nonlinear data structures onto mixtures of low-dimensional linear manifolds. Thanks to a new distortion measure, our model avoids confusion between local sub-models common in other similar networks. There is no local extremum for learning at each neuron. Mixtures of local models are achieved by competitive and cooperative learning under a self-organizing framework. Experiments show that the proposed model is better adapted to various nonlinear data distributions than other models in comparison. We further show a successful application of this model to discovering low-dimensional manifolds of handwritten digit images for recognition.

1 Introduction

Previous research has demonstrated that image variations of many objects can be effectively modeled by low-dimensional manifolds embedded in the high-dimensional data space [1, 2]. The embedded manifolds are generally nonlinear and can be approximated by mixtures of linear models [2, 3]. Some methods have been proposed to coordinate pre-learned mixture models [4] or propagate pre-learned manifolds to new data points [5]. However, a universal coordinate system may not be applicable when separate manifolds are involved.

State-of-the-art mixture models usually treat the input data in a batch mode, and estimate model parameters by using the expectation-maximization (EM) algorithm. However, when the input data are not all available at once, only online algorithms can be used, which also have advantages including adaptability, low storage requirement, and fast training. The adaptive-subspace self-organizing map (ASSOM) proposed by Kohonen [6] is able to discover online an ordered set of linear subspaces. It extends the self-organizing feature map (SOFM) [7] by replacing the single weight vector at each neuron with a set of basis vectors spanning a subspace. However, the ASSOM may not be adequate when data distributions are deviated from the origin. As an improvement,

the adaptive-manifold self-organizing map (AMSOM) proposed by Liu [8] learns mean vectors of local models in addition to the conventional subspaces. The linear manifold self-organizing map (LMSOM) proposed by Zheng *et al.* [9] learns the offsets of local models by using a stochastic gradient-descent algorithm. Nonetheless, since local subspaces extend infinitely in the data space, confusion between local models is common in these mixture models, which is harmful for dimension reduction and data compression because the variance observed by local models could be larger than necessary and data are more likely to be assigned to the wrong prototypes.

In order to achieve truly local representation, in this paper, we propose a mixture model which implements a new distortion measure between the input data and local linear manifolds. The objective function at each neuron has not local extremum, and the mean vector and the local principal subspace is the only global minimum solution. Adaptive online learning of local linear manifolds is then achieved through a stochastic gradient-descent algorithm. Mixture of local models is realized through competitive and cooperative learning. We demonstrate by experiments that: 1) our model largely alleviates ambiguity of local linear manifolds; 2) it can be effectively used to map highly nonlinear manifolds, including separate manifolds. As a potential real-world application, this model is also used to discover low-dimensional manifolds in handwritten digit images and shows promising results.

2 The Locally Linear Online Mapping Algorithm

Let \mathbf{x} be the input, \mathcal{L} be an H -dimensional local linear manifold with an offset vector \mathbf{m} and *orthonormal* basis vectors $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H\}$. The proposed distortion measure is:

$$e(\mathbf{x}, \mathcal{L}) = \|\tilde{\mathbf{x}}_{\mathcal{L}}\|^2 + \alpha \|\hat{\mathbf{x}}_{\mathcal{L}}\|^2, \quad (1)$$

$$\hat{\mathbf{x}}_{\mathcal{L}} = \sum_{h=1}^H \left((\mathbf{x} - \mathbf{m})^T \mathbf{b}_h \right) \mathbf{b}_h, \quad \tilde{\mathbf{x}}_{\mathcal{L}} = \mathbf{x} - \mathbf{m} - \hat{\mathbf{x}}_{\mathcal{L}}, \quad (2)$$

where $\hat{\mathbf{x}}_{\mathcal{L}}$ and $\tilde{\mathbf{x}}_{\mathcal{L}}$ are respectively the projection of \mathbf{x} on \mathcal{L} and the residual of projection. Compared to the usual reconstruction error, equation (1) has the extra $\alpha \|\hat{\mathbf{x}}_{\mathcal{L}}\|^2$, where $0 \leq \alpha \leq 1$ is a parameter which shall be shown to control the desired extension of the local model. From the probabilistic point of view, if we define a Gaussian likelihood $p(\mathbf{x} | \mathbf{m}, \mathbf{B}) \propto \exp(-e(\mathbf{x}, \mathcal{L}))$, where \propto represents equivalence up to a normalization constant, $\mathbf{B} = [\mathbf{b}_1 \ \mathbf{b}_2 \ \dots \ \mathbf{b}_H]$, then reducing $e(\mathbf{x}, \mathcal{L})$ would be equivalent to increasing the data likelihood. This would eventually correspond to two conditionally independent Gaussian distributions according to equation (1), one in \mathcal{L} and the other one in the residual subspace orthogonal to \mathcal{L} .

Now let us investigate the equi-distortion surface defined by $e(\mathbf{x}, \mathcal{L}) = C^2$ for $C > 0$. We consider 1-D linear manifold \mathcal{L} in the 2-D space first. When $\alpha = 0$, $e(\mathbf{x}, \mathcal{L})$ is the usual reconstruction error, and the equi-distortion surface is the two infinitely extending lines l_1 and l_2 in Fig. 1. When $0 < \alpha < 1$, the equi-distortion surface turns into an ellipse as shown in Fig. 1, whose major axis lies on \mathcal{L} with length $2C/\sqrt{\alpha}$, and whose minor axis is perpendicular to \mathcal{L} with length $2C$. The larger α is, the shorter the major

axis will be, and therefore the more local the representation along the linear manifold \mathcal{L} will be. When $\alpha = 1$, $e(\mathbf{x}, \mathcal{L})$ degenerates to the usual Euclidean distance from \mathbf{x} to the prototype \mathbf{m} , and the ellipse in Fig. 1 degenerates to a circle centered at \mathbf{m} . The discussion can be easily generalized to higher dimensional cases.

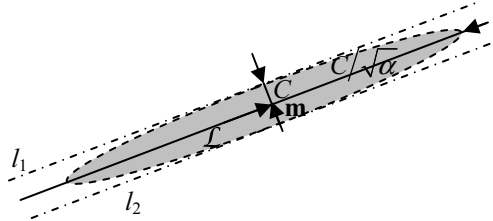


Fig. 1. Equi-distortion surface of the proposed measure function $e(\mathbf{x}, \mathcal{L})$

Let $\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(N)}$ be N samples available to a neuron for learning. The objective of learning at the neuron is to minimize the average distortion measure

$$J(\mathbf{m}, \mathbf{B}) = \frac{1}{N} \sum_{i=1}^N e(\mathbf{x}^{(i)}, \mathcal{L}) \tag{3}$$

with respect to \mathbf{m} and \mathbf{B} . For $0 < \alpha < 1$, the linear manifold \mathcal{L} minimizing $J(\mathbf{m}, \mathbf{B})$ would satisfy: 1) \mathbf{m} is the average of the samples; 2) the basis vectors $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H\}$ span the principal subspace of the samples. This is because $J(\mathbf{m}, \mathbf{B})$ is also the implied objective function of principal component analysis (PCA):

$$\frac{1}{N} \sum_{i=1}^N e(\mathbf{x}^{(i)}, \mathcal{L}) = \frac{\alpha}{N} \sum_{i=1}^N \|\mathbf{x}^{(i)} - \mathbf{m}\|^2 + \frac{1-\alpha}{N} \sum_{i=1}^N \|\tilde{\mathbf{x}}_{\mathcal{L}}^{(i)}\|^2. \tag{4}$$

PCA minimizes $\sum_{i=1}^N \|\mathbf{x}^{(i)} - \mathbf{m}\|^2$ by setting \mathbf{m} to be the average of the samples, and minimizes $\sum_{i=1}^N \|\tilde{\mathbf{x}}_{\mathcal{L}}^{(i)}\|^2$ by setting $\{\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_H\}$ to be the principal eigenvectors of the sample covariance matrix. It can be proved that all the stationary points of $J(\mathbf{m}, \mathbf{B})$ except the global minimum are saddle points, i.e. $J(\mathbf{m}, \mathbf{B})$ has no local extremum. Due to limited space, rigorous proofs will be presented elsewhere.

We use Robbins-Monro stochastic approximation [10] to optimize $J(\mathbf{m}, \mathbf{B})$. Compared to their deterministic counterparts, stochastic algorithms have the advantages of higher computing efficiency at lower memory cost, better suitability to sequentially acquired data, and stronger robustness to local extrema or saddle points due to “noisy” presentation of input samples. Therefore, we aim to minimize a “sample” objective function $J^{(t)}(\mathbf{m}, \mathbf{B})$ of the input $\mathbf{x}^{(t)}$ at instant t through stochastic gradient descent while maintaining orthonormality of $\{\mathbf{b}_h; h = 1, 2, \dots, H\}$

$$J^{(t)}(\mathbf{m}, \mathbf{B}) = e(\mathbf{x}^{(t)}, \mathcal{L}) = \|\tilde{\mathbf{x}}_{\mathcal{L}}^{(t)}\|^2 + \alpha \|\hat{\mathbf{x}}_{\mathcal{L}}^{(t)}\|^2. \tag{5}$$

$$\frac{\partial J^{(t)}(\mathbf{m}, \mathbf{B})}{\partial \mathbf{m}} = -2\tilde{\mathbf{x}}_{\mathcal{L}}^{(t)} - 2\alpha\hat{\mathbf{x}}_{\mathcal{L}}^{(t)}, \tag{6}$$

$$\frac{\partial J^{(t)}(\mathbf{m}, \mathbf{B})}{\partial \mathbf{b}_h} = -2(1-\alpha) \left((\mathbf{x}^{(t)} - \mathbf{m})^T \mathbf{b}_h \right) \tilde{\mathbf{x}}_{\mathcal{L}}^{(t)}. \tag{7}$$

The stepwise updating rules of \mathbf{m} and \mathbf{b}_h are as follows:

$$\mathbf{m}^{(t+1)} = \mathbf{m}^{(t)} + \lambda(t) \tilde{\mathbf{x}}_{\mathcal{L}}^{(t)} + \alpha \lambda(t) \hat{\mathbf{x}}_{\mathcal{L}}^{(t)}, \tag{8}$$

$$\mathbf{b}_h^{(t+1)} = \mathbf{b}_h^{(t)} + (1-\alpha) \lambda(t) \left((\mathbf{x}^{(t)} - \mathbf{m}^{(t)})^T \mathbf{b}_h^{(t)} \right) \tilde{\mathbf{x}}_{\mathcal{L}}^{(t)}. \tag{9}$$

where $\lambda(t) > 0$ is a learning-rate parameter and should satisfy $\sum_{t=0}^{\infty} \lambda(t) = \infty$ and $\sum_{t=0}^{\infty} \lambda^2(t) < \infty$ for convergence of the algorithm. The $\hat{\mathbf{x}}_{\mathcal{L}}^{(t)}$ term in equation (8) can be regarded as a force moving \mathbf{m} along the linear manifold \mathcal{L} towards the projection of \mathbf{x} . Its average effect in the stochastic process is to find the mean vector \mathbf{m} . When $\alpha = 0$, the $\hat{\mathbf{x}}_{\mathcal{L}}^{(t)}$ term diminishes, and \mathbf{m} is no more guaranteed to move to the center, which is the case of the LMSOM [9]. The basis vectors should be orthonormalized after equations (8) and (9) have been applied. The solution obtained through stochastic optimization converges in probability to the optimal solution.

For mixture of local models we use a self-organizing network. Such networks are biologically plausible in the sense that, as a learning system, our cerebral cortex has also been observed to consist of self-organized areas each being specialized in certain tasks. Self-organizing maps have been extensively and successfully applied to various areas of machine learning [7]. In particular, our model first partitions the data space to a number of regions corresponding to the neurons, and then learns local models at neurons. The partition and learning is performed online for the input at each time instant in a competitive and cooperative way. For the input sample $\mathbf{x}^{(t)}$ at the t -th instant, the network determines the winning neuron c via competition:

$$c = \arg \min_{q \in Q} e(\mathbf{x}^{(t)}, \mathcal{L}_q), \tag{10}$$

where Q is the set of neurons and \mathcal{L}_q the linear manifold of the q -th neuron.

The winner and the neurons in its neighborhood then update their local linear manifolds following the previously developed stochastic optimization algorithm. The updating “force” of the neighboring neurons of the winner is attenuated by a neighborhood function $v^{(t)}(c, q)$, which is a decreasing function of the distance between neurons c and q . $v^{(t)}(c, q) = 1$ when $q = c$. For $q \neq c$, $v^{(t)}(c, q)$ should also be a decreasing function of the learning step t , and $v^{(t)}(c, q) \rightarrow 0$ when $t \rightarrow \infty$. So at the final stage of learning, each neuron only learns the data in its partition. The updating formulae for each local model $q \in Q$ are:

$$\mathbf{m}_q^{(t+1)} = \mathbf{m}_q^{(t)} + v^{(t)}(c, q) \lambda(t) \left(\tilde{\mathbf{x}}_{\mathcal{L}_q}^{(t)} + \alpha \hat{\mathbf{x}}_{\mathcal{L}_q}^{(t)} \right), \tag{11}$$

$$\mathbf{b}_{qh}^{(t+1)} = \mathbf{b}_{qh}^{(t)} + (1-\alpha) v^{(t)}(c, q) \lambda(t) \left((\mathbf{x}^{(t)} - \mathbf{m}_q^{(t)})^T \mathbf{b}_{qh}^{(t)} \right) \tilde{\mathbf{x}}_{\mathcal{L}_q}^{(t)}, \tag{12}$$

where \mathbf{m}_q is the mean vector of the q -th neuron and \mathbf{b}_{qh} the h -th basis vector of the q -th neuron, for $h = 1, 2, \dots, H$.

The locally linear online mapping (LLOM) algorithm is summarized as follows:

1. Initialize the parameters (\mathbf{m} , \mathbf{B}) of all the neurons in the network (e.g. in a random way), with the columns of \mathbf{B} being orthonormal;
2. For the input sample $\mathbf{x}^{(t)}$ at instant t , the winner c of the neurons is determined according to equation (10);
3. Update the local linear manifold \mathcal{L}_q at each neuron $q \in Q$ according to equations (11) and (12) and then orthonormalize the basis vectors;
4. Repeat steps 2 and 3 until certain stop criterion is met, e.g. when some predetermined maximum learning steps have been met (when there are not sufficient samples, the bag of samples should be repeatedly presented).

3 Experimental Results on Manifold Mining

Different networks are trained to learn three 1-D linear manifolds corresponding to the three clusters in Fig. 2. Each cluster contains 600 samples. Each network has three neurons. The number of learning steps $T = 10,000$. The neighborhood function is Gaussian. The learning-rate parameter is $\lambda(t) = \lambda_0 T / (T + 99t)$ with $\lambda_0 = 1$. The specific value of α is not crucial when it is in the range $(0, 1)$ according to our experiments, and $\alpha = 0.1$ is used here. As shown in Fig. 2, the ASSOM can not learn these clusters sufficiently. The mean vectors learned by the AMSOM have been “attracted” away from the cluster centers due to confusion between the local models. The orientations of the linear manifolds learned by the LMSOM have evident deviation from the local principal directions of the clusters. The LLOM has shown the best result.

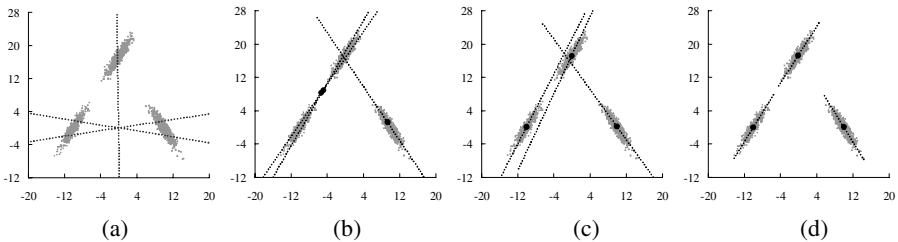


Fig. 2. Learning multiple linear manifolds with (a) the ASSOM, (b) the AMSOM, (c) the LMSOM, and (d) the LLOM. Large black dots represent the learned mean vectors (the ASSOM does not learn mean vectors). Dotted lines represent the learned 1-D linear manifolds (shorter for the LLOM to emphasize local representations).

We have also compared the classification accuracies of the different networks. Each neuron of a network is labeled with the cluster whose samples the neuron wins the most during the training phase. In the test phase, each input sample compares its label to the label of the winning neuron. The training and test were repeated 20 times. Each time all the data were regenerated and all the networks re-initialized with different random seeds. Table 1 summarizes the mean values and the standard deviations of the accuracies for different networks. The LLOM shows obviously better performance than the

other networks. We remark that while the linear discriminant analysis (LDA) may also correctly separate these clusters, it does not give a faithful representation of the data as shown in Fig. 2(d).

Table 1. Classification accuracies of the three clusters in Fig. 2 by using different networks. In the table, TR denotes the training set and TS the test set. μ is the mean value of accuracies and σ the standard deviation.

	ASSOM		AMSOM		LMSOM		LLOM	
	TR	TS	TR	TS	TR	TS	TR	TS
μ	67.43%	66.96%	66.71%	66.46%	86.71%	86.92%	99.997%	99.997%
σ	6.5×10^{-3}	8.2×10^{-3}	1.5×10^{-2}	1.3×10^{-2}	5.4×10^{-2}	5.3×10^{-2}	1.2×10^{-4}	1.2×10^{-4}

Figure 3 shows the results of mapping a 2-D spiral of 1,000 samples onto mixtures of 1-D linear manifolds by using the different networks. Each network contains 15 neurons. $T = 20,000$ and $\alpha = 0.5$ are used. While a smaller α corresponds to more extending linear manifolds and therefore could provide stronger data generalization, according to our experiments, a larger α such as the one used here seems necessary for a more local learning of highly nonlinear manifolds. The other parameters are defined

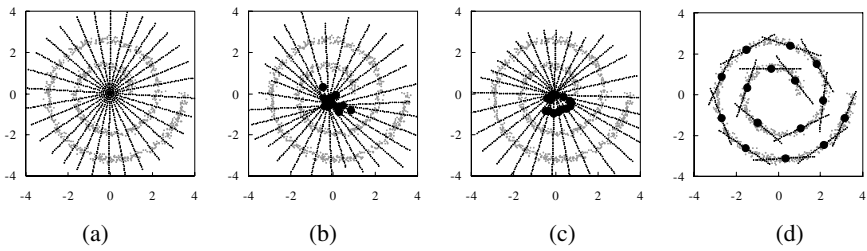


Fig. 3. Mapping a spiral by using (a) the ASSOM, (b) the AMSOM, (c) the LMSOM, and (d) the LLOM

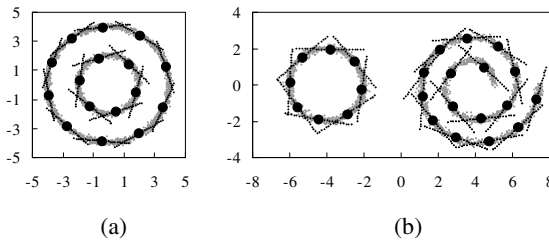


Fig. 4. Mapping separate nonlinear manifolds by using the LLOM

in the same way as in the previous experiment. The LLOM shows obviously the best performance thanks to the local property of neurons. It can also map separate nonlinear

manifolds as shown in Fig. 4, where both of the distributions contain 30,000 samples. Note that many nonlinear dimensionality reduction algorithms, e.g. the locally linear coordination (LLC) algorithm [4], do not work in such cases since a universal low-dimensional coordinate system does not exist.

We have also applied the LLOM to mining low-dimensional manifolds embedded in the MNIST handwritten digit images [11] for recognition. The MNIST database contains 60,000 training images and 10,000 test images. Our system consists of ten LLOM networks \mathcal{N}_k , $k = 0, 1, \dots, 9$, each containing $n \times n$, $n = 3, 4, \dots, 8$ neurons under a rectangular topology. The dimension of local linear manifolds is $H = 1, 2, 3$. The neighborhood function and the learning-rate parameter are defined in the same way as in previous experiments. $T = 30,000$ and $\alpha = 0.1$. For each input digit image \mathbf{x} , we construct an “episode” composed of three images $\mathbf{x}(s)$, $s \in S$ ($|S| = 3$): the original image, the image variant rotated -10° , and the image variant rotated 10° .

For $n = 3$ and $H = 2$, each network has learned 3×3 2-D linear manifolds (defined by the mean vectors \mathbf{m}_{kq} , basis vectors \mathbf{b}_{kq1} and \mathbf{b}_{kq2}) embedded in the high-dimensional distributions of handwritten digit images, as shown in Fig. 5. In this way, the LLOM also helps to visualize the underlying low-dimensional structures.

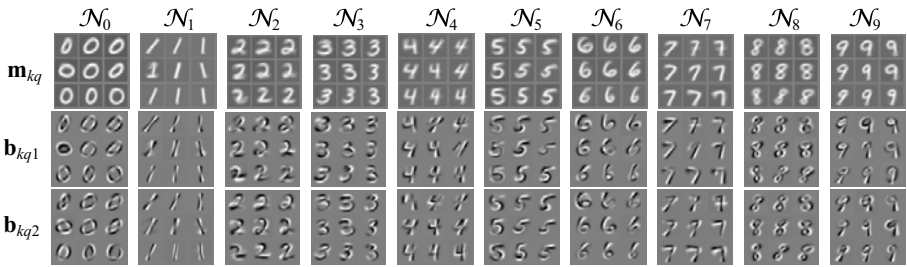


Fig. 5. Handwritten digit image manifolds learned by the LLOM

For recognition of an input image \mathbf{x} , each network \mathcal{N}_k builds a reconstructed image $\hat{\mathbf{x}}_{\mathcal{N}_k}(s)$ for each variant $\mathbf{x}(s)$ of \mathbf{x} . The network with the minimum average error determines the class of \mathbf{x} . $\hat{\mathbf{x}}_{\mathcal{N}_k}(s)$ is reconstructed from all the projections of $\mathbf{x}(s)$ on local models of \mathcal{N}_k [9]. The recognition accuracy of the LLOM networks on the MNIST database is summarized in Fig. 6. The accuracy increases with the number of neurons and the dimension of the local linear manifolds, which reaches 98.26% on the training set and 97.74% on the test set. For a comparison, the system based on the LMSOM achieved 98% on the training set and 97.3% on the test set [9].

Although the accuracy of handwritten digit recognition could be further improved by using more specific methods to character recognition [11], in this paper our interest is not limited to the specific task, but in a general application of the LLOM to automatic feature extraction through manifold mining. The LLOM has the potential to be applied to many other areas where nonlinear dimensionality reduction is desirable.

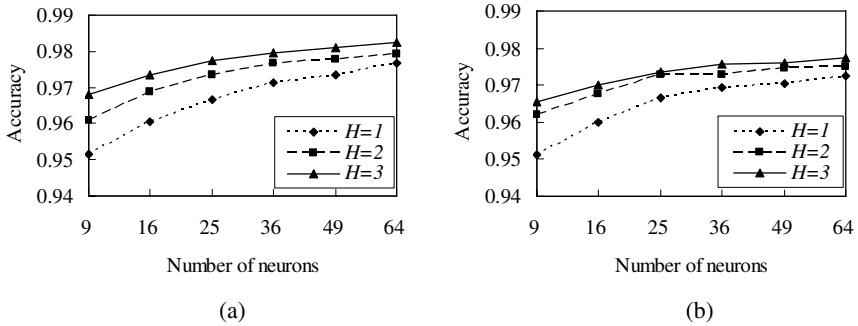


Fig. 6. Recognition accuracy (a) on the training set, and (b) on the test set of the MNIST database by using the LLOM networks

4 Conclusions

The neural model proposed in this paper is able to discover low-dimensional nonlinear manifolds embedded in data online through mixtures of local linear models. It can be applied to areas where nonlinear dimensionality reduction is desirable. The key property of this mixture model is the truly local representation of each sub-model resulted from a new distortion measure, which has largely alleviated confusion between local sub-models, as demonstrated by experiments. The objective function based on the new distortion measure does not contain local extremum, which is also desirable. Some related issues to be addressed in the future include learning temporally dependent data and automatically determining the number of sub-models.

Acknowledgements

This work was supported by the Sun Yat-sen University Science Foundation under Grant 3171910, by the National Natural Science Foundation of China (NSFC) under Grant 60473109, and by the Distinguished Young Scholars of NSFC Grant 60525111. The authors would like to thank the anonymous reviewers for their helpful comments.

References

1. Seung, H.S., Lee, D.D.: The manifold ways of perception. *Science* 290(5500), 2268–2269 (2000)
2. Hinton, G.E., Dayan, P., Revow, M.: Modeling the manifolds of images of handwritten digits. *IEEE Trans. Neural Networks* 8(1), 65–74 (1997)
3. Kambhatla, N., Leen, T.K.: Dimension reduction by local principal component analysis. *Neural Computation* 9(7), 1493–1516 (1997)
4. Teh, Y.W., Roweis, S.: Automatic Alignment of Local Representations. *Advanced in Neural Information Processing Systems* 15, 841–848 (2003)

5. Xiang, S., Nie, F., Song, Y., Zhang, C., Zhang, C.: Embedding new data points for manifold learning via coordinate propagation. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 332–343. Springer, Heidelberg (2007)
6. Kohonen, T.: The adaptive-subspace SOM (ASSOM) and its use for the implementation of invariant feature detection. In: Fogelman-Soulié, F., Galnari, P. (eds.) Proc. Int. Conf. on Artificial Neural Networks. Paris, France: EC2 et Cie, vol. 1, pp. 3–10 (1995)
7. Kohonen, T.: Self-organizing maps, 3rd edn. Springer, Berlin, Heidelberg, New York (2001)
8. Liu, Z.-Q.: Adaptive subspace self-organizing map and its application in face recognition. *International Journal of Image and Graphics* 2(4), 519–540 (2002)
9. Zheng, H., Cunningham, P., Tsymbal, A.: Learning multiple linear manifolds with self-organizing networks. *International Journal of Parallel, Emergent and Distributed Systems* 22(6), 417–426 (2007)
10. Robbins, H., Monro, S.: A stochastic approximation method. *The Annals of Mathematical Statistics* 22(3), 400–407 (1951)
11. Lecun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. *Proc. of the IEEE* 86(11), 2278–2324 (1998)

A Creditable Subspace Labeling Method Based on D-S Evidence Theory*

Yu Zong, Xian-Chao Zhang**, He Jiang, and Ming-Chu Li

School of Software, Dalian University of Technology, Dalian 116621, China
zyphoenixjp_2005@yahoo.com.cn, xczhang@dlut.edu.cn

Abstract. Due to inherent sparse, noise and nearly zero difference characteristics of high dimensional data sets, traditional clustering methods fails to detect meaningful clusters in them. Subspace clustering attempts to find the true distribution inherent to the subsets with original attributes. However, which subspace contains the true clustering result is usually uncertain. From this point of view, subspace clustering can be regarded as an uncertain discursion problem. In this paper, we firstly develop the criterion to evaluate creditable subspaces which contain the meaningful clustering results, and then propose a creditable subspace labeling method (CSL) based on D-S evidence theory. The creditable subspaces of the original data space can be found by iteratively executing the algorithm CSL. Once the creditable subspaces are got, the true clustering results can be found using a traditional clustering algorithm on each creditable subspace. Experiments show that CSL can detect the actual creditable subspace with the original attribute. In this way, a novel approach of clustering problems using traditional clustering algorithms to deal with high dimension data sets is proposed.

1 Introduction

Clustering is a powerful exploration tool capable of uncovering previously unknown patterns in data^[1]. A cluster is a collection of data objects that are similar to one another within the same cluster and are dissimilar to the objects in the other clusters. Recently, many conventional clustering algorithms, such as k-means^[2], CLARANS^[3,4], BRICH^[5], CURE^[6], DBSCAN^[7] etc, have been represented. Those conventional clustering algorithm have sufficient effect to low dimension data sets and fail to detect the meaningful result in high-dimensional space, due to the characteristic of the high dimensional data set, such as inherent sparse, noise and zero difference. Most real world data set was represented by high-dimensional vectors. So the clustering problem, which can find the natural structure of high-dimensional data set, has become the hot and difficult topic. The notion of subspace clustering and CLIQUE^[8]

* Supported by the Nation Science Foundation of China under Grand No.90412007 , the Nation Science Foundation of China No.60503003, the Science Research Project of AnHui Education office No KJ2008B133, and the important Science Research Project of AnHui Education office NO KJ2007A072.

** Corresponding author.

algorithm were both first introduced by Agrawal et al, in 1999. CLIQUE is a grid-based algorithm using an Apriori-like method to recursively navigate through the set of possible subspace in a bottom-up way. The data space is first partitioned by an axis-parallel grid into equi-sized blocks of width ξ called units. Only units whose densities exceed a threshold τ are retained. Both ξ and τ are input parameters of CLIQUE. The bottom-up approach of finding such dense units starts with 1-dimensional dense units. The recursive step from $(k-1)$ -dimensional dense units to k -dimensional dense units takes $(k-1)$ -dimensional dense units as candidates, and generates the k -dimensional units by self joining all candidates having the first $(k-2)$ -dimensional in common. Generated candidates which density doesn't exceed τ , are eliminated. For efficiency reasons, a pruning criterion called 'coverage' is introduced to eliminate dense units lying in less 'interesting' subspaces as soon as possible. For deciding whether a subspace is 'interesting' or not, the Maximum Description Length principle is used. Naturally, this pruning bears the risk of missing some small and significant results. After generating all 'interesting' dense units, clusters are found as a maximal set of connected dense units. And then give a DNF interpret. ENCLUS^[9] is a slight modification of CLIQUE, and the major difference is the criterion used for subspace selection. The criterion of ENCLUS is based on entropy computation of a discrete random variable. The entropy of any subspace S is high when the points are uniformly distributed in S whereas it is lower the more closely the points in S are packed. Subspaces with entropy below an input parameter ω are considered as good for clustering. MAFLA^[10] is another modification of CLIQUE. An adaptive grid sizes are used, which reduces computation cost and improves the clustering quality by concentrating on the portions of the data space which have more points and thus more likelihood of having clusters. All of grid-based methods have a big drawback that they heavily depend on the position of the grids, and only get the axis-parallel results. SUBCLUS^[11] is a density-based subspace clustering algorithm. The essential idea of SUBCLUS is redefined the notion of DBSCAN to adapt to subspace clustering requirement. So it has the same drawbacks of DBSCAN, that is the input parameter ϵ and m . The global parameter makes SUBCLUS can not deal with data sets, which have no uniform density subspace.

Those classical subspace clustering algorithm almost use a local search method to detect subspace and its clusters. But, which subspace contain cluster results is not ascertain, that is one don't know which attribute subset can contain interesting result. In this paper, we regard subspace clustering as the uncertain discussion problem, which was bought by 'unknown'. Based on D-S evidence theory, we firstly develop the criterion to evaluate creditable subspaces which contain the meaningful clustering results. All subsets of the original attribute space have a hypothesis: H_0 : subset of attribute contain cluster and H_1 subset of attribute not contain cluster. So a frame of discernment $\Theta = \{H_0, H_1\}$ is established and the state of it is described by probability assignment function. For finding meaningful clustering result, the probability assignment function must sufficiently capture the distribution of subspace. KNN kernel density estimation can efficiently capture the local characteristic and density distribute, so we use it as the probability assignment function of $\Theta = \{H_0, H_1\}$. and the Belief of subset S is calculated by the probability assignment function. If the Belief exceeds

the user parameter σ , then we call the subset S is a Creditable Subspace. Secondly, we also propose a Creditable Subspace Labeling (CSL) to detect those Creditable subspaces, which embedded in the original attribute space. CSL generates a candidate credible subspace set based on the belief value of each $S = \{j\}, j = 1, 2, \dots, d$. And then we use the Dempster rule to combine different candidate credible subspaces. The Belief values of new candidate credible subspace is calculated and for deciding keep or discard it. Iteratively executing those steps until finding all the credible subspaces C^s of the original space. At last, a conventional clustering algorithm which meets the users' requirement is processed on each credible subspace of C^s . Experiences on amounts simulation data sets show that CSL can find true subspace of original attribute space. This method proves a new path to deal with high dimension data set clustering problem, using conventional clustering algorithm.

The remainder of the paper is organized as follows. In Section 2, we introduce the background of D-S evidence theory. A Creditable Subspace Labeling Method based on D-S Evidence Theory and the framework of CSL are presented in Section 3. A broad experimental evaluation of CSL based on artificial data sets is presented in Section 4. Section 5 draws conclusions and points out our future work.

2 Background of D-S Evidence Theories

D-S evidence theory was presented by Dempster in 1967^[12]. Shafer, Dempster's student, has developed it which has become a whole theory of mathematic reasoning nowadays. D-S theory can be seen as a general extent of classical reasoning theory within the finite domain. The major character of D-S theory is that it supports the description of different level precision and directly introduces the description of uncertain. It supports possibility reasoning, diagnosis, risk analysis, decision-making etc, and has been applied in multi sensors network, diagnosis of hospital areas.

D-S theory is built on no-empty finite domain Ω . Ω is the frame of discernment (FOD), denotes finite system state $\{\theta_1, \theta_2, \dots, \theta_n\}$. The hypothesis of system state H_i is a subset of Ω , which is one element of the power set of $\Omega -- P(\Omega)$. The aim of D-S theory is to educe all the current system states based on some observation of the system states E_1, E_2, \dots, E_m , which couldn't uniquely confirm some system states, but is only the behave of the uncertain of system state. As the fundamental notion of D-S evidence theory, the probability function of some evidence to support a system sate is proposed first.

Definition 1: Assume function $m : P(\Omega) \rightarrow [0,1]$, and satisfy:

$$m(\emptyset) = 0 \quad \sum_{A \subseteq \Omega} m(A) = 1$$

Then, m is called as probability function of $P(\Omega)$.

Definition 2: Belief Function $Bel : P(\Omega) \rightarrow [0,1]$ is defined as:

$$Bel(A) = \sum_{B \subseteq A} m(B) \text{ for all } A \subseteq \Omega.$$

$Bel(A)$ denotes the belief of A is true.

In real application, different probability distribute function will be defined from different sources. So Dempster rule is used to combine those functions.

Definition 3: Suppose m_1 and m_2 are two probabilities distribute function which come from different sources, then the Dempster rule $m = m_1 \oplus m_2$ satisfies:

$$m(\emptyset) = 0$$

$$m(A) = K^{-1} \times \sum_{x \cap y = A} m_1(x) \times m_2(y)$$

Where: $K = 1 - \sum_{x \cap y = \emptyset} m_1(x) \times m_2(y) = \sum_{x \cap y \neq \emptyset} m_1(x) \times m_2(y)$.

Dempster rule of d different sources probability distributes is defined as:

$$m_{1\dots d}(A) = K_d^{-1} \times \sum_{x_1 \cap x_2 \cap \dots \cap x_d = A} m_1(x_1) \times m_2(x_2), \dots, m_d(x_d)$$

When $A \neq \emptyset$ $K_d = \sum_{x_1 \cap x_2 \cap \dots \cap x_d \neq \emptyset} m_1(x_1) \times m_2(x_2), \dots, m_d(x_d)$.

3 A Creditable Subspace Labeling Method Based on D-S Evidence Theory

In this section, the definitions of KNN kernel density estimate and Creditable Subspace are proposed first. And then a framework of CSL is given.

3.1 Relevant Define

Given a data set $D = \{x_1, x_2, \dots, x_n\} | x_i \in \mathfrak{R}^d$. $Attr = \{a_1, a_2, \dots, a_d\}$ is the attribute set of D , and the range of $a_i (i = 1, 2, \dots, d)$ is $[0, Range]$, $Range \in \mathfrak{R}^+$. Any subset $S \subseteq Attr$, is called a subspace. The cardinality of S is called the dimensionality of S . There are two common steps in the subspace clustering algorithm: first to confirm the subspace and then to evaluate whether it has interesting cluster or not. For a given subset $S \subseteq Attr$, there is an proposition:

- H_0 : subset S contains cluster.
- H_1 : subset S not contains cluster.

According to the definition of D-S theory, the frame of discernment Ω is denoted as $\{H_0, H_1\}$ and $H_0 \cap H_1 = \emptyset$. Simultaneously the power set of Ω is defined as $P(\Omega) = \{\emptyset, H_0, H_1, (H_0, H_1)\}$, where $m(\emptyset) = 0$. Subspace clustering has such character: in a given subspace, there are may be contain clustering result C_s , or not. Based on this character we define $m(H_0, H_1) = 0$. In this paper, the frame of discernment only contains two mutex elements. In the quotation [13], Dempster rule is proved as an NP-hard problem. But, the time complex of Dempster rule is $O(d)$, since the frame of discernment only contains mutex element^[14].

Local character and density distribution of data set can be captured by KNN kernel density estimate. So we use the KNN kernel density to denote the probability distribution of subspace S .

Definition 4: Let $D = \{x_1, x_2, \dots, x_n\} x_i \in \mathfrak{R}^d$, $K_d(x)$ be a d -dimensional probability function, $H_x = [h_1^x, h_2^x, \dots, h_d^x]$ be the d -dimensional window-width vector around v_x . Then denote:

$$\hat{f}_n(x) = \frac{1}{nV_x} \sum_{i=1}^n K_d((x-x_i)/H_x)$$

as a multi-dimensional KNN kernel density estimate, where $V_x = \prod_{i=1}^d h_i^x$.

KNN kernel density estimate method has some characters, such as smoothly estimate result, adaptive window-width etc. so it can capture the true distribution of the data set.

Note $mean_pdf(S) = \frac{\sum_{i=1}^n \hat{f}_n(x_i)}{n}$ as the mean KNN kernel density of all data points in the subset S . $Count(S) = \sum_{i=1}^n (\hat{f}_n(x_i) \geq mean_pdf(S))$. $Count(S)$ reflects the local character of data, due to it is described by k neighbors of point. Simultaneously, the kernel radius of KNN kernel density estimate is flexible, so that it can represent the situation, which the density is not uniformity. In this paper, we denote $m_s(H_0) = Count(S)$ and $m_s(H_1) = 1 - Count(S)$, to represent the belief of H_0 and unbelief of H_0 respectively. We give the definition of Belief Subspace under this denotation.

Definition 5: Given a subspace S , if $Bel(S) = \sum_{B \subseteq S} m(B) \geq \sigma$, then S is a Belief Subspace, or Creditable Subspace.

3.2 CSL Algorithm

3.2.1 The idea and Frame of CSL

Fig1 shows the basic framework of CSL. Give a data set D , nearest neighbor number k , kernel density function *kernal* and the belief parameter σ . CSL can find all Creditable Subspace and return Creditable Subspace set C^s . Firstly step of CSL is to initialize the Creditable Subspace set C^s and its candidate sets *candidate*. Secondly, for each dimension $j \in Attr$ of the data set D , denoting $S = \{j\}, j = 1, 2, \dots, d$. Generating a nearest neighbor table $T_{n \times k}$ after finding k nearest neighbor of each $x_i \in D$. The value of KNN kernel density estimation of each x_i is calculated through $T_{n \times k}$. And then $Count(S)$ is calculated as $Count(S) = \sum_{i=1}^n (\hat{f}_n(x_i) \geq mean_pdf(S))$. we obtain the probability distribution about the hypothesis of H_0 and H_1 within each dimension, by set $m_s(H_0) = Count(S)$ and $m_s(H_1) = 1 - Count(S)$ respectively. After calculating the Belief value of each $S = \{j\}, j = 1, 2, \dots, d$, one will be hold which $Bel(S) \geq \sigma$ as candidate Creditable subspace, $candidate = candidate \cup S$. As shown in the for loop of the CSL. The last step of CSL iteratively selects two different candidates Creditable Subspace to combine using Demspter rule and generate new subspace. The Belief value of new subspace is calculated and adds to candidate Creditable Subspace set. The algorithm must span all the possible subsets to obtain the best combination, but this step needs exhaust compute time. Due to it, we give a greedy search to find the local solution. Step 3 in fig1 is a while loop to accomplish this search method. If $candidate \neq \emptyset$, then select a subset s , and make a Demspter rule with $S' \in candidate - S$. The max m is

selected as $m = \max(m_s \oplus m_{s'})$, and then recalculates the belief of new subspace $Bel(S \cup S') = \sum_{B \subseteq S \cup S'} m(B)$. If $Bel(S \cup S') \geq \sigma$, then $candidate = candidate - \{S\} - \{S'\}$, $S \leftarrow \{S\} \cup \{S'\}$, $candidate = candidate \cup \{S\}$. When there no new subset appends to S , let $C^s = C^s \cup S$ and $candidate = candidate - \{S\}$. Iteratively operate this process until $candidate = \emptyset$.

Algorithm 1: CSL

Input: data set D , nearest neighbor number k , kernel function $kernal$ and belief parameter σ .

Output: Belief Subspace set C^s

```

1.  $C^s = \emptyset$ ;  $candidate = \emptyset$ 
2. for  $j = 1:d$ 
    2.1.  $T = \emptyset$ ;  $dft = \emptyset$ ;
    2.2.  $T = find\_knn\_neighbor(D, k)$ ;
    2.3.  $dft = caculate\_kernal(D, T, kernal)$ ;
    2.4.  $m(j) = gernerate\_paf(dft)$ ;
    2.5.  $Bel(j) = cacluate\_bel(m)$ ;
    2.6. if  $Bel(j) \geq \sigma$   $candidate = candidate \cup \{j\}$ 
        end
3. while( $candidate \neq \emptyset$ )
    3.1.  $S = get\_first(candidate)$ ;
    3.2.  $C = S$ ;
    3.3. while( $candidate - S \neq \emptyset$ )
        3.3.1.  $S' \in candidate - C$ ;
        3.3.2.  $m(C \cup S') = orthogonal\_operator(C, S')$ ;
        3.3.3.  $Bel(C \cup S') = cacluate\_bel(m)$ ;
        3.3.4. if  $Bel(C \cup S') \geq \sigma$ 
             $S = C \cup S'$ ;
             $candidate = cadidate - S$ ;
        end
    end
    3.3.5.  $C^s = C^s \cup S$ ;
    3.3.6.  $candidate = candidate - S$ ;
end

```

Fig. 1. CSL Algorithm

3.2.2 Algorithm Complexity

In the for loop, the primary function of $find_knn_neighbor(D, k)$ is to find k nearest neighbor of each $x_i \in D$ on each dimension. The time complexity of this function

is $O(n^2)$. But if a $k-d$ tree structure is used, the time complexity will be reduced to $O(n \log n)$. Thus, the time complexity of $find_knn_neighbor(D,k)$ is less than or equal to $O(n^2)$. Analogously the time cost of $calculate_kernel(D,T,kernal)$, $generate_paf(dft)$ and $calculate_bel(m)$ is $O(n)$ respectively. So the whole time complexity of for loop is $O(d(n^2 + 3n))$ at most. In while loop, the Dempster rule is used to combine probability distribution which comes from different sources, and calculates the belief of new subspace. As fore discussion, the cost of calculate d different probability distribution is $O(d^2)$. Thus, the whole time complexity of CSL is $O(d(n^2 + 3n)) + O(d^2) = O(d(n^2 + 3n) + d^2)$.

4 Results and Analysis

We evaluated precision and efficiency of CSL using several synthetic data sets. All experiments were run on a workstation with a 1.7GHz processor and 2G RAM. The proposed CSL algorithm is implemented in matlab 6.5.

The synthetic data sets were generated by a self-implemented data generator. It permits to control the size and structure of the generated data sets through parameters, such as the size and dimensionality of data set, the dimensions of the subspace, the number of clusters and the range ($[0, Range], Range \in \mathfrak{R}^+$) of each dimension. For the simple, we use the same label method, which was represented in [14], to express data set. 'B', 'C', 'D', 'S' denote the records of data set, the number of clusters, the dimensions of data set and subspace respectively. For example, B10000C10D50S15 expresses a data set that contains 10000 records, each record has 50 dimensions, and there are 10 clusters through 15 dimensions. In this paper, we evaluate LCS from the precision and efficiency aspects.

(1) Precision of CSL

We denote D^s contain the true subspace of D , and then define $precision = |C^s|/|D^s|$ as the precision of the CSL. We perform CSL, CLIQUE ENCLUS and SUBCLUS on B30000C10D50S5, B30000C10D50S10, B30000C10D50S15, B30000C10D50 S20, B30000C10D50S25, B30000C10D50S30, B30000C10D50S35, and compare their precision. Experiment results as shown in fig2. The parameter of CLIQUE ENCLUS and SUBCLUS is the same as given in the quotation. Otherwise, the parameter of CSL is set as $k = \lceil \sqrt{n} \rceil$, $\sigma = m(S) * 0.80$. From fig2, we observe that the precision of CSL is higher than others, because CSL is based on KNN kernel density estimate. This character makes CSL can capture the local distribute of data set, and easy to find the density units location. The precision of SUBCLUS is the lowest, because its idea is the concept of DBSCAN, and can not do well without the parameter's influence.

(2) Efficiency of CSL.

We evaluate the Efficiency of CSL from three aspects: the size and the dimensions of data set, the dimensions of subspace. Experience of evaluating the scalability of CSL against the size of dataset was performed on D5000C6D20S5, D10000C6D20S5, D15000C6D20S5, D20000C6D20S5, 15000C6D20S5, D30000C6D 20S5,

D35000C6D20S5, D40000C6D20S5, D45000C6D20S5. On data sets: D20000C6 D10S5, D20000 C6D15S5, D20000C6D20S5, D20000C6D25S5, D20000C6 D30S5, we evaluate the scalability of CSL against the dimensions of data set. Analogously, we also evaluate the scalability of LCS against the dimensions of subspace on D30000C6D50S2, D30000C6D50S4, D30000C6D50S6, D30000C6D50S8, and D30000C6D50S10.

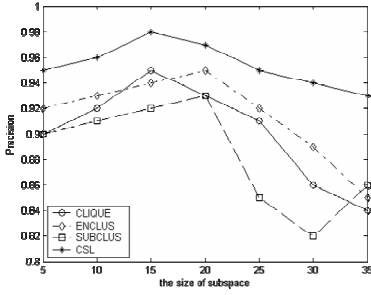


Fig. 2. Precision comparison on synthetic dataset

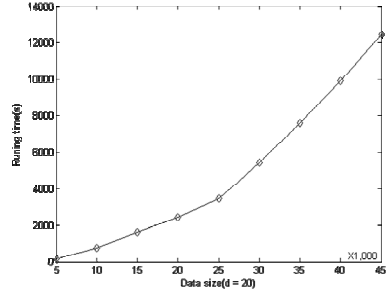


Fig. 3. Scalability of CSL against the size of dataset

Fig 3, 4, 5 show the experiment of three aspects respectively. From these figure, we observe that CS L has better scalability of the size and dimensions of data set, the dimensions of subspace.

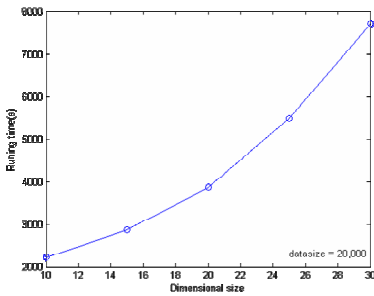


Fig. 4. Scalability of CSL against the dimensions of dataset

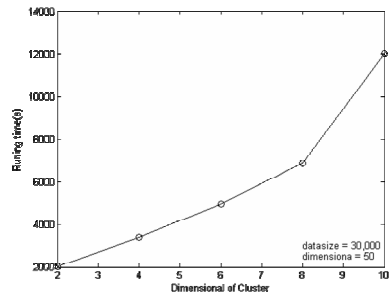


Fig. 5. Scalability of CSL against the dimensionality of the subspace

5 Conclusion

Subspace clustering is an important method to deal with high-dimensional clustering problem, because conventional clustering algorithm (algorithms) can not detect the

true data distribution in high-dimensional space. Subspace clustering tries to find the true clusters embedded in subset of original attribute. But, which subset of attribute contains interest cluster is uncertain. So we regard subspace clustering as an uncertain reasoning problem, which arisen by 'unknown'. In this paper, a Label Creditable Subspace method based on D-S evidence theory is proposed. Simultaneously, we also give the definition of Belief Subspace and its evaluation criterion. LCS iterative finds all the belief subspace of original space. According to the application domain, use a conventional clustering algorithm on the subspace set to get the results. Experiments show that LCS can find the true subspace of data set and has better scalability of the size and dimensions of data set, the dimensions of subspace. The method of this paper shows a new way to use conventional clustering algorithm to deal with high-dimensional data set.

References

- [1] Berkhin, P.: Survey of clustering data mining techniques. Technical report, Accrue Software, San Jose, California (2002)
- [2] Hartigan, J.A., Wong, M.A.: A k-means clustering algorithm. *Applied Statistics* (28), 100–108 (1979)
- [3] Ng, R.T., Han, J.: Efficient and Effective Clustering Method for Spatial Data Mining. In: *Proceeding of the 20th VLDB Conference*, pp. 144–155 (1994)
- [4] Ng, R., Han, J.: CLARANS: A method for clustering objects for spatial data mining. *IEEE Trans. on Knowl., Data Eng.* 14(5), 1003–1016 (2002)
- [5] Zhang, T., Ramakrishna, R., Livny, M.: BIRCH: A New Data Clustering Algorithm and its Applications. *Journal of Data Mining and Knowledge Discovery*, 141–182 (1997)
- [6] Guha, S., Rastogi, R., Shim, K.: CURE: An efficient clustering algorithm for large database. In: *Proceedings of the 1998 ACM SIGMOD international conference on Management of data*, pp. 73–84 (1998)
- [7] Ester, M., Kriegel, H.P., Sander, J., et al.: A density-based algorithm for discovering clusters in large spatial database. In: *Proc.1996 Int. Conf.Knowledge Discovery and Data Mining (KDD 1996)*, Portland, OR, August 1996, pp. 226–231 (1996)
- [8] Agrawal, R., Gehrke, J., Gunopulos, D., et al.: Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. In: *Proc.1998 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD 1998)*, Seattle, WA, June 1998, pp. 94–105 (1998)
- [9] Cheng, C.-H., Fu, A.W., Zhang, Y.: Entropy-based subspace clustering for mining numerical data. In: *Proceedings of the 5th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 84–93. ACM press, New York (1999)
- [10] Goil, S., Nagesh, H., Choudhary, A.: MAFIA: Efficient and scalable subspace clustering for very large data sets. Technical Report CPDC-TR-9906-010, Northwestern University, 2145 Sheridan Road, Evanston IL 60208 (June 1999)
- [11] Kailing, K., Kriegel, H., Kroger, P.: Density-connected Subspace Clustering for High-dimensional Data. In: *Proc. 4th SIAM Int. Conf. on Data Mining*, Lake Buena Vista, FL, pp. 246–257 (2004)
- [12] Dempster, A.: Upper and Lower Probabilities induced by multivalued mapping. *Annals of Mathematical Statistics* 38(2), 325–339 (1967)

- [13] Orponen, P.: Dempsster's rule of combination is #P- complete. *Artificial Intelligence* 44(1-2), 245–253 (1990)
- [14] Jian-Wei, Z., Da-Wei, W., Yu, C., et al.: A Network Anomaly Detector Based on the D-S Evidence Theory. *Journal of Software* 17(3), 463–471 (2006)
- [15] Xiaoyun, Z., Zhihui, S., Baili, Z., et al.: An Efficient Discovering and Maintenance Algorithm of Subspace Clustering over High Dimensional Data Streams. *Journal of Computer Research and Development* 43(5), 834–840 (2006)

Discovering New Orders of the Chemical Elements through Genetic Algorithms

Alexandre Blansch ¹ and Shuichi Iwata²

¹ Laboratoire des Sciences de l'Image, de l'Informatique et de la T l d t ction
Strasbourg, France

blansche@lsiit.u-strasbg.fr

² The University of Tokyo, Graduate School of Frontier Sciences
Tokyo, Japan

iwata@k.u-tokyo.ac.jp

Abstract. The design of new materials is a major issue in many domains (electronics, environment and so on). A large number of databases have been developed in order to help scientists to design new materials. Databases of experimental results can be used to learn prediction models of each property. Data mining methods, can be applied on such databases to discover empirical rules and predict properties.

In this paper we propose a method for discovering new orders of the chemical elements. This reorganization of the chemical elements can be used to improved prediction accuracy of classification methods and to enhance similarities between elements. A genetic algorithm is used to find a satisfying solution according to several evaluation criteria through a Pareto-based multi-objective approach.

We carried out several experiments of prediction of compound formation (ternary chalcopyrite compounds ABX_2 , where X is either S, Se or Te). The first results showed that distance-based evaluation seems promising, as it has been possible to discover groups of similar elements regarding the task.

Keywords: Supervised classification, Order discovery, Genetic algorithm, Material design.

1 Introduction

The design of new materials is a major issue in many domains (electronics, environment and so on). However, material design is a difficult task. Therefore, a large number of databases have been developed in order to help scientists to design new materials. Databases of experimental results can be used to learn prediction models of each property. Data mining methods can be applied on such databases to discover empirical rules and predict properties in order to restrain the research of new material to the more promising cases.

We are interested to know if accurate predictions can be realized using only basic information of the periodic table. In this paper, we propose a method for discovering new orders of the chemical elements. This reorganization of the chemical elements can be used to improved prediction accuracy of classification method and to understand the

physics beyond the predictions. The proposed method uses two types of information: labeled instances from the dataset and the position of the elements in the periodic table (atomic number, chemical group, etc.). A genetic algorithm is used to explore the possible orders of elements.

In Sec. 2 we will present the new order discovering method and results are shown in Sec. 3.

2 Method

In this section, we will present the genetic algorithm used for order discovery. We will first present the data representation (Sec. 2.1) and the goals of chemical elements ordering (Sec. 2.2). Then, we will detail the method (Sec. 2.3) and the fitness functions that can be used (Sec. 2.4).

2.1 Data Representation

In this paper, we will focus on two-classes problems. The datasets concern the formation of specific compound types according to a set of elements. A possible problem can be described as follow: considering the set of elements $E = \{A, B, C\}$ is it possible to create a compound XYZ_2 , where $X \in E$, $Y \in E$, $Z \in E$, $X \neq Y$, $X \neq Z$ and $Y \neq Z$.

We consider only information based on the position of the elements in the periodic table and the labeled instances of the dataset. Datasets are composed of a list of instances, described by the atomic number of each element of the potential compound and the class label (formation or non formation).

Definition 1. A distance measure can be defined between two instances, using the minimum Euclidean distance among all permutations of elements of the instances. For instance, considering two instances $a = (a_1, \dots, a_n)$ and $b = (b_1, \dots, b_n)$ the distance can be defined by:

$$d(a, b) = \min_{\sigma \in S} \left(\sqrt{\sum_{i=1}^n (d(a_i, b_{\sigma(i)}))^2} \right),$$

where n is the number of elements, a_i (resp. b_i) is the order rank (e.g. atomic number) of the i -th element of the instance a (resp. b), $d(a_i, b_j) = |a_i - b_j|$ and S is the set of all permutation of numbers from 1 to n .

2.2 Ordering of Elements

The most common order of the chemical elements is the atomic number. However, this order does not give enough information about the properties of the elements. Elements with close atomic numbers may not have close properties (e.g. H and He are elements 1 and 2, but have very different chemical properties), whereas some elements with very different atomic numbers (e.g. elements of a same group) have similar properties.

In [1], a new order of the elements has been defined to predict the formation of one type of crystal structure of binary compounds. In this order, elements with close ranking have close properties regarding the crystal structure. Moreover, elements of a same group are close to each other. This order of elements inspired other orders (Periodic Numbers, also called Mendeleev Numbers), defined by running through the periodic table group by group. These orders have been used in [2].

Periodic Numbers are more consistent with the chemical properties of the elements. However, it could be interesting to define orders adapted to specific properties, as was the order proposed in [1]. In this paper, we propose an algorithm able to discover new orders of the elements. If we consider the chemical elements up to the actinide series (i.e. the 103 first elements), there is $\frac{103!}{2}$ different possible orders. Thus, the algorithm must be able to discover satisfying solution among the whole set of orders. This is a difficult task as there are linear and non linear relation between the element and the properties of the compounds.

2.3 Genetic Algorithms for Order Discovery

The Traveling Salesman Problem (TSP) is a well-known combinatorial optimization problem. A traveling salesman must find the shortest route to visit several towns. This problem is known to be *NP*-complete and has been widely studied. The problem of ordering the chemical elements is similar to the TSP.

Genetic algorithms have been proposed to find good enough solution [3]. Each individual is an order that will be evaluated according to the function to optimize. However, it is not trivial to define a representation of the solutions and crossover and mutation operators for orders. Several representations and operators have been defined in literature. The most common representation for orders is an ordered list of the items. However in the case of order of chemical elements, we uses the dual representation: the genotype is an array where the first element of the array is the rank of the first chemical element (H) in the new order, the second element of the array is the rank of the second chemical element (He), and so on. This dual representation does not change the crossover and mutation operators, but facilitate the implementation of the data transformation. We tested several crossover and mutation operators. First experiments seemed to show that the crossover operator POS and the mutation operator EM and SIM provide the best results.

2.4 Evaluation of Orders

As other data transformation methods (attribute selection, attribute weighing, attribute construction), the evaluation of the quality of an order can be performed through a filter approach or through a wrapper approach [4].

In the filter approach, the order is evaluated according to some statistics based on the properties of the data. In this paper, we used distance based measures: intraclass distance and interclass distance. These two evaluation criteria can be used with the distance measure defined in Def. 1. In this definition, the distance between two elements is the difference between their ranks in the order. However, as there are no available data for some elements, we defined another measure where elements with no data will not affect the distance measure (Def. 2). In a wrapper approach the quality of an order can be evaluated by the accuracy of a classification algorithm after the data transformation.

Definition 2. The modified distance between two elements of ranks a_i and b_j (with $a_i < b_j$) in an order ord is defined by:

$$d_m(a_i, b_j) = 1 + \sum_{a_i \leq c \leq b_j} Nb(c),$$

where $Nb(c) = \text{card}(o \in D \mid o = \{o_1, o_2, \dots, o_n\} \text{ and } \exists i, o_i = c)$, i.e. $Nb(c)$ is the number of instances containing the element of rank c in the order ord .

These criteria can be used to evaluate the quality of an order based on the dataset. To order correctly elements with no information, we should also take into account some knowledge about the elements. We know that elements of a same group (same column of the periodic table) and elements of a same series (transition metals, metalloids, etc.) have similar properties. According to this knowledge, elements of the same group or series must be close so each other in the order. Thus, we can define a scattering cost of an order according to the periodic table.

Definition 3. The scattering cost $scat$ can be defined by:

$$scat(ord) = \sum_{A, B \in E} \max(G(A, B), S(A, B)) \times \frac{(ord(A) - ord(B))^2}{d_M(A, B)},$$

where $G(A, B) = 1$ iff A and B are in a same group, $S(A, B) = 1$ iff A and B are in a same series and d_M is the distance between two elements in the Mendeleev table.

The algorithm will use a Pareto-based approach for multi-objective optimization [56]. Two criteria will be used at the same time: one data criterion (distance based or accuracy based) and the scattering cost.

3 Experiments

We carried out a series of tests in order to evaluate the feasibility of chemical element ordering. In Sec. 3.1 we will present the datasets used in our experiments. In Sec. 3.2 we show some experimental results.

We used a population of 100 individuals and run the algorithms for 2500 generations. We used an elitist strategy (all individuals on the Pareto front are conserved without any modification). The remaining individuals are created by crossover followed by a mutation (70%), mutation of one individual (25%) and new individuals (5%). Tournament selection is used to select parents. On the Pareto front, a priority is accorded to the data-based criterion. The algorithm has been designed in Java language using the GEAL library (<http://dpt-info.u-strasbg.fr/~blanche/en/geal.html>).

3.1 Data

In this paper, we will focus on the prediction of formation of ternary chalcopyrite compounds ABX_2 , where X is either S, Se or Te. ABX_2 compounds are promising for electronic applications because of semiconducting and nonlinear optical properties

[7][8]. This dataset was extracted from the databases on inorganic substance properties of the A.A. Baikov Institute of Metallurgy and Materials Science [9].

As element X can take only a few number of values, the complete dataset has been divided into three partial datasets, corresponding to the different prediction tasks ABS_2 , $ABSe_2$ and $ABTe_2$. In the three partial datasets, there are only two variable elements, so the new order result can easily be plotted on a grid for visual interpretation. Because of space restrictions, results on the $ABSe_2$ dataset are not shown here. There are 240 positive instances (compound formation) and 49 negative instances (compound non-formation) in the ABS_2 dataset, and 109 positive instances and 85 negative instances in the $ABTe_2$ dataset.

3.2 Results

For each datasets, we tested four versions of our algorithm, using each time two evaluation criteria: intraclass distance and scattering cost, interclass distance (standard or modified) and scattering cost, or accuracy (using the Nearest Neighbor algorithm) and scattering cost. Only representative results are shown in this paper.

The datasets can be plotted on symmetric two-dimensions grids. Each row and each column corresponds to one element. The order of rows and columns is defined by the order of the elements (from left to right, from top to bottom). Each square of the grid is associated to a potential compounds ABX_2 colored in dark grey in case of non formation of compounds, in medium gray in case of formation, and in light grey when no data are available.

On Fig. 1 and 2 are shown the projections of the ABX_2 datasets using the atomic number. One can see that there is no clear separation between formation and non formation of compounds.

On Fig. 3 and 4 are shown the results obtained using the intraclass distance and the scattering cost as evaluation criteria. On the projection of the ABS_2 dataset, most of the formation instances are grouped together into two (symmetric) clusters. The rest of the instances (formation and non formation) are distributed in the middle of the grid. On the projections of the $ABTe_2$ dataset, all instances are grouped together into a compact cluster. Some parts of this cluster are mainly composed of formation instances,

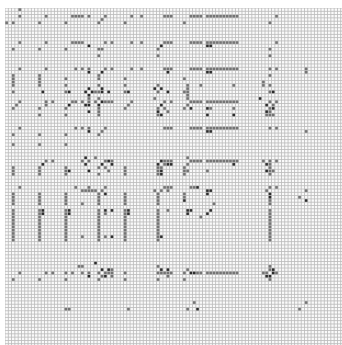


Fig. 1. Projection of ABS_2 (atomic number)

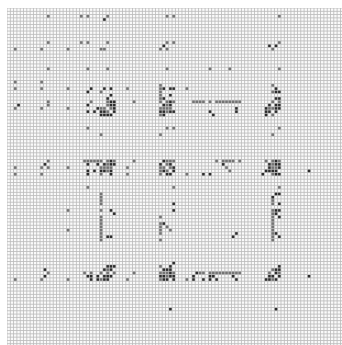


Fig. 2. Projection of $ABTe_2$ (atomic number)

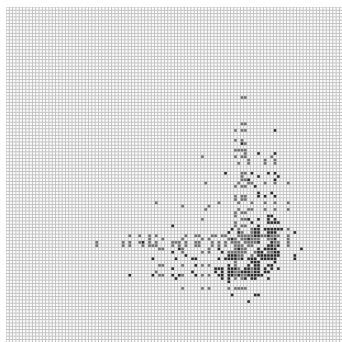
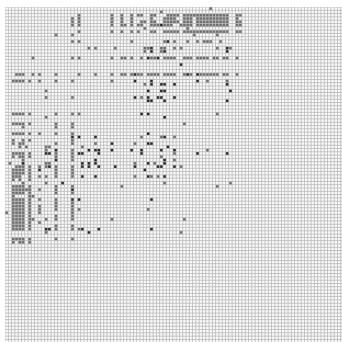


Fig. 3. Projection of ABS_2 (intra-class distance) **Fig. 4.** Projection of $ABTe_2$ (intra-class distance)

others of non formation instances, but it is difficult to see any clear separation between formation and non formation.

On Fig. 5 and 6 are shown the results obtained using the interclass distance and the scattering cost. It is easier to see a separation between formation and non formation. Two types of elements appears: low rank elements (L) and high rank elements (H). On the ABS_2 dataset, L elements seem to be able to form LBS_2 (or ALS_2) compounds with almost any other element, whereas formation (or non formation) is difficult to predict for $HH'S_2$ compounds. The $ABTe_2$ dataset can be divided into four clusters. $LL'Te_2$ are non formation instances, $HH'Te_2$ are formation instances. It is difficult to predict the formation of $HLTe_2$ and $LHTe_2$ compounds. One should note that most of the data appears on the borders of the grid.

On Fig. 7 and 8 is shown the result obtained using the accuracy (obtained by the Nearest Neighbor algorithm) and the scattering cost as evaluation criteria. Formation instances are grouped together into several line-shaped clusters. Non formation instances

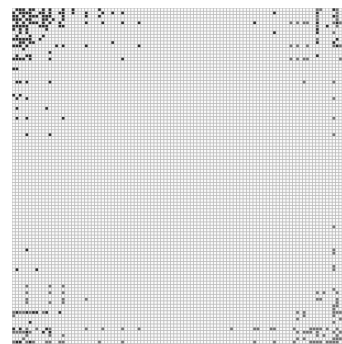
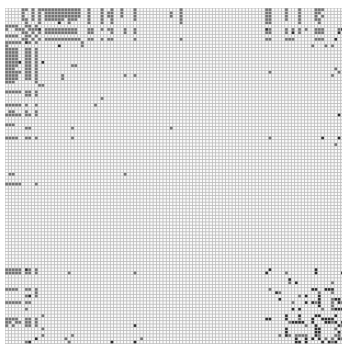


Fig. 5. Projection of ABS_2 (interclass distance) **Fig. 6.** Projection of $ABTe_2$ (interclass distance)

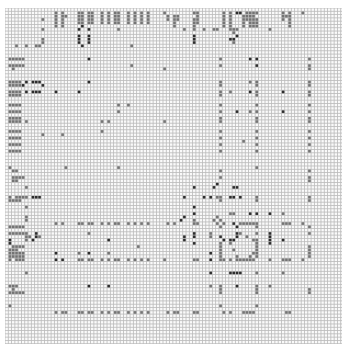


Fig. 7. Projection of ABS_2 (accuracy)

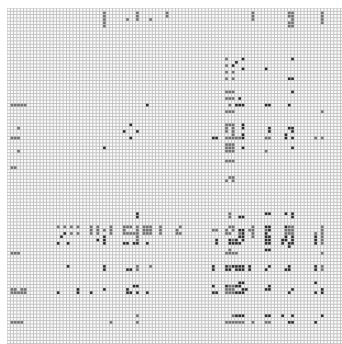


Fig. 8. Projection of $ABTe_2$ (accuracy)

are distributed among the grid, forming mostly very small clusters of less than six instances. This structure is not very helpful to understand the data.

On Fig. 9 and 10 are shown the results obtained using the interclass distance with the modified distance and scattering cost. These results are similar to the results obtained using the standard interclass distance, but the obtained clusters are not too compact, the data are more distributed all over the grid.

These results seems to show that it is possible to perform an automatic reordering of the chemical elements. Evaluation measure based on interclass distance seems more promising to reach interesting results. Using this evaluation measure, groups of chemical elements with similar properties have been detected on the three datasets. However, it seems difficult to order correctly elements with few data available.

While using a standard distance, the separation between formation and non formation is clear but instances are forming compact clusters without incorporating similar elements not present in the dataset. When the modified distance is used, it seems possible to order every elements more correctly.

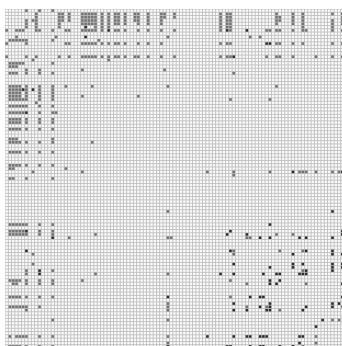


Fig. 9. Projection of ABS_2 (interclass dist., modified)

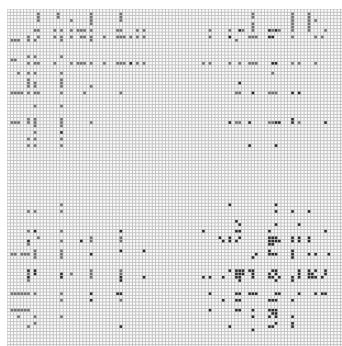


Fig. 10. Projection of $ABTe_2$ (interclass dist., modified)

Even on the best results, some inconsistencies still appear. For instance, on Fig. 10 in the upper-left part, one can see a non formation instance in the middle of a cluster of formation instances. This instance corresponds to the non formation of CeTlTe_2 . The Tl element appears in a large number of instance, but there are only two instances containing Ce. It would have been more logical to give a high rank to the element Ce for this dataset. This show that it will probably be necessary to refine the result after the end of the genetic algorithm. A local optimization method should be developed in order to improve the results.

4 Conclusion

In this paper, we presented a method for discovering new orders of elements through a genetic approach. Two criteria, one data-based and one knowledge-based, are used through a Pareto-based multi-objective approach. New orders of the elements can be used to predict the formation of compounds more efficiently. Moreover, new orders can help to enhance similarities between elements. The first results presented in this paper showed that accuracy-based evaluation does not provides interesting results, at least for the Nearest-Neighbor classification method. Distance-based approach seems more promising, as it has been possible to discover groups of similar elements elements regarding the formation of a specific compound type.

However, there are still improvement to do as elements with few information seem hard to order correctly. Other evaluation function or “intelligent” operators (such as order perturbation inside a group or a series of elements, or at the group or the series level) should be developed in the future. As a further work, we should also apply the proposed method on other datasets. We applied our method only on binary problems (formation or non formation), but it could be interesting to test it on multi-classes problem like the discrimination between several crystal structures. Moreover, we only applied our method on two elements problems and should now be tested with datasets of compounds of three or more elements. On more complex datasets, cluster analysis would be useful to discover groups of similar elements.

Acknowledgments

We should like to thank Dr. N.N. Kiselyova from the the A.A. Baikov Institute of Metallurgy and Materials Science for her assistance in the preparation of training set.

References

1. Pettifor, D.: A chemical scale for crystal-structure maps. *Solid State Communications* 51(1), 31–34 (1984)
2. Villars, P., Cenzual, K., Daams, J., Chen, Y., Iwata, S.: Binary, ternary and quaternary compound former/nonformer prediction via mendeleev number. *Journal of Alloys and Compounds* 367, 167–175 (2004)
3. Larrañaga, P., Kuijpers, C., Murga, R., Inza, I., Dizdarevic, S.: Genetic algorithms for the travelling salesman problem: A review of representations and operators. *Artificial Intelligence Review* 13, 129–170 (1999)

4. Motoda, H., Liu, H.: Feature selection, extraction and construction. In: *The Handbook of Data Mining*, pp. 409–423. Lawrence Erlbaum Associates, Inc. Publishers, Mahwah (2003)
5. Fonseca, C., Fleming, P.: An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation* 3(1), 1–16 (1995)
6. Tan, K., Khor, E., Lee, T.: *Multiobjective evolutionary algorithms and applications*. Springer, Heidelberg (2005)
7. Bai, L., Lin, Z., Wang, Z., Chen, C., Lee, M.H.: Mechanism of linear and nonlinear optical effect of chalcopyrite AgGaX_2 ($X=\text{S}$, Se , and Te) crystals. *Journal of Chemical Physics* 120(18), 8772–8778 (2004)
8. Tanaka, K., Uchiki, H.: Optical second-harmonic generation from CuGaS_2 (112) bulk single crystals. *Optics Communications* 193, 313–317 (2001)
9. Kiselyova, N.N.: *Computer Design of Inorganic Compounds, Application of Databases and Artificial Intelligence*. Nauka, Moscow (2005)

What Is Frequent in a Single Graph?

Björn Bringmann and Siegfried Nijssen

K.U. Leuven, Celestijnenlaan 200 A, B-3001 Leuven, Belgium
{Bjoern.Bringmann,Siegfried.Nijssen}@cs.kuleuven.be

Abstract. The standard, *transactional* setting of pattern mining assumes that data is subdivided in transactions; the aim is to find patterns that can be mapped onto at least a minimum number of transactions. However, this setting can be hard to apply when the aim is to find graph patterns in databases consisting of large graphs. For instance, the web, or any social network, is a single large graph that one may not wish to split into small parts. The focus in network analysis is on finding structural regularities or anomalies in one network, rather than finding structural regularities common to a *set* of them. This requires us to revise the definition of key concepts in pattern mining, such as support, in the *single-graph* setting. Our contribution is a support measure that we prove to be computationally less expensive and often closer to intuition than other measures proposed. Further we prove several properties between these measures and experimentally validate the efficiency of our measure.

Keywords: graph mining, pattern mining, network analysis.

1 Introduction

The traditional *transactional* setting of pattern mining is popular for many types of data with well-known examples being basket analysis [1], or molecular fragment mining [10]. As stated in the abstract, the single-graph setting introduces problems that do not appear in the transactional setting; the most prominent one being the definition of the support of a pattern. Naïve definitions of support have the problem that they are not anti-monotonic; thus they cannot be used effectively in pattern mining, as anti-monotonicity is required to prune the search space. To address this problem, Kuramochi and Karypis [7] as well as Fiedler and Borgelt [5] studied anti-monotonic support measures based on computing maximum independent sets (MIS) in overlap graphs.

Anti-monotonicity is however not the only requirement for efficient frequent pattern mining. It is also important that the frequency measure can be evaluated efficiently. We argue that the computation of overlap-based support measures is not feasible in many graph databases, and that more scalable support measures are needed to enable the use of frequent graph mining algorithms on network data. We propose a new support measure, and provide practical and theoretical evidence that this measure is more scalable, more general and more widely applicable than the support measures mentioned earlier. We show how this measure, and the overlap-graph based measures, relate to each other, thus providing deeper understanding in support measures for graph mining.

2 The Support of a Pattern

A labeled graph $g = (\mathbb{V}_g, \mathbb{E}_g, \lambda_g)$ consists of a set of nodes \mathbb{V}_g , a set of edges $\mathbb{E}_g \subseteq \mathbb{V}_g \times \mathbb{V}_g$ and a labeling function $\lambda_g : \mathbb{V}_g \cup \mathbb{E}_g \rightarrow \Sigma$ that maps each element of the graph to an element of the alphabet Σ . Let G_Σ be the set of all graphs over the alphabet Σ . We define support as a function $\sigma : G_\Sigma \times G_\Sigma \rightarrow \mathbb{N}$.

As stated earlier, minimum support needs to be anti-monotonic to allow efficient search. This means that for all graphs g, p and p' , where p is a subgraph of p' , it must hold that $\sigma(p, g) \geq \sigma(p', g)$. Anti-monotonicity is quite easily upheld in the transactional setting, but is more tricky for the single-graph setting. The cause of this problem is that it is not clear what exactly should be counted.

Occurrence of a Pattern. Given a pattern $p = (\mathbb{V}_p, \mathbb{E}_p, \lambda_p)$ and a data graph $g = (\mathbb{V}_g, \mathbb{E}_g, \lambda_g)$, an occurrence is a function $\varphi : \mathbb{V}_p \rightarrow \mathbb{V}_g$ mapping the nodes of p to the nodes in g such that (I) $\forall v \in \mathbb{V}_p \Rightarrow \lambda_p(v) = \lambda_g(\varphi(v))$ and (II) $\forall (u, v) \in \mathbb{E}_p \Rightarrow (\varphi(u), \varphi(v)) \in \mathbb{E}_g$. The image of a set of nodes in an occurrence is denoted $\varphi(\mathbb{V}_p) = \{\varphi(v) | v \in \mathbb{V}_p\}$; similarly, we define the image of a set of edges.

The problem of the support measure on a single graph is explained in Figure 1. p_1 has one occurrence in g , and p_2 is a specialization of p_1 . What is the support of p_2 in g ? In the transactional setting every instance with at least one occurrence counts. This is undesirable in the single-graph setting, since every graph would have a support of either zero or one. A naïve measure that assigns a support of 2 in our example, would not be anti-monotonic.

Single Graph Support Measures. For the support measure introduced in [7], all possible occurrences φ_i of a pattern p in the graph g are calculated. An overlap-graph is constructed where each occurrence φ_i corresponds to a node and there is an edge (φ_j, φ_k) iff $\varphi_j(\mathbb{E}_p) \cap \varphi_k(\mathbb{E}_p) \neq \emptyset$ (i.e.: φ_j and φ_k share an edge). The support for the pattern p is defined as the size of the maximum independent set (MIS) of the overlap-graph. For example, in Figure 2 there would be four occurrences of the pattern p in the graph g . Even though [7] defined overlap in terms of edges, the concept can also be applied to vertices. For this case, we formalize the following binary relationship:

Definition 1. simple overlap φ and φ' are simple overlaps of p iff $\varphi(\mathbb{V}_p) \cap \varphi'(\mathbb{V}_p) \neq \emptyset$

We denote the support measure based on simple overlaps as σ_\bullet . It can be shown that this support measure is anti-monotonic. However, solving a MIS problem is NP-complete.

A refinement of the simple overlap based support measure was introduced in [5] and named harmful overlap. We will denote this by σ_\circ . The basic idea of this measure is that some of the simple overlaps can be disregarded without harming the anti-monotonicity of the support measure. As before, an overlap graph is constructed and the support is defined as the size of the MIS. Different is the definition of overlap:

Definition 2. harmful overlap φ and φ' are harmful overlaps of p iff $\exists v \in \mathbb{V}_p : \varphi(v), \varphi'(v) \in \varphi(\mathbb{V}_p) \cap \varphi'(\mathbb{V}_p)$

Note that both σ_\bullet and σ_\circ are based on shared nodes here. However, σ_\circ measures can be used either based on σ_\bullet or on σ_\wedge .

Both measures rely on computing an overlap graph, and subsequently solving a MVC problem. We propose a measure of support, which avoids potentially expensive MVC computations¹. It is based on the number of unique nodes in the graph $g = (\mathbb{V}_g, \mathbb{E}_g)$ to which a node of the pattern $p = (\mathbb{V}_p, \mathbb{E}_p)$ is mapped.

Definition 3.

$$\sigma_\wedge(p, g) = \min_{v \in \mathbb{V}_p} |\{\varphi_i(v) : \varphi_i \text{ an occurrence of } p \text{ in } g\}|$$

By taking the node in p which is mapped to the least number of unique nodes in g , we can ensure the anti-monotonicity of σ_\wedge . From our definition of support, we can deduce several computational benefits: **(i)** instead of $O(n^2)$ potential overlaps, where n is the possibly exponential number of occurrences, we only need to maintain a set of data vertices for every node in the pattern, which can be done in $O(n)$; **(ii)** we do not need to solve an NP complete MVC problem; **(iii)** it is not necessary to compute all occurrences: it is sufficient to determine for every pair of $v \in \mathbb{V}_p$ and $v' \in \mathbb{V}_g$ if there is an occurrence in which $\varphi(v) = v'$. The computational burden can be reduced further by taking into account the automorphisms of the pattern graph.

Relationships and Dependencies.

All measures introduced are based on the occurrence of patterns, but they can give different results on the same data. An example for how the three measures work and that they give different results can be found in Figure 2.

Nevertheless, several relationships between these measures hold. We can show that our measure σ_\wedge is an upper bound for the harmful overlap measure σ_\circ , which is in turn an upper bound for the simple overlap based measure σ_\bullet .

Theorem 1. $\sigma_\wedge \geq \sigma_\circ \quad \forall p \in \mathcal{P} : \sigma_\wedge(p, \mathcal{T}) \geq \sigma_\circ(p, \mathcal{T})$

Let $v^* = \arg \min_{v \in \mathbb{V}_p} |\{\varphi_i(v) : \varphi_i \text{ an occurrence of } p \text{ in } \mathcal{T}\}|$. Then we know that $\forall \varphi, \varphi' : \varphi(v^*) = \varphi'(v^*)$ there is a harmful overlap of φ and φ' and hence at most one of the occurrences φ and φ' can be a member of the MVC . From this the claim follows. □

Theorem 2. $\sigma_\circ \geq \sigma_\bullet \quad \forall p \in \mathcal{P} : \sigma_\circ(p, \mathcal{T}) \geq \sigma_\bullet(p, \mathcal{T})$

We know that for all φ, φ' such that φ and φ' overlap harmfully, there is a simple overlap. Hence the overlap graphs for both measures have the same nodes, and the edges for the harmful overlap are a subset of the edges for the simple overlap. Thus, the harmful overlap contains less constraints for the MVC , and the set is at least as big as for the simple overlap. □

Finally it is easy to see that all described measures are bounded by the real number of pattern occurrences in the graph.

¹ This paper is an extended version of a paper presented at a workshop without publications [2].

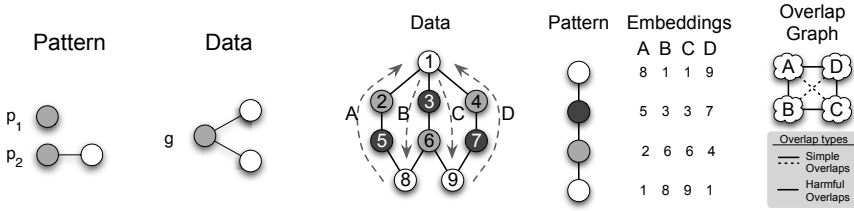


Fig. 1. The support problem in **Fig. 2.** A graph with four different occurrences of a single graph g : p_1 occurs once. p_2 occurs twice. The three discussed measures evaluate to $\sigma_{\bullet} = 1 < \sigma_{\circ} = 2 < \sigma_{\wedge} = 3$.

Table 1. Details of the computations needed to determine the *MIS* support measures. *MIS* could not be computed for pattern 11 and above.

Pattern	1	2	3	4	5	6	7	8	9	10
Nodes in Pattern	2	2	3	3	3	4	4	4	4	5
Image-based support	110	110	100	95	77	97	68	77	64	82
# Occurrences	432	418	1696	1606	815	5428	7380	2254	816	15878
Time for Occurrences	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s	≈ 0s
Edges in overlap-graph	3825	8714	328925	226886	167026	6662049	8362729	1401907	265249	66155623
Time for <i>MIS</i>	1s	1s	31s	57s	4s	958s	2456s	73s	2s	>45m
Size <i>MIS</i>	69	92	42	65	36	24	45	32	62	-

Pattern	11	12	13	14	15	16	17	18	19	20
Nodes in Pattern	5	6	7	8	9	10	11	12	13	14
Image-based support	68	80	69	69	66	66	63	63	62	62
# Occurrences	7988	44254	116580	287954	658540	1386328	2711828	5039624	9125850	16409046
Time for Occurrences	≈ 0s	2s	7s	22s	1m4s	2m56s	7m34s	18m57s	46m14s	110m49s
Edges in overlap-graph	9332671	-	47804219	-	-	-	-	-	-	-

3 Experiments

To compare with the overlap-based support measures from [7,5] we obtained the datasets [\[7\]](#) and [\[5\]](#) described in [7] from the SUBDUE website². We used the same thresholds as used in [7] and obtained for all three measures the same sets of frequent patterns as reported in [7]. A closer look revealed that both datasets are rather [\[7\]](#) than single graphs, consisting only of sets of trees of [\[7\]](#). In the [\[7\]](#) dataset all trees have 21 nodes. A traditional transactional graph miner [10] yields identical results; no additional information on this data can be discovered using single graph mining.

The [\[7\]](#) dataset³ does not have this drawback and consists of four large graphs that correspond to the hyperlink structure of web pages from a computer science department. Nodes are labeled according to the seven types of web-page that they represent. Edges are unlabeled. Figure 4 summarizes the characteristics of the datasets. Table 1 lists details of the computation of the [\[7\]](#) for all patterns found on the Cornell dataset using our measure with a minimum support threshold of 61. [\[3,4\]](#), a state of the art (approximative) solver [8] was used to calculate the [\[7\]](#). The table shows that for larger patterns, the number of occurrences is prohibitive; we were not able to compute

² <http://cygnus.uta.edu/subdue/databases/index.html>

³ <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/theo-20/www/data/>

⁴ <http://www.stasbusygin.org/>

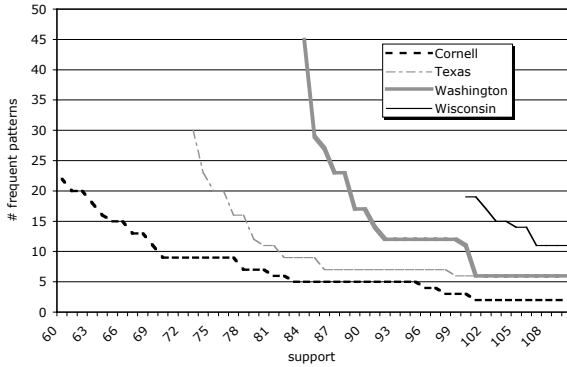


Fig. 3. Patternset sizes for different supports on the 4 WebKB graphs using σ_{\wedge}

Dataset	Number of	
	Nodes	Edges
Cornell	627	1177
Texas	761	1507
Washington	1074	2158
Wisconsin	983	2311

Dataset	Node degree	
	max	avg
Cornell	130	3.75
Texas	149	3.96
Washington	161	4.02
Wisconsin	162	4.70

Fig. 4. Characteristics of the real world datasets used for the experiments

the overlap graph, and consequently could not solve the problem. As usually bigger patterns are of interest, this is a problem for the overlap-based measures.

The results of the experiments on the datasets are summarized in Figure 3. They suggest a relationship between the size of the database and the computational costs of the frequent pattern extraction. Expressing the support relative to the number of nodes in the data graphs, most datasets show the same behavior, except for Washington, where lower relative supports were feasible.

Moreover, we applied our algorithm to a life science database with up to 18.000 nodes and 24.000 edges [9]. However, these results require more research.

4 Conclusions and Future Work

We introduced a new support measure for mining frequent subgraphs in large single graphs and compared it experimentally and theoretically with existing measures. Existing measures are based on constructing overlap graphs, which soon grow impractically large; this makes solving the subsequent NP complete problem impossible. Since the proposed new measure does not suffer from this problem, it can be evaluated in cases where the old measures cannot be evaluated. Furthermore we showed that the new measure is an upper bound for the other measures, allowing us to guarantee a superset of patterns. We believe there are no clear advantages or disadvantages with respect to the interpretability of any of the measures.

We only compared with complete frequent (single)subgraph miners. Further applications may be found in heuristic single graph miners, of which [6] is an example, and graph miners dealing with additional constraints [4].

The proposed support measure is extendible in multiple ways. We present one example here. Our measure was introduced as a node-based support measure and

is easily turned into an edge-based measure. More interestingly, it is possible to generalize our measure to more general substructures than nodes or edges.

Given a parameter k , we can define a support measure based on determining where each connected subgraph with k nodes of the pattern can be matched to.

Definition 4.

$$\sigma_{\wedge}(p, g) = \min_{V \subseteq \mathbb{V}_p, |V|=k, V} |\{\{\varphi_i(V)\} : \varphi_i \in \text{occurrences}(p)\}|$$

Intuitively, we obtain a measure which achieves counts that are closer to the total number of occurrences of a pattern, while the counts are still anti-monotonic. Especially in larger patterns, it is sometimes more intuitive to allow for more overlap between occurrences than when only single nodes or edges are used.

Acknowledgments. We thank Luc De Raedt for helpful comments and discussion, as well as Hannu Toivonen for providing the life sciences database. We thank Christian Borgelt for making the MoSS mining tool publicly available. The authors were supported by the EU FET IST project "Inductive Querying", contract number FP6-516169 and by a doctoral research grant by the KU Leuven research fund.

References

1. Agrawal, R., Mannila, H., Srikant, R., Toivonen, H., Verkamo, A.I.: Fast discovery of association rules. In: *Advances in Knowledge Discovery and Data Mining*, pp. 307–328. AAAI/MIT Press (1996)
2. Bringmann, B., Nijssen, S.: What is frequent in a single graph? In: *International Workshop on Mining and Learning with Graphs (MLG)* (2007)
3. Busygin, S.: A new trust region technique for the maximum weight clique problem. *Discrete Applied Mathematics* 154, 2080–2096 (2006)
4. Chen, C., Yan, X., Zhu, F., Han, J.: gApprox: Mining frequent approximate patterns from a massive network. In: Perner, P. (ed.) *ICDM 2007*. LNCS (LNAI), vol. 4597, Springer, Heidelberg (2007)
5. Fiedler, M., Borgelt, C.: Support computation for mining frequent subgraphs in a single graph. In: *International Workshop on Mining and Learning with Graphs (MLG)* (2007)
6. Holder, L.B., Cook, D.J., Djoko, S.: Substructure discovery in the SUBDUE system. In: *KDD Workshop*, pp. 169–180 (1994)
7. Kuramochi, M., Karypis, G.: Finding frequent patterns in a large sparse graph. *Data Min. Knowl. Discov.* 11(3), 243–271 (2005)
8. Motzkin, T.S., Straus, E.G.: Maxima for graphs and a new proof of a theorem of Turan. *Canadian Journal of Mathematics* 17(4), 533–540 (1965)
9. Sevon, P., Eronen, L., Hintsanen, P., Kulovesi, K., Toivonen, H.: Link discovery in graphs derived from biological databases. In: *Data Integration in the Life Sciences*, vol. 4075 (2006)
10. Yan, X., Han, J.: gSpan: Graph-based substructure pattern mining. In: *ICDM*, pp. 721–724 (2002)

A Cluster-Based Genetic-Fuzzy Mining Approach for Items with Multiple Minimum Supports

Chun-Hao Chen¹, Tzung-Pei Hong², and Vincent S. Tseng¹

¹ Department of Computer Science and Information Engineering,
National Cheng Kung University, Taiwan
chchen@idb.csie.ncku.edu.tw, tsengsm@mail.ncku.edu.tw

² Department of Computer Science and Information Engineering,
National University of Kaohsiung, Taiwan
Department of Computer Science and Engineering
National Sun Yat-sen University, Taiwan
tphong@nuk.edu.tw

Abstract. In the past, we proposed an algorithm for extracting appropriate multiple minimum support values, membership functions and fuzzy association rules from quantitative transactions. The evaluation process might take a lot of time, especially when the database to be scanned could not totally fit into main memory. In this paper, an enhanced approach, called the Cluster-based Genetic-Fuzzy mining approach for items with Multiple Minimum Supports (*CGFMMS*), is thus proposed to speed up the evaluation process and keep nearly the same quality of solutions as the previous one. Experimental results also show the effectiveness and the efficiency of the proposed approach.

Keywords: Data mining, fuzzy set, genetic algorithm, genetic-fuzzy mining, *k*-means clustering, multiple minimum supports, requirement satisfaction.

1 Introduction

Data mining is commonly used for inducing association rules from transaction data [1]. In real applications, different items may have different criteria to judge their importance and quantitative data may exist. We thus divide the fuzzy data mining approaches into two kinds, namely Single-minimum-Support Fuzzy-Mining (*SSF*M) [2, 7, 9] and Multiple-minimum-Support Fuzzy-Mining (*MSF*M) problems [10]. However, the membership functions were usually assumed to be known in advance in the above problems.

For *SSF*M problems, Kaya *et al.* proposed a GA-based approach to derive a predefined number of membership functions for getting a maximum profit within an interval of user specified minimum support values [8]. Hong *et al.* also proposed a genetic-fuzzy data-mining algorithm for extracting both association rules and membership functions from quantitative transactions [6]. For *MSF*M problem, Chen *et al.* proposed a genetic-fuzzy approach for extracting minimum support values, membership functions and fuzzy association rules from quantitative transactions [3]. It evaluated each chromosome by the criterion of requirement satisfaction which was

composed of the number of 1-itemsets and the suitability of membership functions. Although the evaluation only by 1-itemsets was much faster than that by all itemsets or interesting association rules, it was still time-consuming since the database had to be scanned once for each chromosome.

In this paper, the clustering technique is thus used to reduce the execution time in solving the *MSFM* problem. An enhanced approach, called the cluster-based genetic-fuzzy mining algorithm for items with multiple minimum supports (*CGFMMS*), is proposed to speed up the evaluation process and keep nearly the same quality of solutions as that in [3]. The well-known clustering approach, *k*-means, is used to achieve this purpose [11]. Experimental results also show the effectiveness of the proposed approach.

2 The Proposed Framework

The proposed cluster-based genetic-fuzzy mining framework for items with multiple minimum supports (*CGFMMS*) is shown in Fig. 1.

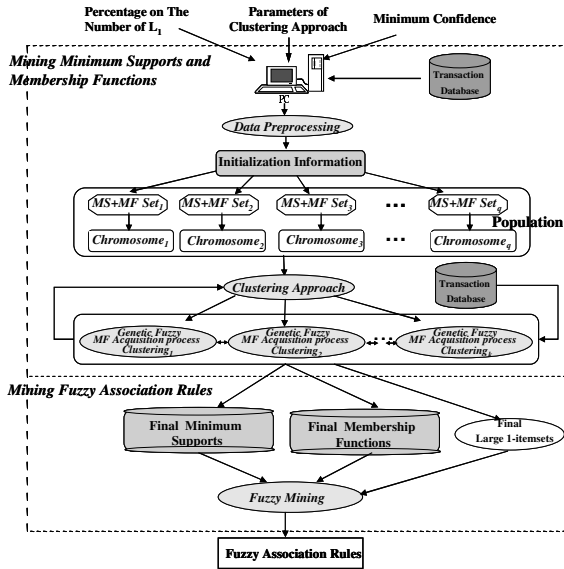


Fig. 1. The proposed CGFMMS framework

It can be divided into two phases, respectively for mining multiple minimum supports and membership functions and for mining fuzzy association rules. In the first phase, the proposed framework maintains a population of sets of minimum support values and membership functions, and uses genetic algorithm to automatically derive the resulting one. Data preprocessing is first done to get initialization information. Initial minimum support values and membership functions are then generated according to the initialization information. The clustering technique is next used to gather similar chromosomes into groups. All the chromosomes then use the requirement

satisfaction derived from the representative chromosomes in the clusters and their own suitability of membership functions to calculate their fitness values. Since the number for scanning a database decreases, the evaluation cost can thus be reduced. The evaluation results are utilized to choose appropriate chromosomes for mating. The offspring sets then undergo recursive evolution until a good set has been obtained. Finally, the derived minimum support values and membership functions are used to mine fuzzy association rules in the second phase. The approach proposed by Lee *et al.* [10] is used here to achieve this purpose.

3 Clustering Chromosomes

In order to develop a good set of minimum support values and membership functions from an initial population, the genetic algorithm selects parent chromosomes for mating in a probabilistic way. An evaluation function is thus used to qualify the derived minimum support values and membership functions. The fitness function of a chromosome C_q is defined as follows:

$$f(C_q) = \frac{RS(C_q)}{Suitability(C_q)} \tag{1}$$

where $RS(C_q)$ is the requirement satisfaction defined as the closeness of the number of derived large 1-itemsets for chromosome C_q to its required number of large 1-itemsets (RNL). The RNL is a criterion used to reflect the user preference on the derived knowledge. The details for calculating RS values can be found in [3]. On the other hand, $suitability(C_q)$ is used to measure the shape suitability of the membership functions. It consists of two factors, namely coverage factor and coverage factor, which is used to reduce the two bad types of membership functions, too redundant and too separate. However, since the transactions must be scanned once for each chromosome to get its requirement satisfaction, it is thus time-consuming. In the past, we proposed a method based on the clustering technique to reduce the evaluation time of large 1-itemsets [4] for *SSF*M problem. In this paper, we thus modify the previous approach to solve the *MSF*M problem. Since in the *MSF*M problem, each item has its own minimum support, using only the two attributes (coverage factor and overlap factor) as used in [4] for clustering is not enough. The average minimum support values of items (called the support factor) will be used as an additional attribute for clustering chromosomes. That is:

$$support_factor(C_q) = \frac{100 * \sum_{j=1}^m \alpha_j}{m} \tag{2}$$

where m is the number of items and α_j is the minimum support value of item I_j . The clustering process is then executed according to the above three attributes. For each cluster, the chromosome which is the nearest to a cluster center is thus chosen as the representative and used to derive its requirement satisfaction. Each chromosome then

estimates its requirement satisfaction by the requirement satisfaction of its representative chromosome. The estimated requirement satisfaction of chromosome C_q which belongs to the cluster $cluster_g$ is defined as:

$$EstimatedRS(C_q) = RS(RepChro_g) \tag{3}$$

where $RepChro_g$ is the representative chromosome of $cluster_g$, and $RS(RepChro_g)$ is the requirement satisfaction of $RepChro_g$. Finally, each chromosome uses its estimated requirement satisfaction and its own suitability of membership functions to calculate its fitness value.

4 The Proposed Mining Algorithm: CGFMMS

The proposed cluster-based genetic-fuzzy mining algorithm for items with multiple minimum supports (CGFMMS):

- Step 1: Generate a population of P individuals.
- Step 2: Calculate the *coverage*, *overlap* and *support factors* of each chromosome.
- Step 3: Divide the chromosomes into k clusters by the k -means clustering approach based on the three attributes. For each cluster g , find the chromosome which is the nearest to the cluster as the representative $RepChro_g$, $1 \leq g \leq k$.
- Step 4: Calculate the requirement satisfaction of each representative chromosome by the following substeps.

Substep 4.1: For each transaction datum D_i , $i=1$ to n , and for each item I_j , $j=1$ to m , transform the quantitative value $v_j^{(i)}$ into a fuzzy set $f_{jk}^{(i)}$ represented as:

$$\left(\frac{f_{j1}^{(i)}}{R_{j1}} + \frac{f_{j2}^{(i)}}{R_{j2}} + \dots + \frac{f_{jl}^{(i)}}{R_{jl}} \right),$$

using the corresponding membership functions represented by the chromosome, where R_{jk} is the k -th fuzzy region (term) of item I_j , $f_{jl}^{(i)}$ is $v_j^{(i)}$'s fuzzy membership value in region R_{jk} , and l is the number of linguistic terms for I_j .

Substep 4.2: For each item region R_{jk} , $1 \leq j \leq m$, calculate its scalar cardinality on the transactions as follows:

$$count_{jk} = \sum_{i=1}^n f_{jk}^{(i)}.$$

Substep 4.3: For each R_{jk} , $1 \leq j \leq m$ and $1 \leq k \leq l$, check whether its $count_{jk}$ is larger than or equal to the minimum support value represented in the chromosome. If R_{jk} satisfies the above condition, put it in the set of large 1-itemsets.

Substep 4.4: Set the requirement satisfaction of each representative chromosome.

- Step 5: Calculate the requirement satisfaction of the representative chromosomes. Calculate the estimated requirement satisfaction of the other chromosomes by Formula (3).

Step 6: Calculate the fitness value of each chromosome by the following formula:

$$f(C_q) = \frac{EstimatedRS(C_q)}{Suitability(C_q)}$$

- Step 7: Execute the crossover operation on the population.
- Step 8: Execute the mutation operation on the population.
- Step 9: Calculate the fitness values of chromosomes by using STEPS 2 to 6.
- Step 10: Use the selection operation to choose individuals for the next generation.
- Step 11: If the termination criterion is not satisfied, go to Step 7; otherwise, do the next step.
- Step 12: Find the chromosome with the highest fitness value. Get the set of minimum support values and membership functions contained in it.
- Step 13: Mine fuzzy association rules using the set of minimum support values and membership functions.

5 Experimental Results

In this section, experiments made to show the performance of the proposed approach are described. 64 items and 10000 transactions were used in the experiments. The population size was set at 50, the crossover rate was set at 0.8, and the mutation rate was set at 0.001. The parameter d of the MMA crossover operator was set at 0.35 according to Herrera *et al.*'s paper [5]. The percentage of the required number of large 1-itemsets was set at 0.8. Experiments were made to compare the proposed method (CGFMMS) with our previous one (GFMM S) [3] for showing the effect of using clusters in evaluation. The average fitness values of the chromosomes along with different numbers of generations for different numbers of clusters are shown in Fig. 2.

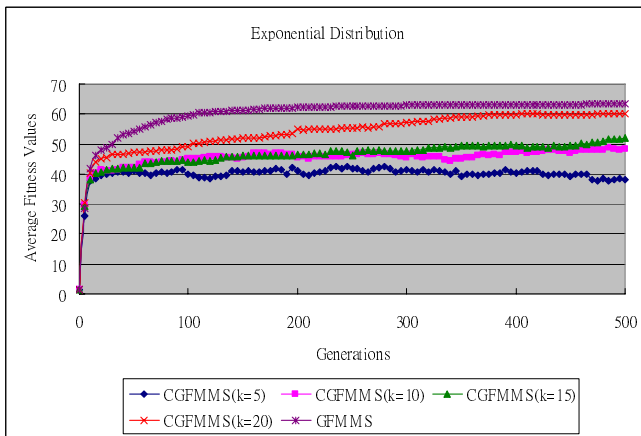


Fig. 2. The comparison between the proposed and the previous approaches

From Fig. 2, it can be observed that the average fitness values by the proposed approach were a little less than those by the previous one. They were close when the number of clusters increased. The results were reasonable since the proposed

approach just estimated the requirement satisfaction of chromosomes. The comparisons for the execution time of the two approaches with different numbers of clusters were also made. The results showed that the proposed approach ran much faster than the previous one. It can thus be concluded that the proposed approach can get a good trade-off between accuracy and execution time.

6 Conclusion and Future Works

In this paper, we have proposed a cluster-based genetic-fuzzy mining algorithm for extracting multiple minimum support values, membership functions and fuzzy association rules from quantitative transactions. The evaluation cost can be significantly reduced due to the time-saving in finding requirement satisfaction. The experimental results show that using the clustering technique to speed up the evaluation process can not only get nearly the same fitness values as the previous approach, but can also significantly reduce execution time. In the future, we will continuously enhance the cluster-based genetic-fuzzy mining framework for more complex problems.

References

1. Agrawal, R., Srikant, R.: Fast algorithm for mining association rules. In: The International Conference on Very Large Databases, pp. 487–499 (1994)
2. Chan, C.C., Au, W.H.: Mining fuzzy association rules. In: The Conference on Information and Knowledge Management, pp. 209–215 (1997)
3. Chen, C.H., Hong, T.P., Tseng, V.S., Lee, C.S.: A genetic-fuzzy mining approach for items with multiple minimum supports. In: The IEEE International Conference on Fuzzy Systems (2007)
4. Chen, C.H., Tseng, V.S., Hong, T.P.: Cluster-based evaluation in fuzzy-genetic data mining. In: The IEEE Transactions on Fuzzy Systems (accepted and to appear)
5. Herrera, F., Lozano, M., Verdegay, J.L.: Fuzzy connectives based crossover operators to model genetic algorithms population diversity. *Fuzzy Sets and Systems* 92(1), 21–30 (1997)
6. Hong, T.P., Chen, C.H., Wu, Y.L., Lee, Y.C.: A GA-based fuzzy mining approach to achieve a trade-off between number of rules and suitability of membership functions. *Soft Computing* 10(11), 1091–1101 (2006)
7. Hong, T.P., Kuo, C.S., Chi, S.C.: Mining association rules from quantitative data. *Intelligent Data Analysis* 3(5), 363–376 (1999)
8. Kaya, M., Alhaji, R.: A clustering algorithm with genetically optimized membership functions for fuzzy association rules mining. In: The IEEE International Conference on Fuzzy Systems, pp. 881–886 (2003)
9. Kuok, C., Fu, A., Wong, M.: Mining fuzzy association rules in databases. *SIGMOD Record* 27(1), 41–46 (1998)
10. Lee, Y.C., Hong, T.P., Lin, W.Y.: Mining fuzzy association rules with multiple minimum supports using maximum constraints. In: Negoita, M.G., Howlett, R.J., Jain, L.C. (eds.) *KES 2004. LNCS (LNAI)*, vol. 3214, pp. 1283–1290. Springer, Heidelberg (2004)
11. MacQueen, J.B.: Some Methods of Classification and Analysis of Multivariate Observations. In: *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability*, pp. 281–297 (1967)

A Selective Classifier for Incomplete Data

Jingnian Chen^{1,2}, Houkuan Huang¹, Fengzhan Tian¹, and Shengfeng Tian¹

¹ School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China

jnchen06@163.com

² Dept. of Information and computing Science, Shandong University of Finance, Jinan, Shandong, 250014, China

Abstract. Classifiers based on feature selection (selective classifiers) are a kind of algorithms that can effectively improve the accuracy and efficiency of classification by deleting irrelevant or redundant attributes of a data set. Due to the complexity of processing incomplete data, however, most of them deal with complete data. Yet actual data are often incomplete and have many redundant or irrelevant attributes. So constructing selective classifiers for incomplete data is an important problem. With the analysis of main methods of processing incomplete data for classification, a selective classifier for incomplete data named RBSR (Relief algorithm-Based Selective Robust Bayes Classifier), which is based on the Robust Bayes Classifiers (RBC) and Relief algorithm, is presented. The proposed algorithm needs no assumptions about data sets that are necessary for previous methods of processing incomplete data in classification. This algorithm can deal with incomplete data sets with many attributes and instances. Experiments were performed on twelve benchmark incomplete data sets. We compared RBSR with the very effective RBC and several other classifiers for incomplete data. The experimental results show that RBSR can not only enormously reduce the number of redundant or irrelevant attributes, but greatly improve the accuracy and stability of classification as well.

1 Introduction

Selective classifiers are a kind of algorithms that can effectively improve the accuracy and efficiency of classification by deleting irrelevant or redundant attributes of a data set. Due to the complexity of processing incomplete data, however, most of them deal with complete data. Yet actual data are often incomplete and have many redundant or irrelevant attributes. So methods about constructing selective classifiers for incomplete data deserve more attention. In order to do some research on this problem, we begin with a review of main methods for processing incomplete data in classification.

Classifiers such as Naïve Bayes classifiers [1] and C4.5 [2] often take two simple methods to deal with incomplete data: ignoring the instances with unknown entries or ascribing these unknown entries to a specified dummy state. Both methods are known to incur potentially dangerous biases in the estimates, see [3] for a detail.

Another method of dealing with incomplete data in classification that can avoid the dangerous biases is to use the EM algorithm [4], Gibbs sampling [5] or gradient descent [6]. But this method relies on the assumption that data are *Missing at Random*

(MAR) [7] and there is no way to verify that a particular data set is actually MAR. Furthermore, this method will suffer of a dramatic decrease in accuracy of estimation when this assumption is violated, which makes the performance of the resulting classifier degenerate [8]. Recently, Williams etc. proposed a method for classifying incomplete data with logistic regression [9]. This method can avoid imputation for the missing data, but it also relies on the MAR assumption.

To avoid the MAR assumption, Ramoni et al. introduced the *Robust Bayes Classifier* (RBC) [10] that needs no assumption about missing data mechanism. RBC uses intervals bounding all the estimates of related probabilities that can be obtained from all the possible completions of an incomplete data set. So it can avoid to some extent potentially dangerous biases in the estimates. As a whole, RBC performs better than classifiers above-mentioned, but it makes the assumption that attributes are independent in each class, which can degrade the performance of classification when violated.

By analyzing the methods above-mentioned, we present a selective classifier for incomplete data: RBSR. An important part of RBSR is the selective Robust Bayes classifier (SRBC), which is described in Section 2. Section 3 describes RBSR. To show its effectiveness, Section 4 presents experiments on twelve benchmark incomplete data sets. At last, conclusions are given in Section 5.

2 SRBC

To construct SRBC on an incomplete data set D whose attribute set is $A = \{A_1, A_2, \dots, A_N\}$ is to establish a RBC on a selected attribute subset that can improve the performance of classification most. This selected attribute subset is acquired by searching the space of attribute subsets of D . In the search process, we evaluate each alternative attribute subset S in terms of the accuracy $f(S)$ of the RBC constructed on S . We also adopt the forward best first search method [11] to get a better search result with lower computational complexity. In the search process a queue Q is kept to hold attribute subsets that had ever been the best or is currently the best. Whether the search process is terminated depends on a threshold T . Specifically, if continuously extending head nodes of Q for T times can not improve the current best performance yet, then terminate search. The process of constructing SRBC is described as follows:

(1) Initialization:

Initialize T ; $t \leftarrow 0$; $A_s \leftarrow \arg \max_{1 \leq i \leq N} \{f(\{A_i\})\}$; Set the current best attribute subset $S_b \leftarrow \{A_s\}$; Set the current best performance $f_{\max} \leftarrow f(\{A_s\})$; Add S_b as a new node to Q ;

(2) If $t < T$, perform step 3), 4) and 5), otherwise, go to 6);

(3) Take out the head node of Q denoted S_h ; $added \leftarrow false$;

For each attribute $A_i \in A - S_h$

{

if $S_h \cup \{A_i\}$ has not been evaluated and $f(S_h \cup \{A_i\}) > f_{\max}$

{

```

    added ← true ; t ← 0 ; Sb ← Sh ∪ {Ai} ; fmax ← f(Sh ∪ {Ai}) ; add
    Sh ∪ {Ai} as a new node to Q ;
  }
}
(4) If the value of added is still false , t ← t+1 ;
(5) Return to step (2) ;
(6) Construct RBC on the final best attribute subset Sb .

```

In the process of constructing SRBC, once an attribute subset is evaluated, a RBC needs to be constructed on it. So the computational complexity of SRBC is often high, especially when the number of attributes or the number of instances of an incomplete data set is larger. In the following section, the more efficient RBSR is presented.

3 RBSR

The key of RBSR is to combine SRBC with ReliefF algorithm that is an extension of Relief. As the space is constrained, we refer readers interested in details of ReliefF to [12]. ReliefF can efficiently delete attributes irrelevant to class variable. However, it does not help with removing redundant attributes [13]. On the other hand, SRBC can effectively select attribute subsets with little redundancy, but its computational complexity is high. So, if firstly irrelevant attributes are deleted with ReliefF and then redundant attributes are removed with SRBC, not only the selected attribute subset by ReliefF can be refined, but also the efficiency of SRBC can be improved. This is the key idea of RBSR. Denote the distance between values of A_i for instances e₁ and e₂ as d(A_i, e₁, e₂), the process of establishing RBSR can be described as follows:

(1) Initialization:

Set the number *n* of instances to be randomly selected, the number of nearest neighbors *k* and the number *N* of attributes to be selected with ReliefF. For each attribute A_i, set its weight W(A_i) ← 0.0 ; set variable *t* ← 0 ;

(2) If *t* ≥ *n*, go to step(4), otherwise do the following operation: Randomly select an instance *r* ; Find *k* nearest neighbors *h_j* in the class of *r* ; From each class *c* ≠ class(*r*) find *k* nearest neighbors *m_j(c)*, *j* = 1, 2, ..., *k* ; For each A_i, let

$$W(A_i) \leftarrow W(A_i) - \sum_{j=1}^k \frac{d(A_i, r, h_j)}{n \times k} + \sum_{c \neq \text{class}(r)} \sum_{j=1}^k \left[\frac{P(c)}{1 - P(\text{class}(r))} \times \frac{d(A_i, r, m_j(c))}{n \times k} \right] ;$$

(3) *t* ← *t* + 1 ; return to step (2) ;

(4) Select the attribute subset S_N consisting of *N* attributes with largest weights ;

(5) Construct SRBC on S_N .

It should be pointed out that RBC (So, SRBC and RBSR) deals only with nominal attributes. Continuous attributes need to be discretized in advance.

4 Experimental Results

In order to testify the validity of RBSR we carried out experiments on twelve benchmark data sets all with missing data. These data sets can be downloaded from the UCI machine learning repository [14]. Description of these data sets is given in Table 1.

Table 1. Data sets used in the experiments

Names	Instances	Classes	Attributes	Names	Instances	Classes	Attributes
Annealing	798	5	38	Cylinder	512	2	39
Arrhythmia	452	16	279	Echocardiogram	132	2	12
Audiology	200	2	70	Horse-colic	368	2	27
B.cancer	699	2	10	L.cancer	32	3	56
Bridges	108	6	12	Mushroom	8124	2	22
Credit	690	2	15	Vote	435	2	16

All our experiments were implemented in the weka system [15]. The implementation was performed on an Intel Pentium IV CPU running at 2.93 GHz and 1GB RAM. Numerical attributes were discretized with “weak.filters.Discretizefilter”.

In all the experiments, the parameters T and k are set to their default values in weak system. Each attribute subset is evaluated with a 5-fold cross validation. The parameters n and N are decided experimentally. On most data sets n is set to 30 except on Arrhythmia where n is set to the number of all instances and on Horse-colic set to 200.

Table 2. The average accuracy of RBC, SRBC and RBSR

Data sets	RBC	SRBC	RBSR
Annealing	95.96 \pm 0.31	91.59 \pm 0.12	96.31 \pm 0.33
Arrhythmia	72.77 \pm 0.89	75.01 \pm 0.62	74.66 \pm 0.62
Audiology	67.99 \pm 0.79	76.53 \pm 0.41	77.24 \pm 0.48
B.cancer	97.11 \pm 0.16	97.31 \pm 0.11	97.31 \pm 0.11
Bridges	61.62 \pm 2.20	66.10 \pm 1.02	66.10 \pm 1.02
Credit	86.18 \pm 0.40	86.65 \pm 0.30	86.17 \pm 0.22
Cylinder	71.36 \pm 0.48	76.02 \pm 0.55	75.80 \pm 0.51
Echocardiogram	98.36 \pm 0.87	97.26 \pm 0.00	97.95 \pm 0.72
Horse-colic	85.20 \pm 0.59	88.09 \pm 0.39	88.58 \pm 0.09
L.cancer	56.13 \pm 1.67	80.32 \pm 3.86	88.06 \pm 2.18
Mushroom	95.96 \pm 0.02	99.68 \pm 0.04	99.68 \pm 0.04
Vote	90.25 \pm 0.19	96.31 \pm 0.00	96.31 \pm 0.00
Average	81.57 \pm 0.71	85.91 \pm 0.62	87.01 \pm 0.53

N is set to $T_a/4+5$ on most data sets except on B.cancer where N is set to $T_a/4+7$ and on Horse-colic set to $T_a/4+10$. Here T_a represents the number of attributes of each data set. Although some of these configurations may not be optimal, RBSR can still significantly improve the performance of RBC and that of SRBC.

In order to compare the performance of RBC, SRBC and RBSR, we run 10 replicates of a 10-fold cross validation on each data set. The average classification accuracy on each data set and corresponding standard deviation are listed in Table 2.

Table 3. The runtime and number of selected attributes of RBSR and SRBC

Data sets	Total attributes	Selected attributes		Runtime (second)	
		SRBC	RBSR	SRBC	RBSR
Annealing	38	8	11	69.91	19.84
Arrhythmia	279	11	9	676.75	181.05
Audiology	70	12	14	269.66	15.86
B.cancer	10	9	9	8.70	7.03
Bridges	12	6	6	3.02	1.20
Credit	15	10	6	15.77	4.31
Cylinder	39	8	4	61.44	10.27
Echocardiogram	12	3	3	2.33	1.49
Horse-colic	27	5	6	13.59	10.03
L.cancer	56	5	6	4.16	2.48
Mushroom	22	3	3	110.81	47.70
Vote	16	3	3	3.11	1.67
Summation	602	83	80	1239.3	302.93

As shown in Table 2, on most of the twelve data sets the classification accuracy of RBSR is much higher than that of RBC and SRBC. Especially on L.cancer, its classification accuracy is 31.93% higher than that of RBC and 7.74% higher than that of SRBC. The Average of all accuracies of RBSR on twelve data sets is 5.44% higher than that of RBC and 1.1% higher than that of SRBC. In addition, it can be seen that the standard deviation of RBSR is lower than that of RBC and SRBC as a whole. This shows that RBSR performs more stably than RBC and SRBC.

The most important is that the efficiency of RBSR is much higher than that of SRBC. Table 3 presents the runtime and the number of selected attributes of these two classifiers on each of the 12 data sets.

From Table 3 it can be seen that on each of the twelve data sets the runtime of RBSR is much less than that of SRBC. It can also be seen that both SRBC and RBSR can sharply reduce the number of irrelevant attributes on all the twelve data sets, and so can greatly simplify the data sets and classifiers. Especially on Arrhythmia, the number of attributes is reduced from 279 to 11 by SRBC and to 9 by RBSR. The number of all attributes selected by RBSR on these 12 data sets is 80 and that by SRBC is 83.

It should be pointed out that RBSR was also compared with other two classifiers: the one combining ReliefF with RBC and the one proposed in [9], and it still performs better than these two. For the space constraint we omit the details.

5 Conclusion

The problem of constructing selective classifiers for incomplete data is important and deserves more attention. By analyzing main methods that have been proposed for processing incomplete data in classification, this paper presents a selective Bayes classifier for incomplete data. At first, the selective classifier SRBC is described. Then, based on SRBC and ReliefF algorithm, the more effective selective classifier RBSR is presented. The proposed RBSR needs no assumption about data sets that are necessary for previous methods of processing incomplete data in classification. Experiments on twelve benchmark incomplete data sets show that RBSR can greatly improve the accuracy and stability of classification. Furthermore, it can also sharply reduce the number of irrelevant or redundant attributes.

Acknowledgements

This work was supported by National Natural Science Foundation of China under Grant No. 60503017 and No. 60673089. The authors would like to thank the anonymous referees for their valuable comments and suggestions.

References

1. Duda, R.O., Hart, P.E.: *Pattern Classification and Scene Analysis*. Wiley, New York (1973)
2. Quinlan, J.R.: *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Francisco (1993)
3. Kohavi, R., Becker, B., Sommerfield, D.: Improving simple Bayes. In: van Someren, M., Widmer, G. (eds.) *Poster Papers of the ECML-97*, pp. 78–87. Charles University, Prague (1997)
4. Dempster, A.P., Laird, D., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm (with discussion), *J. J. Royal Statist. Soc. Ser. B* 39, 1–38 (1977)
5. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. Pattern Analysis and Machine Intelligence* 6, 721–741 (1984)
6. Russell, S., Binder, J., Koller, D., Kanazawa, K.: Local learning in probabilistic networks with hidden variables. In: *Proc. IJCAI 1995, Montreal, Quebec*, pp. 1146–1151. Morgan Kaufmann, San Francisco (1995)
7. Little, R.J.A., Rubin, D.B.: *Statistical Analysis with Missing Data*. Wiley, New York (1987)
8. Spiegelhalter, D.J., Cowell, R.G.: Learning in probabilistic expert systems. In: Bernardo, J., Berger, J., Dawid, A.P., Smith, A.F.M. (eds.) *Bayesian Statistics*, vol. 4, pp. 447–466. Oxford University Press, Oxford (1992)
9. Williams, D., Liao, X., Xue, Y., Carin, L., Krishnapuram, B.: On classification with incomplete data. *IEEE Trans. Pattern Analysis and Machine Intelligence* 29(3), 427–436 (2007)

10. Ramoni, M., Sebastiani, P.: Robust Bayes classifiers. *Artificial Intelligence* 125(1-2), 209–226 (2001)
11. Winston, P.H.: *Artificial intelligence*. Addison-Wesley, Reading (1992)
12. Kononenko, I.: Estimating attributes: Analysis and extensions of Relief. In: Raedt, L.D., Bergadano, F. (eds.) *Machine Learning: ECML 1994*, pp. 171–182. Springer, Heidelberg (1994)
13. Kira, K., Rendell, L.: The feature selection problem: Traditional methods and a new algorithm. In: *Proc. AAAI 1992*, pp. 129–134. AAAI Press, Menlo Park (1992)
14. Hettich, S., Blake, C.L., Merz, C.J.: *UCI Repository of machine learning databases*, Department of Information and Computer Sciences, University of California, Irvine, CA (1998), <http://www.ics.uci.edu/~mlearn/>
15. Witten, I.H., Frank, E.: *Data Mining—Practical Machine Learning Tools and Techniques*, 2nd edn. Morgan Kaufmann, San Francisco (2005)

Detecting Near-Duplicates in Large-Scale Short Text Databases

Caichun Gong^{1,2}, Yulan Huang^{1,2}, Xueqi Cheng¹, and Shuo Bai¹

¹Institute of Computing Technology, Chinese Academy of Sciences, Beijing, 100080, P.R.C.

²Graduate School of Chinese Academy of Sciences, Beijing, 100049, P.R.C.

gongcaichun@software.ict.ac.cn, huangyulan@software.ict.ac.cn,
cxq@ict.ac.cn, bs_7799@hotmail.com

Abstract. Near-duplicates are abundant in short text databases. Detecting and eliminating them is of great importance. SimFinder proposed in this paper is a fast algorithm to identify all near-duplicates in large-scale short text databases. An ad hoc term weighting scheme is employed to measure each term's discriminative ability. A certain number of terms with higher weights are selected as features for each short text. SimFinder generates several fingerprints for each text, and only texts with at least one fingerprint in common are compared with each other. An optimization procedure is employed in SimFinder to make it more efficient. Experiments indicate that SimFinder is an effective solution for short text duplicate detection with almost linear time and storage complexity. Both precision and recall of SimFinder are promising.

Keywords: duplicate detection, short text, term weighting, optimization.

1 Introduction

The rapid technological improvements in Internet and telecommunication have led to an explosion of digital data. A large proportion of such data are short texts, such as mobile phone short messages, instant messages. It is reported that more than 1.58 billion mobile phone short messages are sent each day in Mainland China [1]. Tencent QQ has attracted more than 430 million users, and billions of instant messages are sent each day [2].

Duplicates are abundant in short text databases. In our investigation, more than 40% mobile phone short messages have at least one identical duplicate, and an even larger proportion of them are near-duplicates. Detecting and eliminating these duplicate short messages is of great importance for other short text language processing, such as clustering, opinion mining, topic detection and tracking, community uncovering. Identical duplicate short texts are easy to detect by standard hashing schemes. Identification of near-duplicate short texts is much more difficult because of the following reasons: First of all, a single short text contains usually less than 200 characters, which makes it difficult to extract effective features. Second, there are usually a huge number of texts in a short text database. Third, Informal abbreviations, transliterations and network languages are prevailing in short text databases [2].

In this paper, an algorithm called SimFinder is presented to detect near-duplicates in large-scale short text databases. An ad hoc weighting scheme is employed in SimFinder to make duplicate measure more precise. Only a few terms with higher weights are extracted as features. Fingerprints are generated from these features, and only short texts with the same fingerprint will compare with each other. An optimization solution is also proposed to reduce comparisons further.

2 Related Work

A variety of techniques have been developed to identify academic plagiarism [3,4,5,6], web page duplicates [7,8,9,10], duplicate database records [11,12]. Brin et al. have proposed a prototype system called *COPS* (COpy Protection System) to safeguard intellectual property of digital documents [3]. Shivakumar et al. have developed *SCAM* (Stand Copy Analysis Mechanism) as a part of the Stanford Digital Library project [4]. Broder finds it sufficient to keep each document a “sketch” of “shingles” to compute the resemblance of two documents. Any document pair with at least one common shingle is examined whether it exceeds the threshold for resemblance. Broder’s shingling method works well on duplicate detection in AltaVista search engine [8].

Lyon et al. have investigated the theoretical background to automated plagiarism detection [5]. They observe that independently written texts have a comparatively low level of matching trigrams. The Ferret plagiarism system counts matching trigrams of a pair of documents [5,6]. Shivakumar presents two approaches to compute overlap between all web document pairs simultaneously. Both of them assume that only when document d_i and d_j share more than k fingerprints can they be candidate near-duplicates, where k is a predefined threshold [7].

Manku et al. show that Charikar’s *simhash* [13] is practically useful for identifying near-duplicates in large-scale web page repository [9]. *Simhash* is a fingerprint technique enjoying the property that fingerprints of near-duplicates differ only in a small number of bit positions [9,13]. If the *simhash* fingerprints of two documents are similar, they are deemed to be near-duplicates.

3 SimFinder

As for a large-scale short text database, it is impossible to detect near-duplicates by comparing texts with each other. A certain number of fingerprints are extracted from each text in SimFinder, and only short texts sharing same fingerprints are possible to be near-duplicates.

3.1 Term Weighting and Duplicate Degree

Terms play different roles in texts. Generally speaking, nouns, verbs and adjectives are more discriminative than adverbs, connectives, pronouns and numerals. It is improper to assign a same weight to all terms [14]. Since few terms will occur more than one time in a single short text, the traditional *tf-idf* scheme is inappropriate for short texts.

As for each set G of terms with the same part-of-speech, an empirical weight interval $[a,b]$ is associated in SimFinder, where a and b are the minimal and maximal weight

that may be assigned to terms in G respectively. Let weight interval of G be $[a,b]$, a simple linear interpolation is used to compute the weight of each term $t \in G$ as follows:

$$W(t) = \frac{(b - a)(F(t) - F')}{F - F'} + a \tag{1}$$

Where $F(t)$ is the frequency of term t in the background database, F and F' denote the frequency of the most and least frequently used term in G respectively. The weighting scheme of Equation 1 does not take term length into account. We notice that longer terms are usually more important than shorter terms. Let $|t|$ denote the length of term t , the long-term-preferred weighting scheme can be defined as follows:

$$W(t) = \frac{(b - a)(F(t) - F')}{F - F'} |t| + a |t| \tag{2}$$

Duplicate degree is a measure of similarity between two texts. Texts with duplicate degree higher than a predefined threshold θ are considered as near-duplicates. Let A and B be two texts, the standard duplicate degree called *Jaccard similarity* is defined as follows:

$$d(A, B) = \frac{|S(A) \cap S(B)|}{|S(A) \cup S(B)|} \tag{3}$$

Where $S(A)$ and $S(B)$ are the set of terms contained in text A and B respectively. All terms are considered as equal importance in Equation 3. Let $w(t_i)$ be the weight of term t_i , the weighted variant of duplicate degree can be defined as follows:

$$d(A, B) = \frac{\sum_{t_i \in S(A) \cap S(B)} w(t_i)}{\sum_{t_j \in S(A) \cup S(B)} w(t_j)} \tag{4}$$

3.2 Feature Extraction and Optimization

Since there are no blanks to mark words in Chinese texts, SimFinder segments each short text into a serial of terms. Terms are then sorted in descending order of their weights. Terms with higher weights are called discriminative terms and selected as features.

Remark 1: In real short text databases, when two short texts A and B are near-duplicates, most discriminative terms occur in both A and B . Near-duplicates differ usually only in connectives, pronouns, numerals, and punctuations.

Each N contiguous features (or called N -gram) are hashed into an integer as fingerprint. If two short texts A and B have no fingerprints in common, they are impossible to be near-duplicates. As a result, numerous unnecessary comparisons can be avoided.

As for text A with m terms, no more than $\lambda=k+N$ terms are necessary to be selected as features, where k is the minimal integer satisfying the following inequality:

$$\frac{\sum_{i=1}^k w(t_i)}{\sum_{i=1}^m w(t_i)} > \theta \tag{5}$$

Where $w(t_i)$ denotes the weight of term t_i and θ is the duplicate degree threshold.

Definition 1: Let $D = \{T_1, T_2, \dots, T_n\}$ be the text database, and ArB denote A is duplicated to B , then $R = \{(A, B) \mid ArB, A \in D, B \in D\}$ is called a duplicate relation on D . R is called a transitive duplicate relation if and only if $\forall (A, B) \in R, (B, C) \in R \Rightarrow (A, C) \in R$.

Remark 2: In real short text databases, duplicate transitivity holds in almost all cases. In other words, if ArB and BrC , A and C are near-duplicates in almost all cases.

If ArB and BrC , A and C are called a potential duplicate pair. In traditional text databases, duplicate relation does not always observe transitivity. While almost all short text databases satisfy Remark 2. With Remark 2, potential duplicate pairs can be safely regarded as near-duplicates, so the computation of duplicate degree is unnecessary.

4 Experiments and Evaluations

Various experiments have been conducted with two short text databases. One is a short message corpus composed of 12 million mobile phone short messages (735 megabytes), the other is a BBS title corpus with 5 million BBS titles (157 megabytes).

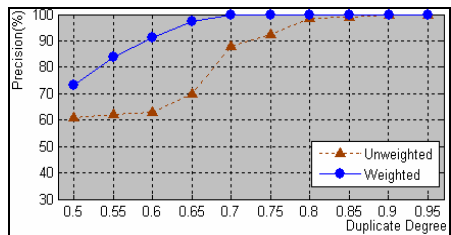
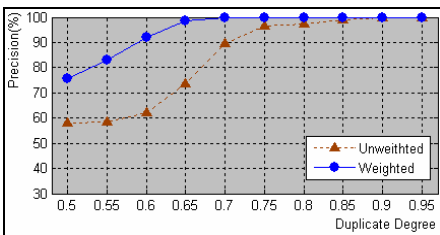


Fig. 1. The precision on short message corpus

Fig. 2. The precision on BBS title corpus

Before we verify the effectiveness of SimFinder, a proper duplicate degree threshold must be determined. For each duplicate degree $d = 0.50 + 0.05i$ ($1 \leq i \leq 10$), 200 pairs of candidate duplicate short texts with duplicate degree in interval $[d - 0.05, d]$ are

selected randomly and are checked manually whether they are near-duplicates. The precision of Equation 3 and Equation 4 are shown in Figure 1 and Figure 2. Experiments indicate that Equation 4 is more effective than Equation 3. The duplicate degree 0.65 is selected as the threshold because the precision is acceptable in both the short message corpus and the BBS title corpus.

A base-line algorithm is employed to generate all possible near-duplicate pairs. Texts with at least two continuous words in common are compared with each other. One million short messages with no identical duplicates have been used to choose gram size and feature number. The recall of algorithm A is defined as the ratio of the number of duplicate text detected by algorithm A to the number of duplicate text detected by the base-line algorithm. Figure 3 shows the effect of gram size on recall, and Figure 4 shows the effect of gram size on efficiency. $N=3$ is selected in SimFinder because the recall is acceptable and the efficiency is promising.

Let $\lambda = k+N$, where k is defined in Ineqation 5, and N has been determined to be 3. Figure 5 and Figure 6 show the effect of feature number on recall and efficiency respectively. As can be seen that the feature number computed as Ineqation 5 is feasible since the recall is almost 1. More features are unnecessary because the recall increases very little.

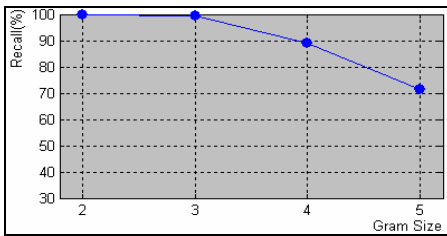


Fig. 3. The effect of gram size on recall

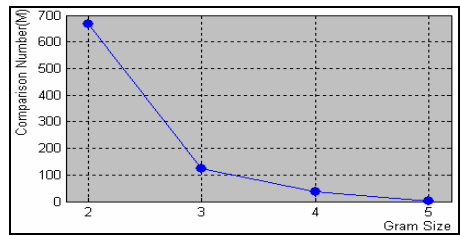


Fig. 4. The effect of gram size on comparison number

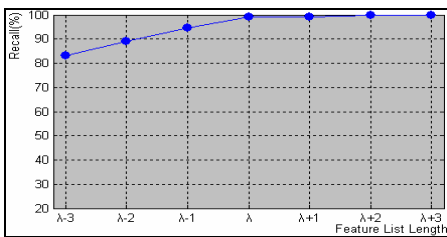


Fig. 5. The effect of feature number on recall

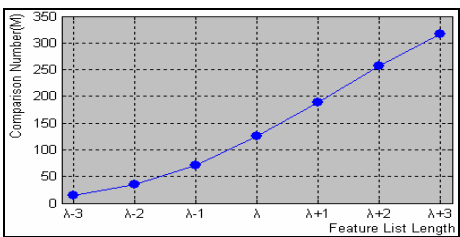


Fig. 6. The effect of feature number on comparison number

Ten thousand potential duplicate pairs are selected randomly to verify the correctness of Remark 2. For each potential duplicate pair (A,B) , duplicate degree $d(A,B)$ is computed using Equation 4. Only 23 of them are less than 0.65, So the optimization has very little negative effect on precision. As for the short message corpus, if no

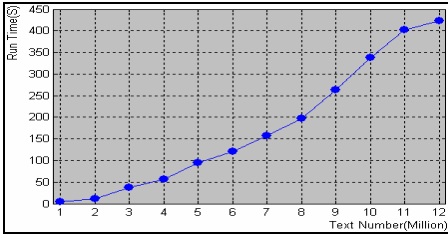


Fig. 7. Run times on short message corpus

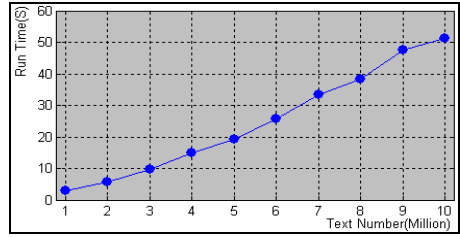


Fig. 8. Run times on BBS title corpus

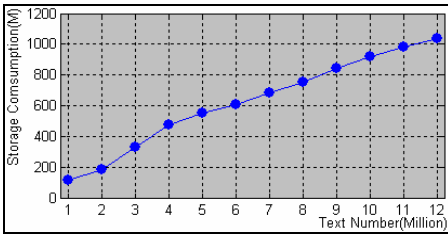


Fig. 9. Storage consumption on short message corpus

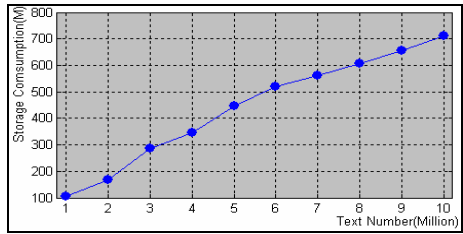


Fig. 10. Storage consumption on BBS title corpus

optimization procedure is included, the duplicate degree of 642,404,813 duplicate pairs must be computed using Equation 4. When optimization procedure is included, only 120,725,627 comparisons are needed. The optimization procedure increases the efficiency of SimFinder more than 4 times.

The SimFinder has been implemented in C++. We use a dawning server S4800A with 4 CPUs and 8G bytes of memory to test the performance of SimFinder. Figure 7 and Figure 8 show the run time of SimFinder on short message corpus and BBS title corpus respectively. Figure 9 and Figure 10 show the storage consumption of SimFinder. As can be seen that both run time and storage consumption are almost linear correlated with the size of corpus.

5 Conclusion

SimFinder is an effective and efficient algorithm to detect and eliminate duplicates in large-scale short text databases. Three techniques have been included in SimFinder: the ad hoc term weighting technique, the discriminative-term selection technique, the optimization technique. Experiments have shown that SimFinder is an encouraging solution for large-scale short text duplicate detection.

Acknowledgments. This research is supported by *The 973 National Basic Research Program of China* under the Grant NO. 2004CB318109 and 2007CB311100.

References

1. Website of Ministry of Information Industry of China, <http://www.mii.gov.cn/>
2. Hu, J.X.: Message text clustering based on frequent patterns (In Chinese). M.S. thesis, Institute of Computing Technology, Chinese Academy of Sciences. Beijing, China (2006)
3. Brin, S., Davis, J., Garcia-Molina, H.: Copy detection mechanisms for digital documents. In: Proceedings of the ACM SIGMOD Annual Conference, San Francisco, CA (May 1995)
4. Shivakumar, N., Garcia-Molina, H.: SCAM:A copy detection mechanism for digital documents. In: Proceedings of 2nd International Conference in Theory and Practice of Digital Libraries, Austin, Texas (June 1995)
5. Lyon, C., Barrett, R., Malcolm, J.: A theoretical basis to the automated detection of copying between texts, and its practical implementation in the Ferret plagiarism and collusion detector. In: Plagiarism: Prevention, Practice and Policies Conference (June 2004)
6. Lyon, C., Barrett, R., Malcolm, J.: Plagiarism is easy, but also easy to detect. *Plagiarism: Cross-Disciplinary Studies in Plagiarism, Fabrication, and Falsification* 1(5), 1–10 (2006)
7. Shivakumar, N., Garnia-Molina, H.: Finding near-replicas of documents on the web. In: Proceedings of Workshop on Web Databases, Valencia, Spain (March 1998)
8. Broder, A.: Identifying and Filtering Near-Duplicate Documents. In: Proceedings of the 11th Annual Symposium on Combinatorial Pattern Matching, Montreal, Canada (June 2000)
9. Manku, G.S., Jain, A., Sarma, A.D.: Detecting near-duplicates for web crawling. In: Proceedings of the 16th International World Wide Web Conference, Banff, Alberta, Canada (May 2007)
10. Henzinger, M.: Finding near-duplicate web pages: A large-scale evaluation of algorithms. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Seattle, Washington, U.S.A (August 2006)
11. Tian, Z.P., Lu, H.J., Ji, W.Y., et al.: An n-gram-based approach for detecting approximately duplicate database records. *International Journal on Digital Libraries* 5(3), 325–331 (2001)
12. Hernandez, M.A., Stolfo, S.J.: The merge/purge problem for large databases. In: Proceedings of the 1995 ACM SIGMOD International Conference on Management of Data, San Jose, CA, U.S.A (1995)
13. Charikar, M.: Similarity estimation techniques from rounding algorithms. In: Proceedings of 34th Annual Symposium on Theory of Computing, Montréal, Québec, Canada (May 2002)
14. Salton, G., Buckley, C.: Term weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal* 24(5), 513–523 (1988)

Customer Churn Time Prediction in Mobile Telecommunication Industry Using Ordinal Regression

Rupesh K. Gopal and Saroj K. Meher

Applied Research Group, Satyam Computer Services Limited
Entrepreneurship Center, Indian Institute of Science campus
Bangalore 560 012, India

Ph.: +91-80-23606830; Fax: +91-80-23601011
rupesh.gopal@gmail.com, saroj_meher@satyam.com

Abstract. Customer churn is considered to be a core issue in telecommunication customer relationship management (CRM). Accurate prediction of churn time or customer tenure is important for developing appropriate retention strategies. In this paper, we discuss a method based on ordinal regression to predict churn time or tenure of mobile telecommunication customers. Customer tenure is treated as an ordinal outcome variable and ordinal regression is used for tenure modeling. We compare ordinal regression with the state-of-the-art methods for tenure prediction - survival analysis. We notice from our results that ordinal regression could be an alternative technique for survival analysis for churn time prediction of mobile customers. To the best knowledge of authors, the use of ordinal regression as a potential technique for modeling customer tenure has been attempted for the first time.

1 Introduction

Customer churn is a significant problem in many firms operating on contractual or subscription business setting, like telecommunication operators, Internet service providers, and cable services operators. In a study focusing on the role of satisfaction to model customer length of stay with the telecom service provider, Bolton [5], finds that customer satisfaction is positively correlated with their tenure. First, the tenure is longer for customers who have high levels of cumulative satisfaction, and second, the effect of perceived losses (for e.g., transaction failures, bad service quality) on the tenure is negative. New customers are particularly vulnerable and if their experiences are not satisfactory, the relationship is likely to be short. Customers who are satisfied with the service provider tend to stay for longer durations. Ordinal regression (OR) is a type of learning when the response variable comes from a finite ordered (i.e., *ordinal*) set. Similar to modeling customer satisfaction (which is not observable!) on an ordinal scale, we can model customer tenure as an ordinal response variable to predict customer tenure. In section 2 we explain the use of classical tenure modeling

approach - survival analysis. In section 3 we explain ordinal regression for modeling customer churn times. In section 4 we discuss our empirical results on a real world mobile telecom dataset. In section 5 we conclude our work.

2 Classical Approaches for Tenure Modeling

Traditionally, tenure modeling or length of stay modeling belongs to a branch of statistics called *Survival Analysis*. Survival analysis (SA) is concerned with analyzing the time to occurrence of an event (e.g., time to churn) in the presence of censored observations [2]. In SA, we begin observing a set of customers at some well-defined point of time (called the *origin time*) and then follow them for some substantial period of time, recording the times at which customer churn occurs. Some customers may churn after the end of study period, i.e., after *censoring time*. Such cases are called right censored observations [3]. Several parametric, semi-parametric, and non-parametric survival regression techniques are available as commercial products and is already part of the telecommunication CRM process. Allison [2], gives a good insight into the use of survival analysis for modeling time-to-event data using a commercial statistical package SAS® [4].

In SA, survival function and hazard rate functions are used to describe the status of customer survival during the tenure of observation. The survival time T is considered as a random variable. The survival function $S(t)$ gives the probability of survival to time t , that is, $S(t) = Pr(T > t) = 1 - P(t)$, where $P(t)$ is the c.d.f of survival time T . The hazard function $h(t)$ is defined as the conditional likelihood that a customer will churn at time t , given that churn did not occur in the interval $(0, t)$, and can be computed from $S(t)$ using $h(t) = -\frac{d}{dt}S(t)$. Thus we can compute survival and hazard probabilities for a customer x_i at each time point t_k in the study. By sorting all customers in ascending survival probabilities, at a specified time t_k , the customers with lowest predicted survival probabilities will have highest likelihood to churn at that time [3].

3 Ordinal Regression for Tenure Modeling

In OR, we arrange customer tenure on a ordinal scale such that $t_1 < t_2 < \dots < t_k < \dots < t_M$. Chu & Keerthi [6], formulate the OR problem as a generalization of support vector machines by determining $M - 1$ thresholds (parallel discriminant hyperplanes) for M ranks by dividing the real line into M consecutive intervals, one for each rank. Alternatively, Frank & Hall [7] decompose the original OR problem into a set of binary classification problems. For a review of other OR formulations see [6]. We have implemented the OR formulation proposed in [7] for the results in this paper. The original M -class OR problem with ranks $\{t_1, t_2, \dots, t_M\}$, is converted into $M - 1$ nested binary classification problems by using the ordering of the original ranks. Training starts by deriving new datasets from the original dataset, one for each of the $M - 1$ new binary classes. Each derived dataset contains the same number of samples as the original, with the same attribute values for each sample, except the class value. In the next step,

each of the $M-1$ classifiers will generate a model for each of the new datasets. For each sample (customer) we estimate the probability that it belongs to a target class (i.e., churn probability at that time) as follows: $Pr(t = t_1) = 1 - Pr(t = t_2 \vee t = t_3 \dots \vee t = t_M) = Pr(t > t_1)$, $Pr(t = t_k) = Pr(t > t_{k-1}) - Pr(t > t_k)$ $k = 2, 3, \dots, M-1$, and $Pr(t = t_M) = Pr(t > t_{M-1})$. To predict the churn time of a customer with unknown churn time, the sample is processed by each of the $M-1$ classifiers and the class with maximum probability is assigned to that customer.

In our experiments, we first start by grouping tenures into ranks such that $(t_a, t_b] \prec (t_b, t_c] \prec (t_c, \infty)$, where $t_a < t_b < t_c < \infty$. It is important to note that this grouping into ranks could come from domain experts, like for example, finding a set of customers who are likely to churn in 6-12 months period or finding set of customers who are likely to stay for more than one year or two years. In the next step, we repeat OR experiments on each rank, with a preference level attached between each atomic time unit (that is, at month level). This hierarchical way is taken to overcome the problem of large number of classes present in the current problem.

4 Experiments and Results

We demonstrate results on the *Churn Modeling Tournament* data obtained from The Center for Customer Relationship Management at Duke University [1]. The data were provided by a major wireless telecommunications company using its own customer records. We used calibration dataset which consists of 100,000 customers for whom there were 169 independent variables, a unique identifier for each customer and the churn label (0 for churn and 1 for no-churn).

4.1 Data Preparation

The churn modeling tournament data cannot be directly used for tenure modeling experiments. We need to represent the data in a manner suitable for our experiments. $X = \{x_i : i = 1, 2, \dots, N\}$ denote customers, a_{ij} denotes j^{th} feature value for i^{th} customer. In churn modeling data, number of months in service is one of the feature and churn is the output label. Customers who are active at the time of sampling are treated as censored observations and churners are considered as complete observations. Months in service becomes the output variable. Only right-censoring is considered in the present study. Since all customers in our dataset are at least 7 months old on the network, this becomes the origin time for our experiments. We choose a study period window of 25 months. Therefore the censoring time will be at 31st month. Hence, customers with tenure $7 \leq t < 32$ are considered to be complete observations. Customers whose tenure $t \geq 32$ months are considered to be censored observations (right censoring). We bin tenure into five ranks such that $A \prec B \prec C \prec D \prec E$. Customers who churned in time periods (in months) $[7, 11]$, $[12, 15]$, $[16, 21]$, $[22, 31]$, and $[32, \infty)$ are assigned ranks A, B, C, D, and E respectively. Note that censored observations are assigned to last rank.

4.2 Data Preprocessing

All attributes which had more than 30% of missing values and those that are summation of two or more variables are also removed. Information gain and chi-squared statistic tests [3] are then used for feature selection. Both methods generate ranking for features. We conduct experiments with top 30, 40, 50 and 99 ranked features. Model with 40 top ranked features selected from information gain criterion seems to give the best result and hence only this result is reported. The dataset is randomly split into to halves, one for training and the other testing purpose. Both the data sets have approximately 25,000 samples. We use the open source data mining package Weka version 3.5.5 [8] for our experiments.

4.3 Empirical Results

First, we compare OR with multi-class classification (MC). The output consists of 5 labels {A,B,C,D,E}. For both the classifiers C4.5 decision tree is used as the base learner. OR is compared with one-against-all multi-class classification scheme. In case of OR we have a preference level attached between output labels. This information is absent in MC setting. The classification accuracy obtained from OR and MC is 86.21% and 83.8% respectively. Mean absolute error (MAE) for OR and MC is 0.066 and 0.2512 respectively. MAE is an important parameter for comparison between OR and MC [6]. So far the model is able to predict only a coarse time of churn (for example, between 12 and 15 months) of customers. We repeat our experiments for each rank to predict churn time at month level. MAE values are given inside the parenthesis. Accuracy and MAE results for ranks A, B, C, and D for OR is 67.51% (0.1339), 70.15% (0.162), 56.85% (0.1488) and 39.57% (0.1262) respectively. Accuracy and MAE results for ranks A, B, C, and D for MC is 64.45% (0.4548), 63.93% (0.3047), 54.58% (0.2449) and 39.53% (0.1726) respectively. We notice that OR is consistently more accurate than MC in generating predictions. We note that MAE is consistently lower for OR compared to MC.

Next, we compare OR with Cox proportional hazards (PH) model, [9], for predicting customer tenure at month level. We use the PROC PHREG procedure available in SAS/STAT® software for Cox PH model [4]. Cox PH model is a semi-parametric survival regression technique using partial likelihood estimation. Cox PH model is useful when the form of survival distribution and hazard function are not known in advance. The results are reported using *cumulative lift curve* [1]. Figure 1 shows the cumulative lift curves by OR and PROC PHREG for 7 through 11 months. We notice that PROC PHREG captures 30-60% of churners in the top decile. Whereas, OR captures only about 20-40% of churners in the top decile. Figure 2 shows the cumulative lift curves by OR and PROC PHREG for 12 through 15 months. Here we notice that PROC PHREG is able to capture at most 20% of churners in the top decile. Whereas, OR captures about 20-45% of churners in the top decile. Top decile plots of ordinal regression and PROC PHREG for 16 through 21 months and 22 through 31 months are not presented here because of space limitations. However, the results from PROC PHREG for these time periods are worse relative to ordinal regression. Full set

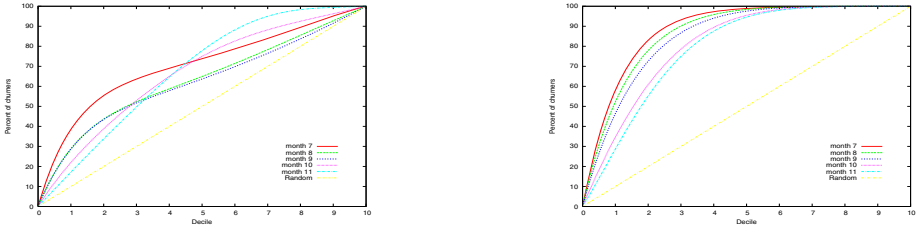


Fig. 1. Cumulative lift curve by OR model (left) and PROC PHREG by SAS/STAT®(right) for customers who churned at month 7 through 11

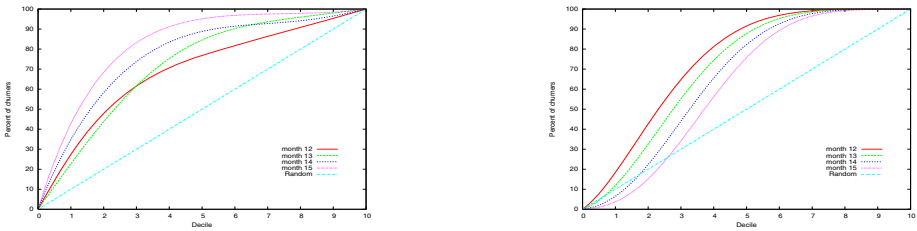


Fig. 2. Cumulative lift curve by OR model (left) and PROC PHREG by SAS/STAT®(right) for customers who churned at month 12 through 15

of results comparing OR and SA is given in our technical report [10]. We notice that PROC PHREG is able to make good predictions about customer churn when their tenure is short (7 to 11 months), where as when tenure prediction is desired for customers who have stayed for a considerably longer time with the service provider, PROC PHREG seems to drastically decrease its accuracy. OR on the other hand is seen to make predictions more uniformly. We anticipate this is due to the balanced datasize of training samples given for training ordinal regression units for each rank.

5 Conclusion and Future Work

In this paper we discussed the use of ordinal regression for modeling tenure of mobile telecommunication customers. Ordinal regression is compared with multi-class classification and is seen to perform better. Next we compared ordinal regression with state-of-the-art method for tenure modeling, survival analysis technique (Cox PH model). Ordinal regression is seen to make more uniform predictions about customer tenure compared to Cox’s model. We would like to emphasize here that ordinal regression is seen to perform better than survival analysis only on the Duke university data. Due to difficulty in getting real world data from telecommunication operators, we are unable to conduct experiments on some more datasets. In future we wish to model tenure of customers at other places where customer satisfaction plays an important role, like insurance and

banking industry. We also wish to compare ordinal regression with parametric survival regression models and different ordinal regression learning schemes on tenure modeling data.

Acknowledgments. We are grateful for the valuable inputs and thoughtful comments by Chiranjib Bhattacharyya, Department of Computer Science and Automation at Indian Institute of Science, Bangalore, India. We thank Gururaj Kallur, Arun Kumar, Sridhar Varadarajan, Srividya Gopalan, Ramya Ramakrishnan and Narasimhan Balakrishnan for their support and encouragement.

References

1. Neslin, S.A., Gupta, S., Kamakura, W., Lu, J., Mason, C.H.: Defection detection: Improving predictive accuracy of customer churn models. Working paper series, Teradata center for customer relationship management at Duke university (2004)
2. Allison, P.D.: Survival analysis using the SAS system: A practical guide, SAS Institute Inc, Cary, NC (1995)
3. Lu, J.: Predicting customer churn in telecommunications industry - An application of survival analysis modeling using SAS. SAS User Group international online proc., Paper No. 114-27 (2002)
4. SAS Institute Inc.: SAS/STAT®Users Guide, Version 6. SAS Institute Inc., 1,2 (1989)
5. Bolton, R.N.: A dynamic model for the duration of the customer's relationship with a continuous service provider: The role of satisfaction. *Marketing Science* 17, 45–65 (1998)
6. Chu, W., Keerthi, S.S.: Support vector ordinal regression. *Neural Computation* 19(3), 145–152 (2007)
7. Frank, E., Hall, M.: A simple approach to ordinal classification. In: Proc. of the European Conf. on Machine Learning, pp. 145–156 (2001)
8. Witten, I.H., Frank, E.: Data mining: Practical machine learning tools and techniques with Java implementations. Morgan Kaufman, San Francisco (2005)
9. Cox, D.R.: Regression models and life tables. *J. of the Royal Stat. Soc., Series B* 34, 187–220 (1972)
10. Gopal, R.K., Bhattacharyya, C., Meher, S.K.: Customer churn time prediction in mobile telecommunication industry using ordinal regression. ARG-TR-Y7-001, Technical report, Satyam Computer Services Limited (2007)

Rule Extraction with Rough-Fuzzy Hybridization Method

Nan-Chen Hsieh

Department of Information Management
National Taipei College of Nursing
No. 365, Min Te Road 11257, Taipei, Taiwan, R.O.C.
nchsieh@ntcn.edu.tw

Abstract. This study presents a rough-fuzzy hybridization method to generate fuzzy if-then rules automatically from a medical diagnosis dataset with quantitative data values, based on fuzzy set and rough set theory. The proposed method consists of four stages: preprocessing inputs with fuzzy linguistic representation; rough set theory in finding notable reducts; candidate fuzzy if-then rules generation by data summarization, and truth evaluation the effectiveness of fuzzy if-then rules. The main contributions of the proposed method are the capability of fuzzy linguistic representation of the fuzzy if-then rules, finding concise fuzzy if-then rules from medical diagnosis dataset, and tolerance of imprecise data.

Keywords: Knowledge discovery in databases, fuzzy if-then rules, soft computing, fuzzy sets, rough sets.

1 Introduction

Medical data often contain imperfect information, while uncertainties, impreciseness and missing values are co-exist. The analysis of medical data thus requires dealing with incomplete and inconsistent information, and manipulates various levels of data representation. However, soft computing techniques are based on quite strong assumptions. They cannot derive conclusions from incomplete knowledge, or manage inconsistent information. The idea of rough set was as a useful mathematical tool to deal with vague concepts and to represent ambiguity, vagueness and uncertainty.

Rough set algorithms [1] do not need membership functions and prior parameter settings. It can extract knowledge from the data itself by means of indiscernibility relations, and generally needs fewer calculations than that of other soft computing techniques. Decision rules extracted by rough set are concise and valuable, which can benefit medical experts by revealing hidden knowledge in the medical dataset. The limitation of traditional rough set theory is concerned with discrete data; quantitative valued had to be discretized for rough set algorithms, which may result in some loss of information. Many researchers proposed the hybridization of fuzzy set and rough set [2-6]. By these approaches, the comparison among objects turned from elements' indistinguishability into their similarity, and the similarity represented by a fuzzy

equivalence relation. This study concentrates on automatically extracting the relevant fuzzy if-then rules in a medical dataset using fuzzy set and rough set theory. As depicted in Fig. 1, a four-stage rough-fuzzy hybridization process for learning fuzzy if-then rules in datasets was proposed.

The rest of this study is organized as follows. Section 2 describes the analytical methodology of this study, and gives an overview of the linguistic summarization of databases and its application in extracting fuzzy if-then rules. Section 3 describes in detail the proposed rough-fuzzy hybridization method in constructing fuzzy rule-base, and shows an example of the ability to extract fuzzy if-then rules in a fuzzy database exhibiting linguistic summaries. The final section draws conclusions.

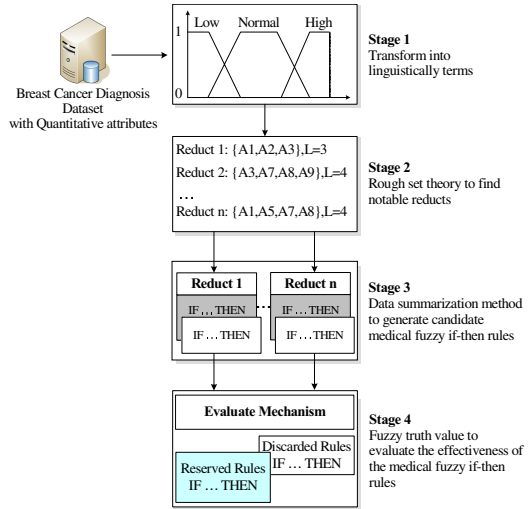


Fig. 1. A four-stage for generating fuzzy if-then rules

2 Assessing Soft Computing Techniques for Generating Fuzzy if-Then Rules

2.1 The Experimental Dataset

The experimental dataset used in this study is a breast cancer diagnosis database obtained from the UCI machine learning repository at <http://www.ics.uci.edu/~mllearn/databases/breast-cancer-wisconsin/>. The Wisconsin diagnostic breast cancer (WDBC) dataset was collected at different periods of time with different characteristic of attributes. The data values of each attribute are quantitative. Several studies are based on this dataset. Setiono [7] proposed a rule extraction technique to generate concise and accurate classification rules in a trained neural network. Tan et al. [8] proposed a two-phase hybrid evolutionary classification technique to extract classification rules to be applied in clinical practice for better understanding and prevention of unwanted medical events. Chou et al. [9] used neural network and MARS techniques to discover the breast cancer pattern.

2.2 Automatically Transform Quantitative Data Values Into Linguistic Terms

Most algorithms for learning rules from examples only accept categorical values, or sharp divide quantitative values into intervals. However, the sharp division of quantitative values either ignores or over-emphasizes the elements near the interval boundary

during data mining. In this study, fuzzy set theory was employed for linguistic representation of quantitative data, thereby producing a fuzzy granulated of the attribute domain. A self-organizing map (SOM) algorithm was used to obtain k midpoints of the granular feature space from each quantitative attribute domain. Next, using fuzzy linguistic representation technique, each attribute domain was characterized as a trapezoidal fuzzy set with individually linguistic terms. The transformed terms are more closely than the linguistic meaning of quantitative data. Moreover, the extracted fuzzy if-then rules can be represented the learned knowledge in terms of human thinking, and tolerated imprecise information more robustly.

The steps for automatically finding fuzzy sets from a given dataset are described herein. Assume that the domain of a quantitative attribute ranges from v_1 to v_2 , and $\{m_1, m_2, \dots, m_k\}$ denote the k midpoints obtained by the SOM algorithm. Using these k midpoints, $k/2+1$ linguistic terms or membership functions can be determined for a trapezoidal fuzzy set. The first membership function is computed as:

$$\begin{aligned}
 f_{first}(x) &= \begin{cases} 1 & \text{if } v_1 \leq x \leq m_1 \\ (m_2 - x)/(m_2 - m_1) & \text{if } m_1 < x < m_2 \\ 0 & \text{if } x \geq m_2 \end{cases} \\
 f_{final}(x) &= \begin{cases} 0 & \text{if } x \leq m_{k-1} \\ (m_k - x)/(m_k - m_{k-1}) & \text{if } m_{k-1} < x < m_k \\ 1 & \text{if } m_k \leq x \leq v_2 \end{cases} \\
 f_p(x) &= \begin{cases} 0 & \text{if } x \leq m_{2p-3} \\ (x - m_{2p-3})/(m_{2p-2} - m_{2p-3}) & \text{if } m_{2p-3} < x < m_{2p-2} \\ 1 & \text{if } m_{2p-2} \leq x \leq m_{2p-1} \\ (m_{2p} - x)/(m_{2p} - m_{2p-1}) & \text{if } m_{2p-1} < x < m_{2p} \\ 0 & \text{if } x \geq m_{2p} \end{cases}, p = 2, \dots, k.
 \end{aligned}$$

The core of rough set theory is finding reducts. A reduct contains a clump of objects in the universal of discourse drawn together by the indistinguishability relation. The reduction step in rough set evolutionary processes keeps only those attributes which preserve the indiscernibility relation. Therefore, minimal subsets of attributes that induce the partitions on the same target attributes with higher support are concerned. However, rough set theory can only handle precise values. Therefore, the trapezoidal fuzzy sets were binalized according to their membership values. For example, the trapezoidal fuzzy set $\{(Low,0.8), (Medium,0.6), (High,0.3)\}$ is binalized as “100”, then the binalized objects are processed by Rough Sets.

2.3 Using Fuzzy Truth Value to Evaluate the Confidence of Fuzzy if-Then Rules

As stated by Yager [10], the linguistic summary is a linguistically quantified proposition containing meta-knowledge about a set of particular objects, and is useful in knowledge discovery. This study aims to consider the notable subsets of tuples in the medical dataset, and to construct linguistic summaries in which attribute values are fuzzy linguistic labels describing each subset of tuples. Thus, for each notable linguistic summary “ Q X

objects in DB are S , the attributes S were determined using rough set theory to the class of objects X , and the validation process is to test the truth of the association between the X and S with respect to the quantifier, Q .

The fuzzy logic based calculus provides the interpreting and validating of the truth statement involving complex linguistic quantifiers, such as “many”, “some” and “few”. Let “ $Q \{t_1, \dots, t_n\}$ are S ” denotes a linguistically quantified statement, and let $\{t_1, \dots, t_n\}$ denotes a set of fuzzy tuples in the medical dataset, DB . The procedure for determining the truth value of a linguistically quantified statement is as follows. If the summary S involves an attribute A , and t_i denotes a tuple that satisfies the summary S , then the membership value of t_i to S is given by:

$$S(t_i) = \max_{\forall k} (\mu_{EQ}(a_k, b_k)), \text{ for all } a_k \in t_i[A], b_k \in S,$$

where $S(t_i)$ denotes the degree to which t_i satisfies the summary S , and the function $\mu_{EQ}(a_k, b_k) = 0$ if and only if $(a_k \neq b_k) \vee (a_k = b_k \wedge \mu(b_k) = 0)$; $\mu_{EQ}(a_k, b_k) = 1 - |\mu(a_k) - \mu(b_k)|$ if and only if $(a_k = b_k \wedge \mu(b_k) \neq 0)$. Then, the individual truth value of “ $\{t_1, \dots, t_n\}$ are S ” for an attribute A over DB is computed as:

$$Truth(\{t_1, \dots, t_n\} \text{ are } S) = \left(\frac{1}{n} \sum_{i=1}^n S(t_i) \right).$$

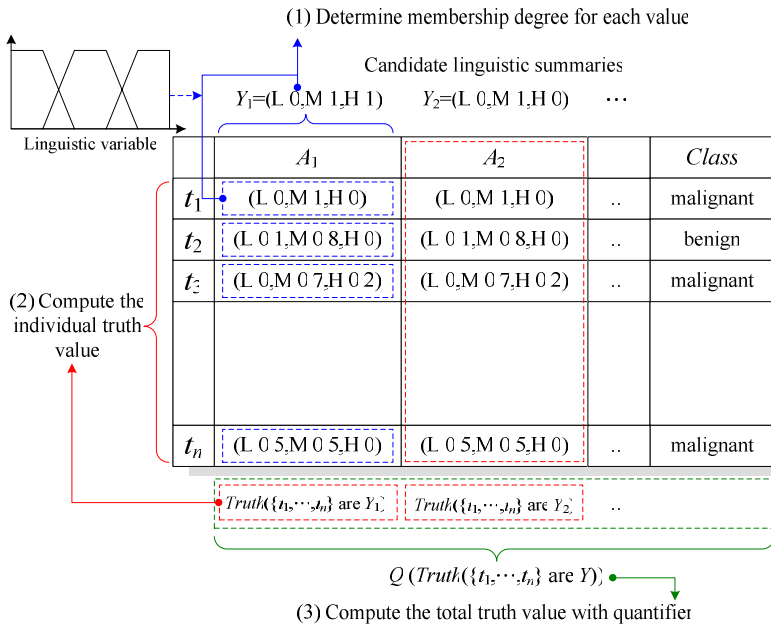


Fig. 2. The procedure for obtaining the truth value of the linguistic summaries

Moreover, when the linguistic summaries are distributed over m attributes with *ANDed* conditions, that is, $S = S_1 \wedge \dots \wedge S_m$, then the total truth value $Truth(\{t_1, \dots, t_n\} \text{ are } S) = \min_{j=1}^m (Truth(\{t_1, \dots, t_n\} \text{ are } S_j))$. When the linguistic summaries are distributed over m attributes with *ORed* conditions, that is, $S = S_1 \vee \dots \vee S_m$, then the total truth value $Truth(\{t_1, \dots, t_n\} \text{ are } S) = \max_{j=1}^m (Truth(\{t_1, \dots, t_n\} \text{ are } S_j))$. Finally, $T = Q(Truth(\{t_1, \dots, t_n\} \text{ are } S))$ denotes the truth value of the linguistically quantified statement “ $Q \{t_1, \dots, t_n\} \text{ are } S$ ” to the fuzzy quantifier Q in agreement. Fig. 2 shows the procedure in obtaining the truth value of the linguistic summaries.

With the judgment standards of support and total truth value, Table 1 shows the fuzzy if-then rules generated by the proposed rough-fuzzy hybridization process. Suitable linguistic quantifier can be employed to interpret linguistic confidence.

Table 1. The fuzzy if-then rules generated by the rough-fuzzy hybridization method

ID	IF	THEN	Support	Accuracy	Total Truth Value
1	Area(Low) AND Concave_points(Low)	benign	0.3304	100%	0.601
2	Perimeter(Low) AND Concave_points(Low)	benign	0.2689	100%	0.852
3	Radius(Low) AND Concave_points(Low)	benign	0.2742	100%	0.928
4	Perimeter(Low) AND Smoothness(Normal) AND Compactness(Normal)	benign	0.0580	100%	0.852
5	Area(Normal) AND Concavity(Normal) AND Symmetry(Low, Normal)	malignant	0.0053	100%	0.805

3 Conclusion

This study proposes a rough-fuzzy hybridization method for learning informative and concise fuzzy if-then rules from examples. The quantitative/categorical interface provided by fuzzy set theory is used for the linguistic representation of examples, and balances the expert perception and system automation. Besides, the reducts provided by rough set theory were found to be a useful tool for finding candidate linguistic summaries. Hence, the discovery of fuzzy if-then rules is similar to the validation of the corresponding linguistic summaries, and the generated fuzzy if-then rules are on the basis of equivalence relation to enhance its readability. Moreover, this study proposed to use fuzzy truth value to evaluate the confidence of fuzzy if-then rules.

Acknowledgments. The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC95-2416-H-227-001.

References

1. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Sciences* 11, 341–356 (1982)
2. Cock, M.D., Cornelis, C., Kerre, E.E.: Fuzzy rough sets: The forgotten step. *IEEE Transactions on Fuzzy Systems* 15, 121–130 (2007)
3. Morsi, N.N., Yakout, M.M.: Axiomatics for fuzzy rough sets. *Fuzzy Sets and Systems* 100, 327–342 (1998)
4. Qin, K., Pei, Z.: On the topological properties of fuzzy rough sets. *Fuzzy Sets and Systems* 151, 601–613 (2005)
5. Radzikowska, A.M., Kerre, E.E.: A comparative study of fuzzy rough sets. *Fuzzy Sets and Systems* 126, 137–155 (2002)
6. Jensen, R., Shen, Q.: Semantics-preserving dimensionality reduction: rough and fuzzy-rough based approaches. *IEEE Transactions on Knowledge and Data Engineering* 16, 1457–1471 (2004)
7. Setiono, R.: Generating concise and accurate classification rules for breast cancer diagnosis. *Artificial Intelligence in Medicine* 18, 205–217 (2000)
8. Tan, K.C., Yu, Q., Heng, C.M., Lee, T.H.: Evolutionary computing for knowledge discovery in medical diagnosis. *Artificial Intelligence in Medicine* 27, 129–154 (2003)
9. Chou, S.-M., Lee, T.-S., Shao, Y.E., Chen, I.-F.: Mining the breast cancer pattern using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications* 27, 133–142 (2004)
10. Yager, R.R.: Database discovery using fuzzy sets. *International Journal of Intelligent Systems* 11, 691–712 (1996)

I/O Scalable Bregman Co-clustering

Kuo-Wei Hsu, Arindam Banerjee, and Jaideep Srivastava

University of Minnesota, Minneapolis, MN, USA
{kuowei, banerjee, srivasta}@cs.umn.edu

Abstract. Consider an $M \times N$ matrix, where the (i, j) th entry represents the affinity between the i th entity of the first type and the j th entity of the second type. Co-clustering is an approach to simultaneously cluster both types of entities, using the affinities as the information guiding the clustering. Co-clustering has been found to achieve clustering and dimensionality reduction at the same time, and therefore it is finding application in various problems. Bregman co-clustering algorithm, which has been recently proposed, converts the co-clustering task to the search for an optimal approximation matrix. It is much more scalable but memory-based implementations have a severe computational bottleneck. In this paper we show that a significant fraction of computations performed by the Bregman co-clustering algorithm naturally map to those performed by an on-line analytical processing (OLAP) engine, making the latter a well suited data management engine for the algorithm. Based on this observation, we have developed a version of Bregman co-clustering algorithm that works on top of OLAP. Our experiments show that this version is much more scalable, achieving an order of magnitude performance improvement over the memory-based implementation. We believe this unlocks the power of this novel technique for application to much larger datasets.

Keywords: Bregman co-clustering, data cube, OLAP, SQL.

1 Introduction

Clustering is an unsupervised learning technique used to group a set of data samples with similar attributes such that the coherence inside a cluster is higher than that between clusters. Co-clustering is an approach which provides two or more simultaneous clusterings of the data samples. It is an emerging research topic relative to standard clustering, which has been widely used for years. Traditionally, clustering algorithms measure the degree of coherence by optimizing various kinds of objective functions defined to minimize the distance between samples. Co-clustering, on the other hand, has attracted great attention because it simultaneously measures the degree of coherence in samples and in attributes. The recently proposed Bregman co-clustering algorithm [2, 3] has shown significant promise, both for the quality of clusterings produced and its computational efficiency. Theoretically, the algorithm is scalable, but practically, the scalability is restricted by the memory space available. For example, when the dataset does not fit entirely in main memory, existing co-clustering algorithms spend a significant fraction of their time in wasteful disk I/O.

In this paper, we address the issue of I/O scalability of co-clustering algorithms. Specifically, we consider the general framework proposed in [3] and propose an approach that utilizes an underlying database to improve its performance. In [5], Chen et al. emphasize the importance of implementing efficient and scalable algorithms that can work on large datasets stored in a database. A key observation in our approach is that co-clustering algorithms require the computation (and often recomputation) of a number of basic summary statistics. This cost can be significantly reduced by using an online analytical processing (OLAP) [8] engine to compute the summary statistics, used to help build the approximation matrix. We contribute to the mapping between the data access operations of the Bregman co-clustering algorithm and those of OLAP. To our knowledge, this topic has not been explored before. Moreover, recent studies [1, 3] have been very successful in applying co-clustering to various applications. Thus, we believe that the techniques presented here will go a long way in making co-clustering applicable to large-scale datasets.

The rest of this paper is organized as follows. Section 2 briefly introduces the Bregman co-clustering algorithm, and Section 3 presents the mapping from the algorithm and OLAP. Section 4 presents experimental results, while Section 5 discusses related works. Finally, conclusions and future work are given in Section 6.

2 Co-clustering and the Bregman Co-clustering Algorithm

Co-clustering, like traditional clustering, uses attributes to group samples. However, unlike traditional clustering, co-clustering clusters rows and columns simultaneously if we use a contingency matrix where a row represents a sample and a column represents an attribute. By grouping similar attributes, co-clustering implicitly achieves dimensionality reduction. This feature reduces the running time and also leads to more informative clusters.

Co-clustering gives accuracy comparable to state-of-the-art approaches in a variety of applications. [3] compares four approaches: SVD [15], NMF [10], a correlation based method, and co-clustering for collaborative filtering. The mean absolute errors (MAEs) on MovieLens dataset¹ for the above four approaches are 0.7721, 0.7636, 0.8214, and 0.7608, respectively. Furthermore, [1] incorporates the statistics about users and movie content, and uses Bregman divergences to find a co-clustering that provides most accurate prediction with the adjustment of covariates.

The Bregman co-clustering algorithm associates a co-clustering task with a matrix approximation task, and the quality of the result is evaluated by the approximation error [3]. Generally, any Bregman divergence can be used for the matrix approximation problem. [3] views a co-clustering task as the search for an optimal approximation matrix, where the approximation is based on the co-clustering so that a better co-clustering leads to a better matrix approximation. The optimality is determined by the minimum Bregman information (MBI) principle [3].

The algorithm builds an approximation matrix according to a user-specified Bregman co-clustering basis, a set of summary statistics. Table 1 summarizes six bases

¹ GroupLens, <http://www.grouplens.org/>

defined in [3]. U is a set of rows and ρ , or $\rho(U)$, maps m rows to k row clusters, while V is a set of column and γ , or $\gamma(V)$, maps n columns to l column clusters. For example, C_2 represents a set of statistics obtained from co-clusters, i.e., $k \times l$ blocks, of row and column clusters, while C_5 corresponds to co-clusters, rows and columns.

Table 1. Definition of six Bregman co-clustering bases [3]

Basis	C_1	C_2	C_3	C_4	C_5	C_6
Def.	$\{\{\rho\},\{\gamma\}\}$	$\{\{\rho,\gamma\}\}$	$\{\{\rho,\gamma\},\{U\}\}$	$\{\{\rho,\gamma\},\{V\}\}$	$\{\{\rho,\gamma\},\{U\},\{V\}\}$	$\{\{U,\gamma\},\{\rho,V\}\}$

In [3], a set of summary statistics is defined as a set of random variables, i.e., $S_A = \{Z' \setminus E[Z|c] = E[Z' \setminus c], \text{ for all } c \text{ in } C\}$, where C is a basis and Z' preserves the summary statistics. The co-clustering problem is described as follows. Given an m -by- n matrix Z , a Bregman divergence d_ϕ , the number of row clusters k , the number of column clusters l , and a co-clustering basis C , the goal is to find an (ρ, γ) that minimizes the expectation of the Bregman divergence between Z and the approximation matrix \hat{Z} which is the Z' giving the minimum Bregman information. It is an optimization problem and the solution is obtained by solving the following equation:

$$\nabla \phi(\hat{Z}) \equiv \nabla \phi(E[Z]) - \sum_{r=1}^s \frac{\Lambda_{c_r}^*}{w_{c_r}}$$

Note that w presents the weight and Λ^* is an optimal Lagrange multiplier [3].

The iterative update approach [3], as in Fig. 1, is a technique to find a locally optimal solution for this NP-hard problem. Initially, for each row (column) the algorithm randomly assigns a row (column) cluster. Next, it calculates the summary statistics according to a user-specified co-clustering basis. When handling row (column) clusters, it treats column (row) clusters as known information and builds an approximation matrix. Subsequently it evaluates and updates row (column) clusters with the best approximation matrix (w. r. t. MBI), i.e., it updates row (column) clusters with the best approximation matrix.

Computing the summary statistics is a key step, and obtaining the Lagrange multipliers for a specified basis is the goal of the computation. The computation requires significant effort, regardless of whether the Bregman divergence corresponds to a closed form solution. Since [3] guarantees a closed form solution for the Lagrange multipliers in terms of the summary statistics for the squared Euclidean distance and I-Divergence, computing the summary statistics is the main bottleneck. The time complexity of the Bregman co-clustering algorithm is relative to the number of non-zero elements in Z [3]. One naive implementation is to adopt a pure memory-based solution, e.g., Matlab. However, such an implementation is under the memory constraint especially when datasets are too large to fit into main memory.


```

Bregman Co-clustering ( $Z, W, C, k, l$ )
Input: Both  $Z$  and  $W$  (weight) are  $m$ -by- $n$  matrices.  $C$  is a co-clustering basis.
         $k$  is the number of row clusters, and  $l$  is number of column clusters.
Output:  $(\rho^*, \gamma^*)$ , optimal row/column clusters
Method:
  Initialize  $\rho, \gamma$  randomly
  Repeat
     $SS \leftarrow \text{ComputeSummaryStatistics}(Z, W, C, \rho, \gamma)$ 
    for  $i = 1$  to  $m$  do
       $\rho(U_i) \leftarrow \arg \min_{\rho' = \rho, \rho'(U_i) = g, 1 \leq g \leq k} E[d_\phi(Z, \hat{Z})]$ , where
       $\hat{Z} \leftarrow \text{BuildApproximationMatrix}(SS, C, \rho', \gamma)$ 
    end for
    for  $j = 1$  to  $n$  do
       $\gamma(V_j) \leftarrow \arg \min_{\gamma' = \gamma, \gamma'(V_j) = h, 1 \leq h \leq l} E[d_\phi(Z, \hat{Z})]$ , where
       $\hat{Z} \leftarrow \text{BuildApproximationMatrix}(SS, C, \rho, \gamma')$ 
    end for
  until convergence
   $(\rho^*, \gamma^*) \leftarrow (\rho, \gamma)$ 
  return  $(\rho, \gamma)$ 

```

Fig. 1. Bregman co-clustering algorithm [3]

3 Use OLAP to Achieve Scalability

For an efficient implementation, we need advanced secondary storage management techniques, which are major functions of relational databases. Furthermore, SQL is superior in providing set-oriented operations, and an OLAP engine prepares the summary statistics inside the database, so we do not need to read all data into memory for the computation.

The first design issue is how to store sparse matrices for OLAP. Three data structures are generally used to store sparse matrices: coordinate storage (COO), compressed sparse row (CSR), and compressed sparse column (CSC). CSR requires much less storage space than COO does but the difference of processing time is not significant [6]. We employ COO to store matrices, because it corresponds to a denormalized table and others are normalized ones while querying one denormalized table is generally faster than querying and joining two normalized ones.

In addition, we use star schema to create the data cube used in our implementation. A data cube consists of all combinations of hierarchical relationships [8], and hence it can be represented in a lattice form, as Fig. 2, where a node will give a smaller data cube or a table that contains aggregates along all possible combinations of specified dimensions. To build the approximation matrix based on C_2 , for example, we need the weighted average of elements in each block, i.e., average of each interaction of column and row clusters, while aggregates along all combinations of row and column clusters can be obtained from Node 9 in Fig. 2. As another example, C_5 requires the summary

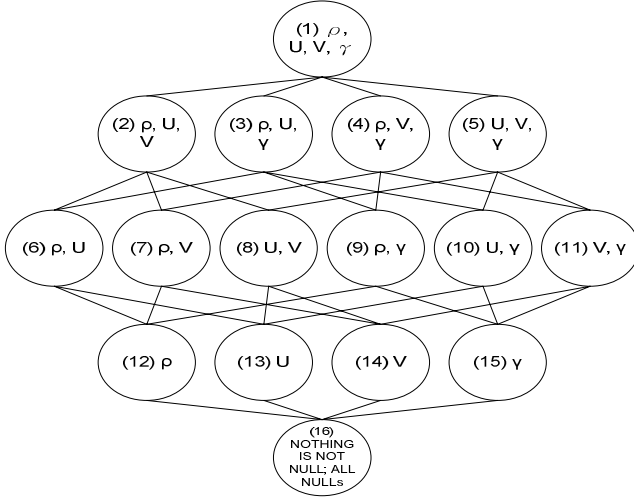


Fig. 2. Mapping Bregman co-clustering bases to the data cube

Table 2. Bregman co-clustering bases and the corresponding nodes in the data cube lattice

Basis	Way to build $\hat{Z}_{i,j}$ for squared Euclidean distance	Nodes
C_1	$E\{[Z_{i,v_j} \in \rho(U_i)]\} + E\{[Z_{v_i,j} \in \gamma(V_j)]\} - E\{[Z_{v_i,v_j}]\}^2$	12,15,16
C_2	$E\{[Z_{i,v_j} \in \rho(U_i) \wedge Z_{v_i,j} \in \gamma(V_j)]\}$ (\wedge presents the logic AND)	9
C_3	$E\{[Z_{i,v_j} \in \rho(U_i) \wedge Z_{v_i,j} \in \gamma(V_j)]\} + E\{[Z_{i,v_j} \in U_i]\} - E\{[Z_{i,v_j} \in \rho(U_i)]\}$	9,12,13
C_4	$E\{[Z_{i,v_j} \in \rho(U_i) \wedge Z_{v_i,j} \in \gamma(V_j)]\} + E\{[Z_{v_i,j} \in V_j]\} - E\{[Z_{v_i,j} \in \gamma(V_j)]\}$	9,14,15
C_5	$E\{[Z_{i,v_j} \in \rho(U_i) \wedge Z_{v_i,j} \in \gamma(V_j)]\} + E\{[Z_{i,v_j} \in U_i]\} + E\{[Z_{v_i,j} \in V_j]\} - E\{[Z_{i,v_j} \in \rho(U_i)]\} - E\{[Z_{v_i,j} \in \gamma(V_j)]\}$	9,12,13, 14,15
C_6	$E\{[Z_{i,v_j} \in U_i \wedge Z_{v_i,j} \in \gamma(V_j)]\} + E\{[Z_{v_i,j} \in V_j \wedge Z_{i,v_j} \in \rho(U_i)]\} - E\{[Z_{i,v_j} \in \rho(U_i) \wedge Z_{v_i,j} \in \gamma(V_j)]\}$	7,9,10

statistics from Node 9, 12, 13, 14, 15. Table 2 summarizes the mappings between Bregman co-clustering bases and the data cube lattice.

4 Performance Evaluation

Our implementation is built in C# with Microsoft SQL Server 2005 as the backend database. We evaluate it by comparing it against the Matlab implementation for all Bregman co-clustering bases. Fig. 3 illustrates experimental results.

² For I-Divergence, we replace + and - with * (multiplication) and / (division), respectively. The MBI problem has a closed form solution in both cases even though for a general Bregman divergence the solution is not necessarily in a closed form [3].

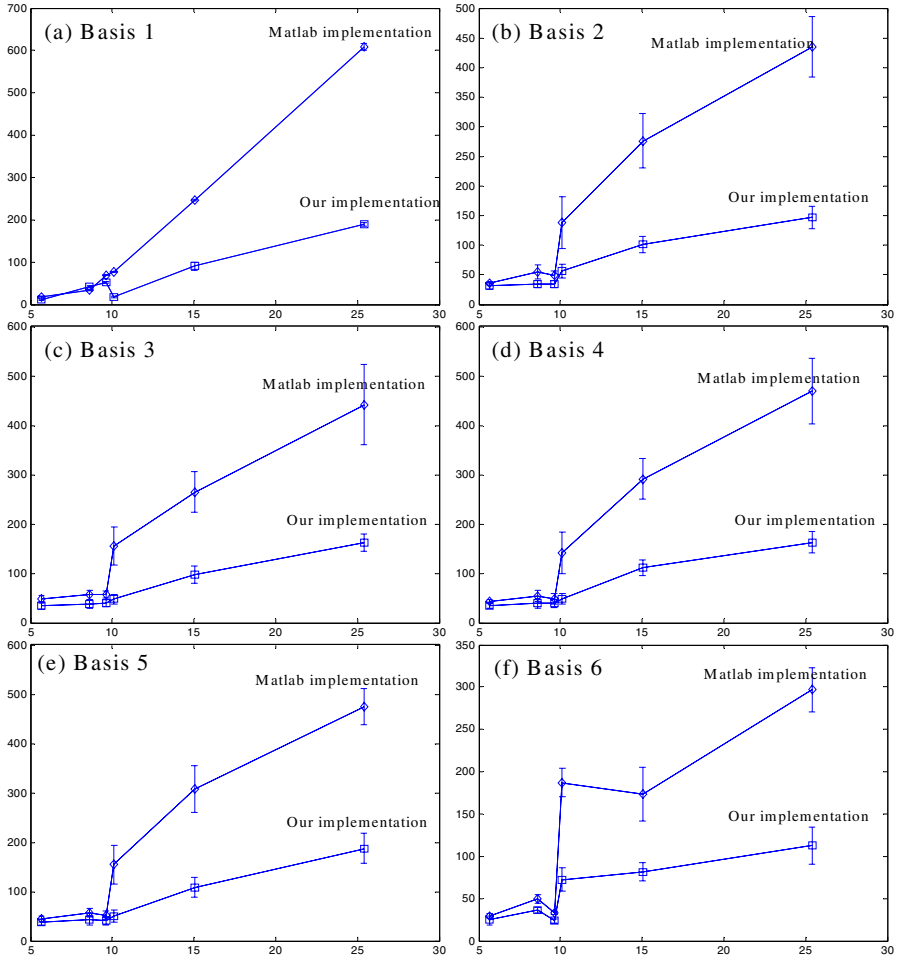


Fig. 3. Experimental results. The horizontal axis shows the number of non-zero elements ($\times 10^3$) in a data matrix. All datasets are different in size. The vertical axis presents the average running time in seconds, including the I/O. A vertical bar indicates the standard deviation.

Matrices used here (i.e., `gre_1107`, `nnc1374`, `pores_2`, `dw2048`, `zenios`, `lnsp3937`, and `e20r5000`, ordered by size) are available on <http://math.nist.gov/MatrixMarket/>. Both k and l are set to 10, and we allow at most 20 iterations in a run. For each dataset (except the largest one) and for each basis, we run each implementation (with the squared Euclidean distance function) 10 times. Fig. 3 presents that both implementations produce similar shapes of curves but ours achieves a slower growth of the curve. As the number of non-zero elements increases, their differences also increase. For the memory consumption, the Matlab implementation requires more than 800M bytes of memory for the second largest dataset and it can not even process the largest one, but

ours processes the largest dataset without difficulties. Since this is an I/O-bound problem, differences from compilers and languages are not the main issue.

5 Related Work

One example for integrating data mining algorithms into OLAP is DBMiner [9] Using OLAP to quickly collect properties of clusters for advanced analyses, DBMiner [9] supports classification, association rules mining, and cluster analysis; in contrast, we employ data cubes to implement the Bregman co-clustering algorithm. [7] extends SQL to compute the summary statistics for classification, whereas we follow standard operations and enjoy higher portability and flexibility. [11] discusses the vertical, horizontal, and hybrid table schemas for implementing EM algorithm in SQL. The vertical table schema is a flexible design even though it comes with the highest overhead. Our schema acts as the vertical one because mapping the vertical table schema to COO is straightforward. Nevertheless, our schema does not suffer from the overhead since there are less join operations used to create a cube. Furthermore, the approach proposed in [11] is not as flexible as ours, because the number of dimensions it can handle is restricted to the length of a SELECT statement. [12] argues that integrating clustering algorithms into a database is practical, while [13] proposes a pure SQL-based approach to perform the K-means clustering over large datasets. However, it is not the nature of SQL to compute the loss function for each sample for each cluster. Thus, we compute the Bregman divergences in memory. For implementations of the K-mean algorithm, [14] compares SQL to C++ and concludes that the SQL implementation presents a slower growth. We present a similar phenomenon. [16] proposes a system, WekaDB, to help Weka³ handle large datasets. WekaDB is slower than Weka but it can handle much larger datasets. In some cases, WekaDB is faster than a pure SQL-based approach. We agree with the conclusion: Using a database as a backend and main memory as a buffer would provide higher scalability.

6 Conclusions and Future Work

In this paper, we proposed a novel and efficient implementation for the Bregman co-clustering algorithm, which has been proven to generate substantially better co-clustering results than other algorithms. Our implementation utilizes an OLAP engine to obtain the summary statistics used for the construction of approximation matrices, and then it reads data from a database to memory for the computation of Bregman divergences. This paper contributes to the mapping between Bregman co-clustering bases to the data cube, and we also demonstrate that a database can act as an effective computation engine for data mining. Experimental results show that our implementation provides higher scalability by using OLAP. Future work includes the study of index structures and the extension to multi-dimensional co-clustering [4].

³ <http://www.cs.waikato.ac.nz/ml/weka/>

References

- [1] Agarwal, D., Merugu, S.: Predictive discrete latent factor models for large scale dyadic data. In: 13th ACM SIGKDD, pp. 26–35 (2007)
- [2] Banerjee, A., Dhillon, I.S., Ghosh, J., Merugu, S., Modha, D.S.: A generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. In: 10th ACM SIGKDD, pp. 509–514 (2004)
- [3] Banerjee, A., Dhillon, I.S., Ghosh, J., Merugu, S., Modha, D.S.: A generalized Maximum Entropy Approach to Bregman Co-clustering and Matrix Approximation. *J. Machine Learning Research* 8, 1919–1986 (2007)
- [4] Banerjee, A., Basu, S., Merugu, S.: Multi-way Clustering on Relation Graphs. In: 7th SIAM Int'l. Conf. on Data Mining (SDM) (2007)
- [5] Chen, M.-S., Han, J., Yu, P.S.: Data Mining: An Overview from Database Perspective. *IEEE Trans. on Knowledge and Data Engineering* 8(6), 866–883 (1996)
- [6] Goharian, N., Jain, A., Sun, Q.: Comparative Analysis of Sparse Matrix Algorithms for Information Retrieval. *J. Systemics, Cybernetics and Informatics* (2003)
- [7] Graefe, G., Fayyad, U., Chaudhari, S.: On the Efficient Gathering of Sufficient Statistics for Classification from Large SQL Databases. In: 4th ACM SIGKDD, pp. 204–208 (1998)
- [8] Gray, J., Chaudhuri, S., Bosworth, A., Layman, A., Reichart, D., Venkatrao, M., Pellow, F., Pirahesh, H.: Data Cube: A relational Aggregation Operator Generalizing Group-By, Cross-Tab, and Sub-Totals. *Data Mining and Knowledge Discovery* 1(1), 29–53 (1997)
- [9] Han, J.: OLAP Mining: An Integration of OLAP with Data Mining. In: IFIP Conf. Data Semantics (DS-7), pp. 1–11 (1997)
- [10] Hofmann, T.: Latent semantic models for collaborative filtering. *ACM Trans. on Information Systems* 22(1), 89–115 (2005)
- [11] Ordonez, C., Cereghini, P.: SQLEM: Fast Clustering in SQL using the EM Algorithm. In: SIGMOD Conference, pp. 559–570 (2000)
- [12] Ordonez, C., Omiecinski, E.: Efficient Disk-Based K-Means Clustering for Relational Databases. *IEEE Trans. on Knowledge and Data Engineering* 16(8), 909–921 (2004)
- [13] Ordonez, C.: Programming the K-means Clustering Algorithm in SQL. In: 10th ACM SIGKDD, pp. 823–828 (2004)
- [14] Ordonez, C.: Integrating K-Means Clustering with a Relational DBMS Using SQL. *IEEE Trans. on Knowledge and Data Engineering* 18(2), 188–201 (2006)
- [15] Sarwar, B., Karypis, G., Konstan, J., Ridel, J.: Application of dimensionality reduction in recommender systems – a case study. In: WebKDD Workshop (2000)
- [16] Zou, B., Ma, X., Kemme, B., Newton, G., Precup, D.: Data mining using relational database management systems. In: 10th PAKDD, pp. 657–667 (2006)

Jumping Emerging Patterns with Occurrence Count in Image Classification*

Lukasz Kobyliński and Krzysztof Walczak

Institute of Computer Science, Warsaw University of Technology
ul. Nowowiejska 15/19, 00-665 Warszawa, Poland
{L.Kobyliński, K.Walczak}@ii.pw.edu.pl

Abstract. In this paper we propose an application of jumping emerging patterns (JEPs) to the classification of images. We define a new type of patterns, namely the jumping emerging patterns with occurrence count (occJEPs), which allow reasoning in transaction databases with recurrent items. Such data is a frequently used representation of images, for which classification is one of the most important data mining problems that needs to be solved accurately and efficiently. We provide a formal definition of the new type of patterns, an outline of an algorithm for finding occJEPs and a comparison with other rule- and pattern-based classifiers for a selection of sample images.

1 Introduction

In this article we address the problem of discovering JEPs and using them for supervised learning in image databases, where the images are described by multi-sets of features. This is an enhancement of the transactional database representation, where instead of a binary relation between items and database records, an occurrence count is associated with every item in a set. We propose a new type of JEPs to accomplish this task, the jumping emerging patterns with occurrence count (occJEPs), show an outline of an algorithm for finding occJEPs and compare their discriminative value with other recent classification methods.

2 Jumping Emerging Patterns in Transaction Databases

The concept of discovering jumping emerging patterns efficiently and using them in the classification of transactional datasets has been introduced in [1]. Such patterns have proved to be a very accurate alternative to previously proposed rule- and tree-based classifiers.

Formal Definition. We restrict further discussion on emerging patterns to transaction systems [2]. A transaction system is a pair $(\mathcal{D}, \mathcal{I})$, where \mathcal{D} is a finite sequence of transactions (T_1, \dots, T_n) (database), such that $T_i \subseteq \mathcal{I}$ for $i = 1, \dots, n$

* The research has been partially supported by grant No 3 T11C 002 29 received from Polish Ministry of Education and Science.

and \mathcal{I} is a non-empty set of items (itemspace). A support of an itemset $X \subset \mathcal{I}$ in a sequence $D = (T_i)_{i \in K \subseteq \{1, \dots, n\}} \subseteq \mathcal{D}$ is defined as $\text{supp}_D(X) = \frac{|\{i \in K : X \subseteq T_i\}|}{|K|}$.

Given two databases $D_1, D_2 \subseteq \mathcal{D}$ we define an itemset $X \subset \mathcal{I}$ to be a jumping emerging pattern (JEP) from D_1 to D_2 if $\text{supp}_{D_1}(X) = 0 \wedge \text{supp}_{D_2}(X) > 0$. A set of all JEPs from D_1 to D_2 is called a JEP space and denoted by $JEP(D_1, D_2)$. A minimal JEP is a jumping emerging pattern X , such that no proper subset of X is a JEP.

3 Jumping Emerging Patterns with Occurrence Count

We propose an extension of the definition of a transaction system that is necessary for the introduction of emerging patterns with occurrence count. The primary motivation for introducing this new type of JEPs is to allow a wider range of problems being directly approachable by means of pattern analysis. Although there are many possibilities of transforming data to a transactional form, it may not be a feasible solution due to unacceptable enlargement of attribute space, information loss or limited possibility of transformed data interpretation. For example, a relational dataset containing numbers of similarly colored objects visible on an image, may be directly transformed to a transactional form by populating the itemspace with all the possible attribute values or by discretization and creating items for each of the resulting discrete ranges. The first solution is in most cases impractical because of the itemspace size, while the second always introduces some loss of information. In both cases the possibility of relating to data semantics is limited in the transactional form of data, as for example discovering patterns between image classes that differ in the number of colored objects by a certain value.

Formal Definition. Let a transaction system with recurrent items be a pair $(\mathcal{D}^r, \mathcal{I})$, where \mathcal{D}^r is a database and \mathcal{I} is an itemspace (the definition of itemspace remains unchanged). We define database \mathcal{D}^r as a finite sequence of transactions (T_1^r, \dots, T_n^r) for $i = 1, \dots, n$. Each transaction is a set of pairs $T_i^r = \{(t_i, p_i); t_i \in \mathcal{I}\}$, where $p_i : \mathcal{I} \rightarrow \mathbb{N}$ is a function, which assigns the number of occurrences to each item of the transaction. Similarly, a multiset of items X^r is defined as a set of pairs $\{(x, q); x \in \mathcal{I}\}$, where $q : \mathcal{I} \rightarrow \mathbb{N}$. We say that $x \in X^r \iff q(x) \geq 1$ and define $X = \{x : x \in X^r\}$. We will write $X^r = (X, Q)$ to distinguish X as the set of items contained in a multiset X^r and Q as the set of functions, which assign occurrence counts to particular items.

The support of a multiset of items X^r in a sequence $D^r = (T_i^r)_{i \in K \subseteq \{1, \dots, n\}} \subseteq \mathcal{D}^r$ is defined as $\text{supp}_{D^r}^r(X^r, \theta) = \sum_{i \in K} \frac{\phi(X^r, T_i^r, \theta)}{|K|}$, where ϕ is a function of three arguments: a multiset $X^r = (X, Q)$, a transaction $T^r = (T, P)$ and an occurrence threshold $\theta \geq 1$: $\phi(X^r, T^r, \theta) = 1 \iff \forall_{x \in \mathcal{I}} p(x) \geq \theta \cdot q(x)$ and $\phi(X^r, T^r, \theta) = 0$ otherwise. The occurrence threshold allows for differentiating transactions containing the same sets of items with a specified tolerance margin of occurrence counts.

Let a decision transaction system [2] be a tuple $(\mathcal{D}^r, \mathcal{I}, \mathcal{I}_d)$, where $(\mathcal{D}^r, \mathcal{I} \cup \mathcal{I}_d)$ is a transaction system with recurrent items and $\forall T^r \in \mathcal{D}^r |T \cap \mathcal{I}_d| = 1$. Elements of \mathcal{I} and \mathcal{I}_d are called condition and decision items, respectively. A support for a decision transaction system $(\mathcal{D}^r, \mathcal{I}, \mathcal{I}_d)$ is understood as a support in the transaction system $(\mathcal{D}^r, \mathcal{I} \cup \mathcal{I}_d)$.

For each decision item $c \in \mathcal{I}_d$ we define a decision class sequence $C_c = (T_i^r)_{i \in K}$, where $K = \{k \in \{1, \dots, n\} : c \in T_k\}$. Notice that each of the transactions from \mathcal{D}^r belongs to exactly one class sequence. In addition, for a database $D = (T_i^r)_{i \in K \subseteq \{1, \dots, n\}} \subseteq \mathcal{D}^r$, we define a complement database $D' = (T_i^r)_{i \in \{1, \dots, n\} - K}$.

Given two databases $D_1, D_2 \subseteq \mathcal{D}^r$ we call a multiset of items X^r a jumping emerging pattern with recurrent items (occJEP) from D_1 to D_2 , if $\text{supp}_{D_1}(X^r, 1) = 0 \wedge \text{supp}_{D_2}(X^r, \theta) > 0$, where θ is the occurrence threshold. A set of all occJEPs with a threshold θ from D_1 to D_2 is called an occJEP space and denoted by $\text{occJEP}(D_1, D_2, \theta)$. We distinguish the set of all minimal occJEPs as $\text{occJEP}_m, \text{occJEP}_m(D_1, D_2, \theta) \subseteq \text{occJEP}(D_1, D_2, \theta)$. Notice also that $\text{occJEP}(D_1, D_2, \theta) \subseteq \text{occJEP}(D_1, D_2, \theta - 1)$ for $\theta \geq 2$. In the rest of the document we will refer to multisets of items as itemsets and use the symbol X^r to avoid confusion.

Finding occJEPs. We will now give an outline of the algorithm to discover a set of occJEPs in a given decision transaction system with recurrent items. Let C_c be a decision class sequence of a database \mathcal{D}^r for a given decision item c and C'_c a complement sequence to C_c . We define $D_1 = C'_c, D_2 = C_c$ and the aim of the algorithm to discover $\text{occJEP}_m(D_1, D_2, \theta)$.

At first, notice that only the patterns, which are not supported in D_1 are possible candidates for occJEPs. In case of single-item patterns $X^r = (X, Q)$, where $X = \{x\}, x \in \mathcal{I}$ it is the case, when $\forall T^r = (T, P) \in D_1 q(x) > p(x)$. In case of multi-item patterns at least one of the item counts of the candidate pattern has to be larger than the corresponding item count in the database. We can write this as: $X^r = (X, Q)$ is an occJEP candidate $\iff \forall T^r = (T, P) \in D_1 \exists x \in X q(x) > p(x)$.

The first step of the algorithm is then to create a set of conditions in the form of $[q(i_j) > p_1(i_j) \vee \dots \vee q(i_k) > p_1(i_k)] \wedge \dots \wedge [q(i_j) > p_n(i_j) \vee \dots \vee q(i_k) > p_n(i_k)]$ for each of the candidate itemsets $X^r = (X, Q), X \subseteq 2^{\mathcal{I}}$, where j and k are subscripts of items appearing in a particular X^r and n is the number of transactions in D_1 . Solving this set of inequalities results in its transformation to the form of $[q(i_j) > r_j \wedge \dots \wedge q(i_k) > r_k] \vee \dots \vee [q(i_j) > s_j \wedge \dots \wedge q(i_k) > s_k]$, where r and s are the occurrence counts of respective items. The counts have to be incremented by 1, to fulfill the condition of $\text{supp}_{D_1}^r(X^r, \theta) = 0$.

As both the itemspace and database sizes may be large enough to make the solution of the above set of inequalities difficult in practical implementations, we propose the following two countermeasures. Firstly, as the purpose of discovering the patterns is to use them in a classifier, we limit the search to only minimal occJEPs. This way we can eliminate all multi-item conditions, for which a condition with a lesser number of items and the same or lower number of item occurrences exists. The second possibility of ensuring a limited-time solution to this problem is introducing a maximum pattern size parameter. We reduce the

problem of finding all possible occJEP candidates, such that $X \subseteq 2^{\mathcal{I}}$, to itemsets not larger than a specified size δ .

Having found the minimum occurrence counts of items in the candidate itemsets, we then calculate the support of each of the itemsets in D_2 with a threshold θ . The candidates, for which $\text{supp}_{D_2}^r(X, \theta) > 0$ are the minimal *occJEPs*(D_1, D_2, θ).

For the example given by Table 1 we can see that the support of candidate patterns $\text{supp}_{D_2}^r(X_1^r, 3) = 0$, $\text{supp}_{D_2}^r(X_2^r, 2) > 0$ and $\text{supp}_{D_2}^r(X_3^r, 1) > 0$. X_2^r and X_3^r are thus minimal occJEPs with threshold values $\theta = 2$ and $\theta = 1$ respectively. By the definition of occJEPs, X_2^r is also an occJEP for $\theta \in [1, 2]$.

Table 1. Finding occJEPs in a transaction database with recurrent items. Calculating the support count of candidate itemsets in complementary database. $X_1^r = \{3 \cdot i_1\}$, $X_2^r = \{4 \cdot i_1, 3 \cdot i_3\}$, $X_3^r = \{2 \cdot i_1, 2 \cdot i_2, 2 \cdot i_3\}$.

D_2	$p(i_1)$	$p(i_2)$	$p(i_3)$	$\phi(X_1^r, T^r, 3)$	$\phi(X_2^r, T^r, 2)$	$\phi(X_3^r, T^r, 1)$
T_1^r	4	9	2	0	0	1
T_2^r	3	5	7	0	0	1
T_3^r	8	1	6	0	1	0
				$\text{supp}_{D_2}^r = 0$	$\text{supp}_{D_2}^r = 1/3$	$\text{supp}_{D_2}^r = 2/3$

The outline of the complete algorithm is then as follows: (a) formulate minimum occurrence conditions for all δ -sized subsets of items of D_1 ; (b) solve the inequalities, while eliminating non-minimal candidates; (c) count the supports of the resulting candidate itemsets in D_2 , using threshold θ and having occurrence counts incremented by one; (d) return itemsets with support greater than zero as minimal occJEPs.

4 Tile-Based Image Representation

We use a symbolic, color and texture-based representation of images divided into tiles, to capture enough information to be able to reason about their underlying content. The symbolic representation also allows for direct discovery of the proposed occJEPs in databases of images and performing a pattern-based classification.

The images are uniformly divided into a grid of $x \times y$ pixel tiles, where x is the number of rows and y is the number of columns, and for each of the tiles the color and texture features are calculated. This initial procedure may be performed during the database indexing phase and is not tied to the actual classification process. The next step is creating a dictionary of typical feature values. This is performed by clustering all available feature vectors of the tiles of the learning database to find a chosen number of group centroids, which then become elements of the dictionary. The tiles of images are then labeled with identifiers of the most similar entries present in the dictionary. The representation of a particular image consists of a list of all identifiers associated with its tiles, with or without the occurrence counts.

5 Experimental Results

To assess the performance of the proposed jumping emerging patterns with occurrence count in classification we have chosen a test dataset of images made available by the authors of the SIMPLIcity CBIR system [4]. Presented here are the results of classification between the following classes: *flower*, *food*, *elephant* and *mountain*. The accuracy of both the approach described above and the method of classification with class association rules proposed earlier in [3] has been compared with C4.5 classifier and a classifier based on regular JEPs. We have used ten-fold cross-validation to reduce any influence imposed by the partition of the available dataset into training and test images.

The occJEP-based classifier has been created by discovering all minimal occJEPs to each of the classes present in the test data. We can formally define the set of patterns in a classifier $occJEP_C^\theta$ for a given occurrence threshold θ as: $occJEP_C^\theta = \bigcup_{c \in \mathcal{I}_d} occJEP_m(C'_c, C_c, \theta)$, where $C_c \subseteq \mathcal{D}_L^r$ is a decision class sequence for decision item c and C'_c is a complementary sequence in a learning database \mathcal{D}_L^r . Classification of a particular transaction in the testing database \mathcal{D}_T^r is performed by aggregating all minimal occJEPs, which are supported by it, similarly as in [1].

Classification results of each of the dataset pairs are presented in Table 2. We have performed experiments for threshold $\theta \in [1, 3]$ and a selection of three values for which the accuracy was the highest is shown here. In most cases $\theta = 1.5$ gave the best results, in one case there was no improvement of accuracy for $\theta > 1$. The maximum length of discovered occJEPs was set at $\delta = 3$. The images were partitioned into 8×8 tiles and the dictionary size was set at 8 values.

Table 2. Classification accuracy of the four test datasets

θ	accuracy (%)					
	<i>flower/</i> <i>food</i>	<i>flower/</i> <i>elephant</i>	<i>flower/</i> <i>mountain</i>	<i>food/</i> <i>elephant</i>	<i>food/</i> <i>mountain</i>	<i>elephant/</i> <i>mountain</i>
1	96.84	98.47	95.84	91.50	96.00	92.50
1.5	98.47	99.47	96.37	93.50	95.00	93.50
2	96.37	98.95	96.89	90.50	95.00	94.00

The comparison of classification performance of the C4.5 algorithm, class association rules, an occJEP- and JEP-based classifiers is presented in Table 3. For occJEPs, we have set the threshold θ to a value which had previously proved to give the best results (compare with Table 2), while $\delta = 3$. Apart from classification accuracy, we also note the average number of found occJEPs, JEPs, CARs and the size of the generated tree, in case of the C4.5 classifier. We can see that in all three test cases the classification accuracy of the proposed method was the highest, as is the number of found patterns. A possible explanation of the poor CARs performance is its too strict rule selection method, as only a few are

used during the actual classification. The possibility to adjust the occurrence threshold value is a distinguishing feature of the occJEP classifier and surely accounts for its accuracy.

Table 3. Comparison of classifiers

datasets	accuracy (%)				patterns/rules/tree size			
	occJEP	JEP	CAR	C4.5	occJEP	JEP	CAR	C4.5
<i>flower/food</i>	98.47	89.67	91.10	96.32	1443.8	8.5	4.2	6.6
<i>flower/elephant</i>	99.47	92.16	93.26	97.42	391.7	20.2	8.8	8.8
<i>flower/mountain</i>	96.89	89.83	91.15	92.21	139.5	28.2	9.3	10.0

6 Conclusions

We have presented a concept of extending the definition of a jumping emerging pattern to include occurrence counts of its items and use such patterns for classification. Jumping emerging patterns with occurrence count (occJEPs) carry the same highly discriminative information as regular JEPs, but also allow reasoning in databases with transactions of recurrent items. We have proposed a generalization of a decision transaction system, where each transaction is a multiset of items, and a new method of counting supports in such databases.

We have experimentally shown that occJEP-based classifiers may outperform the accuracy of other current classification methods, appropriate for relational or recurrent transactional data. The presented method has an advantage over classifying discretized data with currently used algorithms of a reduced itemspace and allowing the user to adapt the classification process to differences between classes by changing the threshold parameter.

References

1. Li, J., Dong, G., Ramamohanarao, K.: Making use of the most expressive jumping emerging patterns for classification. *Knowl. Inf. Syst.* 3, 1–29 (2001)
2. Terlecki, P., Walczak, K.: Jumping Emerging Pattern Induction by Means of Graph Coloring and Local Reducts in Transaction Databases. In: An, A., Stefanowski, J., Ramanna, S., Butz, C.J., Pedrycz, W., Wang, G. (eds.) *RSFDGrC 2007*. LNCS (LNAI), vol. 4482, Springer, Heidelberg (2007)
3. Kobyliński, L., Walczak, K.: Class Association Rules with Occurrence Count in Image Classification. *TASK Quarterly* 11, 35–45 (2007)
4. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLiCity: Semantics-sensitive integrated matching for picture libraries. *IEEE Trans. on Patt. Anal. and Machine Intell.* 23, 947–963 (2001)

Mining Non-coincidental Rules without a User Defined Support Threshold

Yun Sing Koh

School of Computing and Mathematical Sciences,
Auckland University of Technology, New Zealand
ykoh@aut.ac.nz

Abstract. Traditional association rule mining techniques employ the support and confidence framework. However, specifying minimum support of the mined rules in advance often leads to either too many or too few rules, which negatively impacts the performance of the overall system. Here we propose replacing Apriori's user-defined minimum support threshold with the more meaningful MinAbsSup function. This calculates a custom minimum support for each itemset based on the probability of chance collision of its items, as derived from the inverse of Fisher's exact test. We will introduce the notion of coincidental itemsets; given a transaction dataset there is a chance that two independent items are appearing together by random coincidence. Rules generated from these itemsets do not denote a meaningful association, and are not useful.

1 Introduction

The two major problems with traditional association rule mining are the high cost of generating association rules and the large number of excess rules that are generated. Traditional association rule mining algorithms, such as Apriori [1], use the support confidence framework, which requires a user defined support threshold. In light of this, there has been much research into developing techniques [2,3,4] to find a meaningful support threshold. Setting a hard support threshold is no longer sufficient. If the support threshold is set too high, we produce rules which are of common knowledge and we may prune rare itemsets which have a high confidence and have valuable information. However setting the minimum support threshold too low would produce many trivial rules. It also does not guarantee a strong association. Strongly associated rules have more predictive power, and are more useful.

We propose to use the minimum absolute support (MinAbsSup) function [5] that generates a minimum absolute support value for each candidate itemset. This replaces the original minimum support threshold in Apriori. When the support of an itemset is below its minimum absolute support value we assume that it is occurring due to random coincidence and it is pruned. This method proposed is statistically more meaningful compared to other methods used to arbitrarily choose a minimum support threshold. MinAbsSup function was proposed to eliminate the occurrences of itemsets that occur due to some random process. These itemsets are occurring together by coincidence are not strongly associated or statistically significant. We call this the *coincidental itemset*

problem. These are important because itemsets that have high support and high confidence may be appearing so frequently in a transaction database that they cannot help but appear together, while itemsets that have a low support but high confidence may be occurring due to chance and could be considered as “noise.” In these rare cases a high confidence value may not be sufficient to determine a valid rule. When dealing with rare cases we are dealing with low support, thus even a small variation in the support of an itemset would dramatically effect its confidence value.

2 Preliminaries and Related Work

The following is a formal statement of association rule mining for transaction databases. Let $I = \{i_1, \dots, i_m\}$ be the universe of items. A set $X \subseteq I$ of items is called an itemset. A transaction $t = (tid, X)$ is a tuple where tid is a unique transaction ID and X is an itemset. A transaction database D is a set of transactions. The *support* of an itemset X in D , denoted by $\text{supp}(X)$, is the proportion of transactions in D that contain X . The rule $X \rightarrow Y$ holds in the transaction set D with *confidence* where $\text{conf}(X \rightarrow Y)$ is the proportion of transaction that contain X also containing Y . There has been a lot of research into developing efficient algorithms for mining itemsets with a variable minimum support threshold [2,3,6,4]. These algorithms are exhaustive in their generation of rules, and so spend time looking for rules with high support and high confidence. If the varied minimum support value is set close to zero, they will take a similar amount of time to that taken by Apriori to generate low-support rules in amongst the high-support rules. These methods generate all rules that have high confidence and high support. To include rare items, the minsup threshold must be lower, which consequently generates an enormous set of rules consisting of both frequent and infrequent items. A uniform minimum support threshold is not effective for datasets with a skewed distribution because they tend to generate many trivial patterns or miss potential low-support patterns. Hence another approach is to use association rule mining without support threshold, but it usually introduces another constraint such as similarity or confidence pruning. However none of these researches have directly considered the coincidental itemset problem. There has been some research that is relevant to the coincidental itemset problem. In order to improve the support-confidence framework, some have proposed using an additional measure [7,8,9,10]. Brin et al. (1997) proposed a pruning method based on the chi-square model. They use the chi-square test to prune out the insignificant rules by using it to test whether the antecedent and the consequent of the rule are statistically associated. A rule is significant if and only if it is statistically associated. However the chi-square test is only an approximation of the true level of association and does not work well with rare itemsets, which are genuinely associated and have low support and high confidence.

3 Coincidental Itemset Problem

There are two types of candidate itemsets: non-coincidental itemsets and coincidental itemsets. Non-coincidental itemsets are generated by some non-random process. Whereas, there are two different circumstances in which a coincidental itemset may

occur. Items within an itemset may be appearing so frequently that they cannot help but appear together and in turn generate a non-coincidental rule. For example, in an obstetrician medical dataset, we may generate a rule $\{\text{pregnant} = \text{yes}\} \rightarrow \{\text{female}\}$ with support of 0.95 and confidence 1.00. This particular rule would not be interesting as it is of common knowledge. Itemsets with low support but high confidence may seem interesting, but some of these rules may in fact be occurring due to chance, and should be considered as *noise*. For example, in a general medical dataset we may generate a rule $\{\text{meningitis}\} \rightarrow \{\text{leg fracture}\}$ with support of 0.01 and confidence 1.00. A real dataset will contain noise, which usually occurs at levels of low support. In Apriori, setting a high minimum support threshold would cut the noise out but also prune out interesting rare rules. Inherently we want to be able to detect these expensive rare rules with low support. We are interested in finding a method to filter out noise from interesting items, and detecting non-coincidental occurrences of itemsets. However up until now there has not been a method that allows us to distinguish itemsets that are occurring by coincidence. This is called the *coincidental itemset problem*.

We are interested in finding rules without having to set an ad-hoc support threshold. Here we introduce the use of the minimum absolute support function which replaces the user-defined minimum support threshold in the original Apriori algorithm. This is used to prune out the rules for itemsets that are likely occurring together due to chance. Previously, the minimum absolute support (MinAbsSup) function was introduced by Koh et al. (2006) to differentiate noise and valid itemsets in rare rule mining. Here we extend the usage of the function to the area of pruning coincidental frequent itemsets.

3.1 Apriori with the MinAbsSup Function

In this section we look at how MinAbsSup is used in Apriori as shown in the algorithm below. Only candidate itemsets whose support is above their calculated MinAbsSup value will be extended. As the MinAbsSup value is generated on-the-fly for each candidate itemset, we no longer need to use a user-defined minimum support threshold. Note that only rules with support larger than their MinAbsSup value will be kept, the rest are pruned just as if their support was less than the user-defined minimum support in standard Apriori.

Apriori with MinAbsSup pruning algorithm

Input: Transaction database D , universe of items I ,
threshold θ

Output: Non-coincidental frequent itemsets

$N \leftarrow |D|$

$Idx \leftarrow \text{invert}(D, I)$

$k \leftarrow 1$

$L_k \leftarrow \{\{i\} \mid i \in \text{dom } Idx, \text{count}(\{i\}, Idx) \geq 1\}$

while ($L_k \neq \emptyset$)

$k \leftarrow k + 1$

$C_k \leftarrow \{x \cup y \mid x, y \in L_{k-1}, |x \cap y| = k - 2\}$

$L_k \leftarrow \{c \mid c \in C_k, \text{MinAbsSup_check}(c, L_{k-1}, N, Idx, \theta)\}$

end while

return $\bigcup_{t=2}^{k-1} L_t$

MinAbsSup check, MinAbsSup_check(c, L, N, Idx, θ)

Input: Itemset c , Level L , Size of dataset N , Inverted Index Idx , threshold θ

Output: True or False

$i \leftarrow \arg \min \{j \mapsto \text{count}(j, Idx) \mid j \in c\}$

$a \rightarrow \text{count}(\{i\}, Idx)$

$b \rightarrow \text{count}(c \setminus \{i\}, Idx)$

return $(\forall x \in c \mid c \setminus \{x\} \in L) \text{ AND } \text{count}(c, Idx) > \text{MinAbsSup}(N, a, b, \theta)$

4 Experimental Results and Performance Study

We compared the performance of the MinAbsSup function and the standard Apriori algorithm on eleven different datasets from the UCI Machine Learning Repository [11]. Table 1 displays the results from our implementation of Apriori with MinAbsSup, and normal Apriori. Each row of the table represents an attempt to find rules from the database named in the left-most column. In the experiments, minconf is set to 0.90. For the Apriori algorithm, this involves setting minimum support to include itemsets that occur more than once. To give an indication of the amount of work Apriori is doing to find low-support rules, we set a time constraint of ten thousand seconds to process each dataset.

Table 1. Experiment results

Dataset	MinAbsSup Function (minconf = 0.90)				Apriori (minconf = 0.90)				
	Rules	Pass	Avg Freq Itemsets	Time (s)	Minsup	Rules	Pass	Avg Freq Itemsets	Time (s)
Lenses	1	3	4	0.1	0.000	83	5	34	0.1
LiverDis.	3	3	79	11.4	0.000	5286	8	312	9.4
TeachingEval.	12	4	19	0.7	0.000	5519	272	7	2.1
Bridges	93	6	21	1.2	0.000	632257	13	3000	105.3
Solar-Flare	144	6	27	1.0	0.000	7098832	14	20304	1079.0
Flag	592	7	82	13.1	0.078	35999090	16	150995	10052.9
Anneal	3146	8	232	13.6	0.003	162534621	17	129172	20981.6
Zoo	4029	8	84	2.6	0.050	79028358	18	32587	11352.0
Soybean-Large	121884	11	571	73.4	0.380	166877116	17	38237	21728.2
House-Vote	604948	15	1648	815.3	0.010	40173819	18	98444	10101.4
Mushroom	1560134	13	9492	1959.3	0.110	22748820	16	9948	10336.7

When Apriori with MinAbsSup is compared against Apriori, the reduction in the number of rules (with all possible consequent lengths) generated is drastic. The reduction ranges from a factor of 15 to 60809, depending on the particular dataset. By setting the arbitrary threshold too low we may be flooded with many trivial rules. We would need wade through the rules to find those that may be of some interest. However setting the support too high we may miss out useful rules. To take the Lenses dataset as an example, normal Apriori finds 83 rules. The list below shows a subset of the rules

found using normal Apriori with its confidence and lift value. We concentrate on this particular subset because they contain similar a consequent. The rest of the rules in the subset were not found as the itemsets could not be differentiated from noise.

$\{\text{astigmatic} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 1, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 1, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 2, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 2, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 3\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 3, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{classes} = 3, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{astigmatic} = 1, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1, \text{classes} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1, \text{classes} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1, \text{classes} = 3\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 1, \text{astigmatic} = 1, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 2, \text{astigmatic} = 1, \text{classes} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 2, \text{astigmatic} = 1, \text{classes} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 2, \text{astigmatic} = 1, \text{classes} = 3\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 2, \text{astigmatic} = 1, \text{spectacle} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60
 $\{\text{age} = 2, \text{astigmatic} = 1, \text{spectacle} = 2\} \rightarrow \{\text{tear production rate} = 3\}$ 1.00, 1.60

From this particular grouping Apriori with MinAbsSup finds $\{\text{astigmatic} = 1\} \rightarrow \{\text{tear production rate} = 3\}$. Note that our algorithm did not find the rest of the rules. Note that all these rules have the same lift and confidence value. All of the rules have the same consequent $\{\text{tear production rate} = 3\}$. From the set below, we did not find trivial rules. Trivial rules are rules whose antecedents cover exactly the same records as one of their parent rules. From the list the rules found the rule $\{\text{astigmatic} = 1\} \rightarrow \{\text{tear production rate} = 3\}$ is considered as the parent rule.

Note that the set of rules generated by normal Apriori in this section should not be considered as the most compact set of rules. In order to obtain a compact set of rules, we require some form of post-pruning method to eliminate trivial and redundant rules. Some plausible pruning techniques have been previously researched [129]. However, for experimental purposes we are evaluating the performance of our algorithm without other pruning techniques. The results here show that MinAbsSup reduces the need for post-pruning, and this in turn reduces time and space requirements.

5 Conclusion

Setting a suitable minimum support threshold has been investigated by many researchers. In this paper, we introduced the MinAbsSup function which replaces the fixed minimum support threshold of standard Apriori. This calculates a custom minimum support

for each itemset based on the itemset's probability of chance collision, preventing coincidental rules from being generated. Here we show that MinAbsSup efficiently finds rules which are non-coincidental without using arbitrary support thresholds. In this paper we are only concerned about setting a suitable threshold. To produce non-trivial and non-redundant rules we still would benefit from some form of post pruning technique.

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of the 20th International Conference on Very Large Data Bases VLDB, Santiago, Chile, pp. 487–499 (1994)
2. Liu, B., Hsu, W., Ma, Y.: Mining association rules with multiple minimum supports. In: Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 337–341 (1999)
3. Yun, H., Ha, D., Hwang, B., Ryu, K.H.: Mining association rules on significant rare data using relative support. *The Journal of Systems and Software* 67(3), 181–191 (2003)
4. Wang, K., He, Y., Han, J.: Pushing support constraints into association rules mining. *IEEE Transactions Knowledge Data Engineering* 15(3), 642–658 (2003)
5. Koh, Y.S., Rountree, N., O'Keefe, R.: Finding non-coincidental sporadic rules using apriori-inverse. *International Journal of Data Warehousing and Mining* 2(2), 38–54 (to appear, 2006)
6. Tao, F., Murtagh, F., Farid, M.: Weighted association rule mining using weighted support and significance framework. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 661–666. ACM Press, New York (2003)
7. Brin, S., Motwani, R., Silverstein, C.: Beyond market baskets: generalizing association rules to correlations. *SIGMOD Rec.* 26(2), 265–276 (1997)
8. Silverstein, C., Brin, S., Motwani, R.: Beyond market baskets: Generalizing association rules to dependence rules. *Data Mining and Knowledge Discovery* 2(1), 39–68 (1998)
9. Meo, R.: Theory of dependence values. *ACM Trans. Database Syst.* 25(3), 380–406 (2000)
10. Wu, X., Zhang, C., Zhang, S.: Efficient mining of both positive and negative association rules. *ACM Trans. Inf. Syst.* 22(3), 381–405 (2004)
11. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
12. Huang, S., Webb, G.: Pruning derivative partial rules during impact rule discovery. In: Ho, T.-B., Cheung, D., Liu, H. (eds.) PAKDD 2005. LNCS (LNAI), vol. 3518, pp. 71–80. Springer, Heidelberg (2005)

Transaction Clustering Using a Seeds Based Approach

Yun Sing Koh and Russel Pears

School of Computing and Mathematical Sciences,
Auckland University of Technology, New Zealand
{ykoh, rpears}@aut.ac.nz

Abstract. Transaction clustering has received a great deal of attention in the past few years. Its functionality extends well beyond traditional clustering algorithms which basically perform a near-neighbourhood search for locating groups of similar instances. The basic concept underlying transaction clustering stems from the concept of large items as defined by association rule mining algorithms. Clusters formed on the basis of large items that are shared between instances offer an attractive alternative to association rule mining systems. Currently, none of the techniques proposed offer a good solution to scenarios where large items overlap across clusters. In this paper we overcome the aforementioned limitations by using cluster seeds that represent initial centroids. Seeds are generated from sets of transaction items that occur together above a certain threshold and such seeds may overlap in their itemsets across clusters.

1 Introduction

Clustering is the process of finding naturally occurring groups in data. Clustering is one of the most widely studied techniques in the context of data mining and has many applications, including disease classification, image processing, pattern recognition, and document retrieval. Traditional clustering techniques deal with horizontal segmentation of data, whereby clusters are formed from sets of non-overlapping instances. Many efficient algorithms exist for the traditional clustering problem [1,2,3,4]. In contrast, transaction clustering has fundamentally different requirements, and has been gaining increasing attention in recent years. Unlike traditional clustering, transaction clustering requires that transactions be partitioned across clusters in such a manner that instances within a cluster share a common set of large items, where the concept of large follows the same meaning attributed to frequent items in association rule mining [5]. Thus it is clear that transaction clustering requires a fundamentally different approach from the traditional clustering techniques. Compounding the level of difficulty is the fact transaction data is known to have high dimensionality, sparsity, and a potentially large number of outliers [6].

Current research in both data mining and information retrieval suggests that transaction clustering functionality needs to extend well beyond a near-neighbourhood search for similar instances [7,8]. In this paper we propose a new approach to the problem of transaction clustering based on an initial seeding of cluster centroids. Our approach consists of two phases: a seed generation phase followed by a transaction allocation phase.

2 Related Work

In the recent past there has been an increasing level of interest in transaction clustering. All such approaches have employed quite different methods when compared to traditional clustering methods. Wang et al. (1999) utilised the concept of large items [5] to cluster transactions. Their approach measures the similarity of a cluster based on the large items in the transaction dataset. Each transaction is either allocated to an existing cluster or assigned to a new cluster based on a cost function. The cost function measures the degree of similarity between a transaction and a cluster based on the number of large and small items shared between that transaction and the given cluster. To speed-up the method proposed above, Yun et al. (2001) introduced a method called SLR (Small-Large Ratio). Their method essentially uses the measurement of the ratio between small to large items to cluster transactions. Both the large item [7] and SLR [9] method suffer a common drawback. In some cases, they may fail to give a good representation of the clusters. Xu et al. (2003) proposed a method using the concept of a caucus. The basic idea of introducing a caucus to cluster transactions is motivated by the fact that cluster quality is sensitive to the initial choice of cluster centroids [6]. Fundamentally different from most other clustering algorithms, their approach attempts to group customers with similar behaviour. In their approach they first determine a set of background attributes from the dataset that are significant. A set of caucuses, consisting of different subsets of items is then constructed to identify the initial cluster centroids. The main drawback of this method is that it requires the user to define the initial centroids which is difficult as it requires some form of prior knowledge about the dataset.

3 Transaction Clustering by Seeding

Let $D = \{t_1, \dots, t_n\}$ be a set of transactions. Each transaction is a set of items $\{i_1, \dots, i_m\}$. C is a partition of the transaction, $\{C_1, \dots, C_k\}$ of $\{t_1, \dots, t_n\}$. Each C_i is called a cluster. Overall the clustering is divided into two main phases: seed generation and allocation phases.

3.1 Seed Generation Phase

We start by describing a method for finding the optimal number of clusters. Our initial choice of seeds are the large items in the dataset and we thus begin by setting a minimum support threshold, θ , where $0 < \theta < 1$. Any item in the dataset that has support above $|D| * \theta$ is considered a large item. Let L_i denote the set of large items or large itemsets. We now allow the items L_i to be extended to itemsets L_{i+1} in the same way as Apriori generates candidate frequent itemsets. For a large itemset to be considered a cluster seed the frequency of co-occurrence of all pairs of subsets within the seed must occur together with a frequency above a threshold value at a given significance level. This effectively ensures that cluster seeds of size ≥ 2 have items that co-occur together at a frequency that is statistically significant. In addition, we require that all cluster seeds satisfy an improvement constraint when they are extended. We first define the concept of relative support.

Definition 1 (Relative Support). The relative support of an itemset X_k of size k is defined to be the ratio of the support of X_k to the support of Y_{k-1} which is that $(k - 1)$ -sized subset of X_k with the maximum support. Thus,

$$RS(X_k) = \frac{supp(X_k)}{supp(Y_{k-1})}$$

Definition 2 (Extension of a Seed). Given two existing seeds, X_{k-1} and Y_{k-1} , X_{k-1} is extended to a new seed $X_{k-1} \cup Y_{k-1}$ if and only if:

$$\begin{aligned} \phi(X_{k-1}, Y_{k-1}) &> \chi_c^2, \\ RS(X_{k-1} \cup Y_{k-1}) - RS(X_{k-1}) &> \sigma, \text{ and} \\ RS(X_{k-1} \cup Y_{k-1}) - RS(Y_{k-1}) &> \sigma \end{aligned}$$

where ϕ denotes the chi square correlation coefficient, χ_c^2 , the chi square cut-off threshold at the $c\%$ confidence level and σ is a user-defined threshold.

The rationale behind extension lies in the fact that the new itemset to be added to the seed has a statistically strong correlation with the existing seed and that the inclusion of the new itemset will improve the relative support of the seed above a user defined minimum threshold. The algorithm for the seed clustering phase is shown below.

Algorithm Seed Generation Phase

Input: Transaction database D , θ value, σ value, universe of items I

Output: Cluster Seeds, $S = \{s_1 \dots s_k\}$

$k \leftarrow 1$

$s_k \leftarrow \{\{i\} | i \in I, \text{count}(\{i\}) \geq |D| * \theta\}$

while $l_k \neq \emptyset$ do

$k \leftarrow k + 1$

$l_k \leftarrow \{x \cup y | x, y \in s_{k-1}, |x \cap y| = k - 2\}$

$s_k \leftarrow \{x \cup y | x \cup y \in l_k, \phi(x, y) \geq \chi_c^2, RS(x \cup y) - RS(y) > \sigma,$

$RS(x \cup y) - RS(x) > \sigma\}$

end while

return $\bigcup_{t=1}^{k-1} s_t$

3.2 Allocation Phase

The seeds produced in the initial phase are considered as the initial centroids for the clusters. In this phase, transactions are assigned to clusters on the basis of similarity to cluster centroids. In order to measure similarity we use a modified version of the Jaccard similarity coefficient [10]. For each transaction, t , we calculate the similarity between t and the existing centroid, c_k . The similarity, sim , is between t and the c_k is calculated as:

$$sim(t, c_k) = \frac{|t \cap c_k|}{|t \cup c_k| - |t \cap c_k| + 1}$$

Given $t_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$ and $c_1 = \{\{b\}, \{c\}\}$, here $t_1 \cap c_1 = \{\{b\}, \{c\}\}$ and $t_1 \cup c_1 = \{\{a\}, \{b\}, \{c\}, \{d\}, \{e\}\}$. Using our measure, the similarity between t_1 and c_1 is calculated as $2/(5 - 2 + 1) = 0.5$. The greater the overlap between t and C_k , the greater the value of sim coefficient.

Algorithm Allocation Phase

Input: Transaction database, $D=\{t_1, \dots, t_n\}$, Cluster Seed, $S=\{s_1, \dots, s_k\}$
Output: Cluster, $C = \{C_1, \dots, C_k\}$

```

 $J_{prev} \leftarrow 0$ 
 $C \leftarrow \{C_k \leftarrow \emptyset | k \in S\}$ 
/* Assign transactions to clusters with the highest similarity */
 $C \leftarrow \{C_k \cup t | \arg \max\{k \mapsto \text{sim}(t, s_k) | s_k \in S\}, t \in D\}$ 
 $C \leftarrow \{C_k | C_k \neq \emptyset, C_k \in C\}$  /* Removes the empty clusters */
 $J_{curr} \leftarrow \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{\sum_{t \in C_j} \text{sim}(t, c_j)}{|C_j|}$ 
/* Refine clusters */
while  $J_{prev} < J_{curr}$  do
     $J_{prev} \leftarrow J_{curr}$ 
     $c \leftarrow \{C_k | \{i | i \in C_k, \text{count}(\{i\}, D) \geq |D| * \theta\}, C_k \in C\}$ 
     $C \leftarrow \{C_k \cup t | \arg \max\{k \mapsto \text{sim}(t, c_k) | c_k \in c\}, t \in D\}$ 
     $C \leftarrow \{C_k | C_k \neq \emptyset, C_k \in C\}$ 
     $J_{curr} \leftarrow \frac{1}{|C|} \sum_{j=1}^{|C|} \frac{\sum_{t \in C_j} \text{sim}(t, c_j)}{|C_j|}$ 
end while
return  $C$ 

```

Once all transactions are allocated to clusters, further refinement is accomplished by recomputing the centroids which may need to be updated with large items belonging to transactions allocated to a given cluster but not presently part of its centroid. The updating of centroids will result in the need for reorganisation of the clusters, thus the process of centroid update and cluster reorganisation will need to be repeated in tandem until a suitable point of stabilisation is reached. In order to determine the point at which stabilisation is reached, we use a fitness function adapted from particle swarm optimisation approach was proposed to find the optimal clusters. For all cluster $\{C_1, \dots, C_k\}$, the fitness function is calculated as:

$$J = \frac{1}{k} \sum_{j=1}^k \frac{\sum_{t \in C_j} d(t, c_j)}{|C_j|}$$

Typically, we want to maximise the fitness value generated. The fitness measure calculates the average similarity between every transaction in a cluster to its centroid. We show the algorithm for allocation phase as Algorithm 2 above.

4 Experimental Results

In order to evaluate the effectiveness of our seed based approach to transaction clustering we conducted an experimental comparison with the large item approach [7]. We used seven different real world datasets taken from the UCI Machine Learning Repository [11]. The first stage of analysis involved an overall comparison of cluster quality. Secondly, we selected one dataset, namely the Congressional Vote and conducted a more in depth analysis of the properties of the clusters produced by the two approaches.

In this section we report on cluster quality, as measured by the Root Mean Square Standard Deviation or RMSSTD index [12]. Table 1 below shows that our Cluster Seed-

ing approach outperforms Large Items across all datasets tested. In terms of cluster quality the cluster seeding algorithm consistently returned lower RMSSTD values than its Large Item counterpart. With respect to processing time the Cluster Seeding approach returned run times that were consistently lower, with the difference in timing between the two approaches widening with increasing database size, as evidenced with the soybean and mushroom dataset. Thus we can conclude that the cluster seeding approach scales better with respect to dataset size.

Table 1. Experiment results

Dataset	No Trans	Cluster Seeding			Large Item		
		No Clusters	RMSSTD Index	Time(s)	No Clusters	RMSSTD Index	Time(s)
Zoo	101	7	21.2	4.4	7	24.9	13.7
Bridges	108	8	17.2	3.1	7	18.7	4.3
Hepatitis	155	8	25.9	3.6	8	26.5	23.1
Flag	194	8	40.8	16.7	10	41.7	188.3
Soybean-Large	307	5	44.8	15.1	5	45.2	239.2
Congressional Votes	435	5	19.9	15.8	4	20.6	155.6
Mushroom	8124	8	28.5	972.78	6	30.0	28684.0

The Congressional Votes dataset consists of the United States Congressional Voting Records in 1984. Each record represents one Congressman’s vote on 16 different issues. In order to make the comparison between the algorithms fair, we ran both algorithms with settings that resulted in the same number of clusters. Both algorithms produced 4 clusters, out of which three had the vast majority of instances labelled as Democrat while the other had a clear majority of instances with the Republican label. However, an investigation of the homogeneity within the clusters revealed significant differences in the formation of the clusters. Table 2 shows these differences.

Table 2. Comparison between Seed Clustering and Large Items on the Votes dataset

Cluster Seeding				Large Item			
Clus	Cluster Label	Coverage	No of Non-Homogeneous Attributes	Clus	Cluster Label	Coverage	No of Non-Homogeneous Attributes
0	Republican (95.1%)	46.6%	3	0	Republican (96.3%)	46.9%	3
1	Democrat (96.8%)	26.7%	4	1	Democrat (96.5%)	33.6%	7
2	Democrat (94.3%)	15.1%	3	2	Democrat (100%)	9.9%	10
3	Democrat (92.6%)	11.6%	5	3	Democrat (100%)	9.6%	8

The Cluster Label column indicates the party label belonging to the majority of instances in a given cluster, with the number beside it denoting the percentage of instances in that cluster that contain the label. The Coverage column tracks the percentage of instances falling into a given cluster. For each cluster we record the support received by each attribute; if this support falls below 70% then we consider the attribute to be

non-homogeneous. The most significant differences between the two approaches are apparent when we compare the number of non-homogeneous attributes. It is clear from Table 2 that the Cluster Seeding approach produces clusters with a much higher degree of homogeneity with an average value of 3.75 for the number of non-homogeneous attributes, versus 7.0 for the Large Items approach.

In order to further quantify the differences between the two approaches we focused on the three clusters containing Democrats as the two algorithm performed very similarly for the Republican cluster. Ideally, a clusterer should show sharp differences in voting patterns between the three Democrat clusters. We used the set symmetric operator to assess the difference in voting patterns amongst Democrats. We evaluate $C_i \oplus C_j = (C_i - C_j) \cup (C_j - C_i)$ for pairs of values (i, j) in the range $[1 \dots 3]$ for each of the two clusters. Table 3 summarises the results. The larger the value of the set symmetric operator the larger the contrast or difference between the clusters involved. Table 3 shows that the Cluster Seed algorithm produces a better differentiation between the Democrat clusters with an overall set symmetric cardinality of 25 as opposed to 14 for the Large Item approach.

Table 3. Summary of differences in voting patterns across different combinations of Democrat clusters

Cluster Seeding		Large Item	
Cluster Combination	Set Symmetric Cardinality	Cluster Combination	Set Symmetric Cardinality
C1, C2	4	C1, C2	7
C2, C3	11	C2, C3	4
C1, C3	10	C1, C3	3

5 Conclusion

In this paper we proposed a new approach to the problem of transaction clustering. Our approach differed from previous work in that we used seeds containing frequent items to guide the allocation of transactions to clusters. Our seeds were generated in such a fashion to actively promote the presence of frequent items across different clusters. Our experimentation on several real world datasets showed that our approach produced clusters with a much higher degree of homogeneity when compared to the current state-of-the-art algorithm.

References

1. Jain, A.K., Murty, M.N., Flynn, P.J.: Data clustering: a review. *ACM Comput. Surv.* 31(3), 264–323 (1999)
2. Ganti, V., Gehrke, J., Ramakrishnan, R.: CACTUS: Clustering categorical data using summaries. In: *KDD 1999: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 73–83. ACM Press, New York (1999)

3. Gibson, D., Kleinberg, J.M., Raghavan, P.: Clustering categorical data: An approach based on dynamical systems. In: VLDB 1998: Proceedings of the 24rd International Conference on Very Large Data Bases, pp. 311–322. Morgan Kaufmann Publishers Inc., San Francisco (1998)
4. Guha, S., Rastogi, R., Shim, K.: ROCK: A robust clustering algorithm for categorical attributes. *Information Systems* 25(5), 345–366 (2000)
5. Agrawal, R., Imielinski, T., Swami, A.: 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)
6. Xu, J., Xiong, H., Sung, S.Y., Kumar, V.: A new clustering algorithm for transaction data via caucus. In: Whang, K.-Y., Jeon, J., Shim, K., Srivastava, J. (eds.) PAKDD 2003. LNCS (LNAI), vol. 2637, pp. 551–562. Springer, Heidelberg (2003)
7. Wang, K., Xu, C., Liu, B.: Clustering transactions using large items. In: CIKM 1999: Proceedings of the Eighth International Conference on Information and Knowledge Management, pp. 483–490. ACM Press, New York (1999)
8. Cutting, D.R., Pedersen, J.O., Karger, D., Tukey, J.W.: Scatter/gather: A cluster-based approach to browsing large document collections. In: Proceedings of the Fifteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 318–329 (1992)
9. Yun, C.H., Chuang, K.T., Chen, M.S.: An efficient clustering algorithm for market basket data based on small large ratios. In: COMPSAC 2001: Proceedings of the 25th International Computer Software and Applications Conference on Invigorating Software Development, pp. 505–510. IEEE Computer Society, Washington (2001)
10. Ivchenko, G.I., Honov, S.A.: On the jaccard similarity test. *Journal of Mathematical Sciences* 88(6)
11. Newman, D., Hettich, S., Blake, C., Merz, C.: UCI repository of machine learning databases (1998), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
12. Sharma, S.: Applied multivariate techniques. John Wiley & Sons Inc., Chichester (1996)

Using Ontology-Based User Preferences to Aggregate Rank Lists in Web Search

Lin Li¹, Zhenglu Yang¹, and Masaru Kitsuregawa²

¹ Dept. of Info. and Comm. Engineering, University of Tokyo, Japan

² Institute of Industrial Science, University of Tokyo, Japan
{lilin, yangzl, kitsure}@tkl.iis.u-tokyo.ac.jp

Abstract. This paper studies rank aggregation by using ontology-based user preferences in the context of Web search. We introduce a set of techniques to combine the respective rank lists produced by different attributes of user preferences. Furthermore, the learned user preferences are structured as a taxonomic hierarchy (a simple ontology). We use the learned ontology to store the attributes such as, the topics that a user is interested in and the degrees of user interests in these topics. The primary goal of our work is to form a broadly acceptable rank list among these attributes by making use of *rank-based* aggregation. Experiment results on a real click-through data set show that our user-centered rank aggregation techniques are effective in improving the quality of the Web search in terms of user satisfaction.

1 Introduction

Nowadays, it becomes increasingly difficult for users to retrieve desired information due to the continued rapid growth in data volume and the ambiguity of short queries in Web searches. As we know, different users have different intentions for a same query. In order to satisfy the diverse needs of users, search engines should be adaptive to the individual contexts in which users submit their queries. Lawrence et al. [6] addressed an overview of the context of the Web search. User preferences are a kind of useful contexts. Shen et al. [11] developed a client-side Web search agent to perform implicit feedback and inferred user model from short-term search contexts to improve Web searches. The user preferences can be represented by a bag of words or a taxonomic hierarchy. The bag of word representation does not consider term correlations because terms in user preferences are considered in isolation from one another. The taxonomic hierarchy can overcome this drawback and has been widely accepted [2, 7, 10]. It is also the basic structure of modeling our user preferences. Furthermore, user preferences consist of a number of attributes, such as what kind of topics that users are interested in, and how much users are interested in each topic. Each attribute describes a user's favorite in different aspects. In most cases, any individual attribute is deficient in accurately representing user preferences. Combining user knowledge depicted by each attribute can help us understand user preferences well, which finally results in an effective rank mechanism in the Web search. To leverage

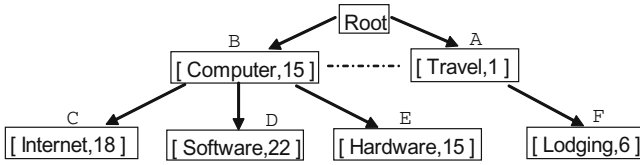


Fig. 1. Hierarchical Model of User Preferences

the rankings produced by the different attributes, rank aggregation intends to form a single rank list supported by a broad consensus among these attributes. There are two approaches: score-based and rank-based. The score-based rank aggregation merges the values of the attributes [2, 12]. However, it is important to observe that if the rank mechanism is score-based, the sequence implied by the scores makes it more meaningful than the actual scores themselves. On the other hand, the rank-based rank aggregation fuses the rank lists produced by the values of the attributes and has been studied and employed in many applications in the last half century [4, 9, 14]. Renda et al. [9] compared rank and score based methods without training data in the context of metasearch, and showed that Markov chain rank-based methods compete with score-based methods. Dwork et al. [4] developed the theoretical groundwork for describing and evaluating rank aggregation methods. Their main point is to effectively combat *spam*. In this paper we introduce methods to effectively improve the Web search in a context-aware manner.

In the rest of this paper, Section 2 describes rank-based rank aggregation, including how to produce and fuse user-centered rank lists. We report the experimental results in Section 3, and draw conclusions in Section 4. From now on, the term “rank aggregation” means “rank-based” rank aggregation for simplicity.

2 Rank Aggregation

In the following part, we will discuss how to get the respective rank lists from the learned use preferences and the proposed rank aggregation methods.

2.1 Hierarchical Similarity Measures

Our user preferences are structured as a semantic hierarchy shown in Figure 1. Technical details about how to learn and update user preferences from click-through data are in our previous work [7]. For an effective rank mechanism, the more similar a search result is to user preferences, the higher position it will be put in the final rank list. To produce such a new rank list, hierarchical similarity measures are needed to assess the relatedness between user preferences and search results. We choose five content-ignorant measures from [8] because we want to see how much we can benefit from the hierarchical structure. The measures are defined as

$$S_1(i, j) = 2 \cdot M - l, \tag{1}$$

$$S_2(i, j) = \alpha S_1(i, j) + \beta h \quad (\alpha = 0.05, \beta = 1), \quad (2)$$

$$S_3(i, j) = e^{-\alpha \cdot l} \quad (\alpha = 0.25), \quad (3)$$

$$S_4(i, j) = \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \quad (\beta = 0.15), \quad (4)$$

$$S_5(i, j) = e^{-\alpha \cdot l} \cdot \frac{e^{\beta \cdot h} - e^{-\beta \cdot h}}{e^{\beta \cdot h} + e^{-\beta \cdot h}} \quad (\alpha = 0.2, \beta = 0.6), \quad (5)$$

where h means the depth of the subsumer (the deepest node common to two nodes), l is the naïve distance (the number of edges or the shortest path length between two nodes), i and j are nodes (topics) in Figure 1, and M is the maximum depth of topic directory possessed by user preferences. The values in parentheses are the optimal values of parameters [8].

2.2 Rank Lists Produced by Ontology-Based User Preferences

The above five content-ignorant measures can evaluate the hierarchical similarity between search results and user preferences. The degree of user preferences has effect on the similarity as well. There are various ways of combining the two kinds of similarity scores dependent on applications. Their product is commonly used in classic IR like our previous work. However, as we mentioned in Section 1, the ranking implied by the scores has more sense than the actual scores themselves. In the following discussion we calculate two user-centered rank lists plus the result list returned by Google for rank-based fusion, as distinguished from the traditional score-based combination.

(1) **Hierarchical Semantic Similarity.** User preferences include a number of topics (nodes) in Figure 1. We further define the semantic similarity between one search result and one user as the maximum value among all the values computed by any one of Equations 1-5. The search results then are re-ranked and form a rank list in order of one attribute of user preferences (i.e., the topics a user is interested in). The priori work [2], however, just selected one of them without analyzing their differences.

The five measures have their own features from their definitions. For example, compared to Equation 1, Equation 3 also uses the naïve distance alone, but makes use of a nonlinear function. Equation 2 is a linear combination of the naïve distance and the depth. Different from Equation 2, Equation 5 transfers the naïve distance and the depth by a nonlinear function, respectively, and then combines them by multiplication. Equation 4 is the transformation of the depth of the subsumer through a nonlinear function. Based on these differences, we think that it is necessary to experimentally compare their performances when they are applied in the context of the Web search and no priori work has done it. The experimental results are reported in Section 3.

(2) **Degree of User Interests.** We find that Equations 1-5 are not round in re-ordering search results. With the help of Figure 1, let us explain the problem clearly. The naïve distance between node A and node C (i.e., 3) is the same as that between node B and node F , and the subsumer of A and C (i.e., “root”) is the same as that of B and F as well. As a result, computed by any equation from

Equations 1.5, the similarity score between A and C is equal to that between B and F . In this situation, these measures cannot order the two pairs. Our solution for this problem is intuitive that the degree of the user interests in a topic (node) can alleviate this problem. The more times a user clicks one topic, the more interested the user is in it. The user's clicked times can produce a complementary rank list of search results.

(3) **Google List.** Google applies its patented PageRank technology on the Google Directory to rank the sites. To keep our rank aggregation from missing the high quality Web pages in Google, we also consider the original rank list of Google Directory Search. As we know, there is a PageRank value accompanied with each search result, representing the popularity or authority of results. It certainly could be used to weigh the topics associated with results. Unfortunately, these values are not publicly available for the present, but the ordering of search results can be easily obtained. From this point of view, our rank-based aggregation is suitable in this situation since it is exactly good at processing rank lists. Certainly, it is reasonable for us to guess the approximate values of PageRank if we favor the score-based combination, but this topic is out of the scope of this paper. In our methods the original rank lists as inputs can intactly and unbiasedly reflect Google's standpoint.

2.3 Rank Aggregation Methods

We study the problem of combining sets of rank lists from different attributes of user preferences into a single rank list. Voting provides us with a traditional class of algorithms to determine the aggregated rank list. The most common voting theory, named after its creator, is known as Borda's rule [1] which argues that the majority opinion is the truth, or at least the closest that we can come to determining it [13]. However, the problem with Borda's rule is that it does not optimize any criterion. We make use of Footrule distances [3] to weigh edges in a bipartite graph and then find a minimum cost matching. This method was proved in [4] to approximate the optimal ranking that approximately minimizes the number of disagreements with the given inputs.

Modified Borda's Rule. Borda's rule is a single winner election method in which voters rank candidates in order of preferences. The winner of an election is determined by giving each candidate a certain number of points corresponding to the position in which she is ranked by each voter. Once all points have been counted, the candidate with the most points is the winner.

Our idea is that we treat each attribute of user preferences as a voter. It means that each attribute re-orders the search results in the same way as each voter selects a list of candidates. Let $A = a_1, a_2, \dots, a_m$ be the set of positions in the rank list, and let the attributes of user preferences plus the result list of Google be named by elements of n (i.e., n voters in an election). We shall assume for the present that every element of n can be expressed by a linear order in the position set A . We denote a linear order by a sequence $A_i = a_{i_1}, a_{i_2}, \dots, a_{i_m}$ where for $j < k$, a_{i_j} is preferred to a_{i_k} . For each voter, the ranked results

should be given some points. The closer a search result is to the top of the list, the more points it will be given. Especially in the context of the Web search, the top search results have much higher possibility to be clicked than others. Most Web search users just browse the top 10 or 20 results. If they do not find the desired information, they will modify their queries to start a new search, instead of continuing checking the results. Therefore, modified Borda's rule is applied here. The voter awards the first-ranked candidate with one point (i.e., 1). The second-ranked candidate receives half of a point (i.e., 1/2), the third-ranked candidate receives on a third (i.e., 1/3), etc. This kind of point distribution gives more weights to the top results. When all elements of n have been counted, and each A_i can be thought of as a position vector, we sort the search results by several formulas, defined as

$$L_1(a_k) = \sum_{i=1}^n 1/a_{i_k}, \quad L_2(a_k) = \sqrt{\sum_{i=1}^n (1/a_{i_k})^2}, \quad (6)$$

$$GM(a_k) = \left(\prod_{i=1}^n 1/a_{i_k} \right)^{1/n}. \quad (7)$$

Equation 6 represents the L_1 norm and the L_2 norm of these position vectors, and the geometric mean of the n points is expressed in Equation 7. We take into consideration the median of the n points as well. Borda's rule is commonly classified as a positional voting system because from each voter, candidates receive a certain number of points. Computationally it is very easy, as it can be implemented in linear time.

Bipartite Graph. Borda's rule does not assure us that it can find the optimal rank list because it does not optimize any criterion. A graph theory based method is proposed here, to approximate the optimal ranking. We define a weighted balanced bipartite graph $G = (V_1 \cup V_2, W)$. $V_1 = r_1, r_2, \dots, r_m$ is a set of search results to be ranked. $V_2 = p_1, p_2, \dots, p_m$ is the m available positions in the rank list. For any two vertices $r \in V_1$ and $p \in V_2$, rp is an edge in G ; thus G is also a complete bipartite graph. The weight $W(r, p)$ is the total distance of a ranking value that places r at position p . The task of rank aggregation is to minimize the number of disagreements with the respective lists. Therefore, if all the search results are put in proper positions, the total distance (i.e., the number of disagreements) should be the smallest. Now we meet two difficulties in achieving this goal. One is how to compute the distance. The other one is what kind of approaches can minimize the distance.

To weigh the edges in G , according to Diaconis et al. [3], the two distance measures that we consider are:

$$Footrule_D(\pi, \sigma) = \sum_{i=1}^n |\pi(i) - \sigma(i)|, \quad Footrule_S(\pi, \sigma) = \sum_{i=1}^n (\pi(i) - \sigma(i))^2, \quad (8)$$

where π and σ are regarded as rank lists. Diaconis et al. [3] also suggest two other measures. One roughly seems similar to *Footrule_D*, and the other is unsuitable for general use, having very small variance about a mean that is very close to its maximum value. Therefore, we choose *Footrule_D* and *Footrule_S* here. We then adjust the two measures to compute the total distance that is the weight in an edge, now defined as $\sum_i^n |A_i(r) - p|$ or $\sum_i^n (A_i(r) - p)^2$. Minimizing the total distance to n could be solved by the well-known Hungarian algorithm that finds a minimum cost perfect matching in the bipartite graph. A matching in a graph is a set of edges where no two of which share an endpoint. The most similar work to ours is Dwork et al. [4] who only used *Footrule_D* as the distance measure. However, our experiments compared the two measures and observed that *Footrule_D* performed the worst among all the methods, even inferior to the score-based method. The largest improvement is reached by *Footrule_S*. In addition, their main application is to effectively combat *spam* while we study the rank aggregation in terms of user preferences to improve the Web search.

3 Experiments

3.1 Dataset and Evaluation Metrics

Given a query, Google API offered us the top 20 search results. In order to collect the real click-through data, we randomized the order of the results before returning them to 12 invited users and asked them to evaluate whether the clicked results are relevant or not. After the data were collected over a ten-day period (From October 23nd, 2006, to November 1st, 2006), we had a log of about 300 queries averaging 25 queries per subject and about 1200 records of the clicked Web pages in total. The evaluation metrics are listed as follows.

(1) **AvgRank** indicates the average rank of search results, defined as:

$$AvgRank(q) = \sum_{p \in S} R(p) / |S|. \quad (9)$$

Here S denotes the set of search results selected by a subject for query q , $R(p)$ is the position of p in the result list, and $|S|$ is the cardinality of the set S . A smaller *AvgRank* represents a better quality.

(2) **DCG** [5] gives more weight to highly ranked search results, defined as:

$$DCG(i) = \begin{cases} G(1) & \text{if } i = 1 \\ DCG(i-1) + G(i)/\log(i) & \text{otherwise} \end{cases} \quad (10)$$

By averaging over a set of test queries, the average performance of our methods can be analyzed. In the experiments, we used $G(i) = 2$ for highly relevant Web pages, $G(i) = 1$ for relevant Web pages, and $G(i) = 0$ for non-relevant search results. A larger *DCG* means a better quality.

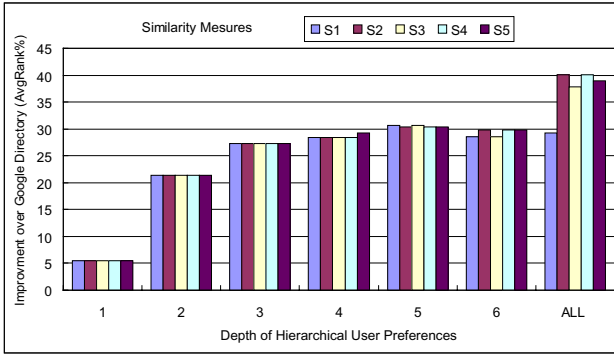


Fig. 2. Depth of Hierarchical User Preferences

3.2 Experimental Results

Depth of Topic Directory for User Preferences. The first step is to determine how deep the depth of topics is when modeling user preferences. We did a preliminary performance analysis on different depths. The re-ranking mechanism is addressed in [7]. Measured by *AvgRank* in Equation 9, Figure 2 illustrates the improvement over Google Directory Search per similarity measure versus the depths considered in learning user preferences.

It shows that the deeper the topic directory we process, the bigger improvement is generally reached. If our algorithm stores the whole topic directory, the biggest improvement is over 40%. In addition, we observed that when the depth is set to 1 (2 or 3), the five similarity measures performed almost the same. The reason is that in our dataset, most of the relevant and non-relevant search results share the same subsumer in a very low depth of the hierarchy. We need to store the deeper topic directory to tell the relevant results from the non-relevant ones. Furthermore, from Figure 2 when the depth of topic directory increases to 3, the improvement is big, from 5% to above 25%. However, when the depth is increased continually from 3 to 6, the improvement changes slowly. Due to this observation and the large size of the whole Google Directory [4], only the top 4 topic directory is encoded into the user preferences in the following experiments, which is a trade-off between accuracy and storage memory.

Effect of Similarity Measures and Rank Aggregation Methods. Figure 3 illustrates the performances of rank aggregation methods and the five similarity measures defined in Equations 1-5. The *Score-Based* method in the figure is the same as that in [7]. From this figure, the highest improvement over Google Directory Search is about 13%, produced by *Footrule_S*. L_1 norm, L_2 norm, and *Footrule_S* performed better than *Score-Based*, while the qualities of *Median*

¹ Google uses ODP as basis for its Google Directory service, and ODP has more than 590,000 categories.

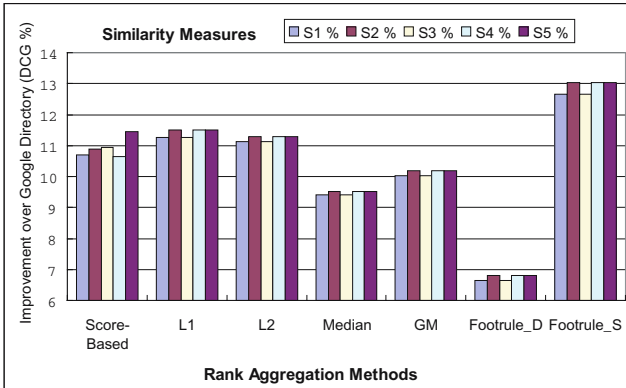


Fig. 3. Effect of Similarity Measures and Rank Aggregation Methods

and *Footrule_D* are inferior to that of *Score-Based*. Although Borda’s Rule neither optimizes any criterion nor satisfies the Condorcet property [13], this kind of method outperformed the score-based combination. The Hungarian algorithm based on the *Footrule* distance that finds a minimum cost perfect matching in the bipartite graph showed the best results obtained by the distance measure *Footrule_S* in Equation 8. In addition, we know that S2, S4, and S5 perform similarly, while S1 and S3 perform similarly as well. The reason is that the former three measures give much more weight on depth than length, and the latter two measures only consider length. Given the same length and depth, the five measures will compute different values due to different transformation functions. Thus the score-based method is easily influenced by the selected function. On the other hand, the rank-based methods are robust. Even the transformation function is different, as long as the measures take into account the same information, they will produce similar performance. Moreover, in the rank-based methods, S2, S4, and S5 performed slightly better than S1 and S3, which tells us that the depth of subsumbers carry more useful information than the naïve distance in our dataset. Note that S4 uses the depth alone, but competes with S2 and S5. In the score-based method, however, S5 is the winner, much better than the other measures.

4 Conclusions

In this paper we proposed a set of techniques for rank aggregation. Experimental results on a real click-through data set demonstrate the effectiveness of our methods. We observed that some rank-based aggregation methods performed better than the *Score-Based* method and the *Footrule_S* method performed best in our evaluation. Furthermore, we analyzed the influence of the topic depth of the ontology-based user preferences on the quality of the Web search, and compared the performances of five similarity measures. If the measures utilize similar

information from users, they will perform similarly regardless of what kind of transformation functions is being used. But the score-based combination is sensitive to the selected function. In the future we plan to put these methods into larger datasets, and further mine more user-centered information and optimize Web searches in terms of user's satisfaction.

References

- [1] Borda, J.: Mémoire sur les élections au scrutin. *Comptes rendus de l'Académie des sciences* 44, 42–51 (1781)
- [2] Chirita, P.A., Nejdl, W., Paiu, R., Kohlschütter, C.: Using ODP metadata to personalize search. In: *Proc. of SIGIR 2005*, Salvador, Brazil, pp. 178–185 (2005)
- [3] Diaconis, P., Graham, R.L.: Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)* 39(2), 262–268 (1977)
- [4] Dwork, C., Kumar, R., Naor, M., Sivakumar, D.: Rank aggregation methods for the web. In: *Proc. of WWW 2001*, Hong Kong, China, pp. 613–622 (2001)
- [5] Jarvelin, K., Kekalainen, J.: IR evaluation methods for retrieving highly relevant documents. In: *Proc. of SIGIR 2000*, Athens, Greece, pp. 41–48 (2000)
- [6] Lawrence, S.: Context in web search. *IEEE Data Eng. Bull.* 23(3), 25–32 (2000)
- [7] Li, L., Yang, Z., Wang, B., Kitsuregawa, M.: Dynamic adaptation strategies for long-term and short-term user profile to personalize search. In: *Proc. of AP-Web/WAIM 2007*, Huang Shan, China, pp. 228–240 (2007)
- [8] Li, Y., Bandar, Z., McLean, D.: An approach for measuring semantic similarity between words using multiple information sources. *IEEE Trans. Knowl. Data Eng.* 15(4), 871–882 (2003)
- [9] Renda, M.E., Straccia, U.: Web metasearch: Rank vs. score based rank aggregation methods. In: *Proc. of SAC 2003*, Melbourne, USA, pp. 841–846 (2003)
- [10] Schickel-Zuber, V., Faltings, B.: Inferring user's preferences using ontologies. In: *Proc. of AAAI 2006*, Boston, Massachusetts, USA (2006)
- [11] Shen, X., Tan, B., Zhai, C.: Implicit user modeling for personalized search. In: *Proc. of CIKM 2005*, Bremen, Germany, pp. 824–831 (2005)
- [12] Speretta, M., Gauch, S.: Personalized search based on user search histories. In: *Proc. of WI 2005*, Compiègne, France, pp. 622–628 (2005)
- [13] Young, H.P.: Condorcet's theory of voting. *American Political Science Review* 82(4), 1231–1244 (1988)
- [14] Zhu, S., Fang, Q., Deng, X., Zheng, W.: Metasearch via voting. In: Liu, J., Cheung, Y.-m., Yin, H. (eds.) *IDEAL 2003*. LNCS, vol. 2690, pp. 734–741. Springer, Heidelberg (2003)

The Application of Echo State Network in Stock Data Mining

Xiaowei Lin¹, Zehong Yang², and Yixu Song³

State Key Laboratory of Intelligent Technology and System
Tsinghua National Laboratory for Information Science and Technology
Department of Computer Science and Technology, Tsinghua University,
Beijing, China, 100084
linxw05@mails.tsinghua.edu.cn, yangzehong@sina.com,
songyixu@sohu.com

Abstract. Stock data, which is among the most complicated time series, is difficult to analyze and mine. Neural network has been a popular method for data mining in financial area since last decade. In this paper, we explore the use of Echo State Networks (ESNs) to perform time-series mining on stock markets. The Hurst exponent is applied to adaptively determine initial transient and choose sub-series with greatest predictability before training. With the capability of short-term memory provided by ESN, a stock prediction system is built to forecast the close price of the next trading day based on history prices and technical indicators. The experiment results on S&P 500 data set suggest that ESN outperforms other conventional neural networks in most cases and is a suitable and effective way for stock price mining.

Keywords: Echo State Network, Neural networks, Stock data mining, Short-term price prediction.

1 Introduction

Data mining on stock markets has received focus from investors and researchers for a long time due to its potential profits. Artificial neural networks (ANNs) are the most widely used approach on stock markets and show good performance in many cases. Among ANNs, back-propagation neural network (BPNN), time delay neural network (TDNN) and recurrent neural network (RNN) are popular. However, they have their own limitations. BPNN can learn only an input-output mapping of static patterns that is independent of time [1]. The fixed time delays of a TDNN take a risk of a mismatch between the choice of time-delay values and temporal location of important information in the input patterns [1]. RNN is difficult to develop [2].

Echo State Network (ESN) is a novel RNN whose basic idea is to use a large “reservoir” RNN as a supplier of interesting dynamics from which the desired output is combined [4]. The “reservoir” contains information about the past input history in a way which reflects the recent history well and decays with the delay time, which is consistent with stock markets. ESN is good at chaotic time-series forecast and obtained

the best result in Mackey-Glass series prediction. Because it is simple to develop, ESN has been utilized in wireless communication [3], robot control [5], speech recognition [6, 7], etc. Unfortunately, there is no application of ESN in financial area.

This paper investigates the effectiveness of ESN to predict daily stock price through history data. The Hurst exponent is utilized to choose a persistent sub-series with the greatest predictability for training. We test nearly all the stocks in S&P 500 and compare the results of ESN with BPNN, Elman neural network and radial basis-function neural network (RBFNN). The experiments demonstrate that ESN outperforms other neural networks in most cases and is an effective way in short-term stock time-series prediction.

The rest of the paper is organized as follows: Section 2 introduces ESN and our prediction system. Section 3 describes experiments and shows the results. Section 4 makes a conclusion and proposes some advice for future research.

2 Echo State Network and Stock Prediction System

2.1 Architecture of ESN

Our stock prediction system is based on a standard ESN (See Figure 1) with 600 internal units, only 5% of which are interconnected. The internal connection weight matrix is rescaled to a spectral radius of 0.1 to ensure the echo state property. The activation of internal state $x(n+1)$ at time step $n+1$ is updated according to

$$x(n+1) = 1/(1 + \exp(-\alpha \times (W^{in}u(n+1) + Wx(n) + W^{back}y(n)) + v)) \quad (1)$$

where W^{in} is input connections sampled from a uniform distribution between $[-0.1, 0.1]$; W is internal connections sampled from a uniform distribution between $[-1, 1]$; W^{back} is feedback connections sampled from the uniform distribution between $[-5, 5]$; $u(n+1)$ is the input at time step $n+1$; $y(n)$ is the output at time

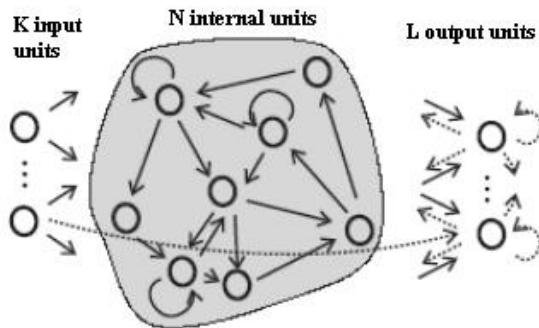


Fig. 1. The Architecture of ESN [8]

step n ; v is noise randomly sampled from $[-10^{-5}, 10^{-5}]$ and $\alpha = 5$. The output of ESN is computed according to

$$y(n+1) = W^{out}(u(n+1), x(n+1), y(n)) \quad (2)$$

where W^{out} is output connections. In ESN, only W^{out} should be modified during training. The topology of the hidden layer and other weight matrixes remain unchanged. Here we use least-squares algorithm to train an ESN.

2.2 Hurst Exponent

Before training, an initial transient should be dismissed first so that after the transient time the internal network state is determined by the preceding input history [5]. Here we apply the Hurst exponent [9] to decide initial transient.

The Hurst exponent provides a measure for the long-term memory and fractality of a time series [10]. If its value H is between 0.5 and 1, it indicates a persistent series that the history influences the future. The closer H is to 1, the stronger the impact is. Hurst exponent can be computed according to rescaled range analysis (R/S analysis) [10].

We extract a set of close price sub-series from training samples, which all end at the last day but begin from different points, and calculate their Hurst exponent. The sub sample whose length is larger than the testing size and whose Hurst exponent is the closest to 1 is the final training set.

2.3 Data Preparation

Because in short-term prediction, the influence of macroeconomic environment and a company's financial conditions is negligible, we only consider raw prices and technical indicators in our system. We tried various kinds of combination to find the optimal one. All the input data are listed as follows and linearly normalized to $[-1, 1]$.

- High: the maximum price of a stock ticker during the intra-day trading.
- Low: the minimum price of a stock ticker during the intra-day trading.
- Open: the first price of a stock ticker during the intra-day trading.
- Close: the final price of a stock ticker during the intra-day trading.
- 5-day High: the highest High Price during the past 5 days.
- 5-day Close Moving Average: the average of Close in the past 5 days.

3 Experiments and Results

In our experiments, the stock information between Dec., 6th, 2001 and Nov., 25th, 2005 (the first 1000 days for training and the last 100 days for testing) of 491 stocks in S&P 500 is adopted. We also applied BPNN, Elman network and RBFNN (using Matlab Neural Network Toolbox) on the same task for comparison. Note that their topology and parameters are chose by manual experiments to guarantee satisfying results on most stocks. Average percent error (APE, See Equation 3) is the criteria to judge all networks.

$$APE = |y_t - \hat{y}_t| / \hat{y}_t \times 100\% \tag{3}$$

where y_t is the actual output and \hat{y}_t is the desired one. Table 1 lists some prediction results that are randomly chosen.

Table 1. Stock Prediction Results (Next-Day Close Price Prediction)

Stock	ESN	BPNN	Elman	RBF
ACE	1.15%	3.87%	3.75%	3.09%
AHC	1.69%	8.77%	10.06%	9.15%
AMD	1.94%	2.66%	2.54%	2.99%
BBT	0.84%	0.87%	0.83%	0.91%
CIEN	3.81%	3.55%	3.68%	2.48%
GD	0.65%	2.07%	3.25%	1.32%
JCP	1.58%	2.04%	2.01%	2.26%
KMG	1.51%	5.18%	6.37%	8.25%
NBR	1.78%	6.55%	6.11%	4.27%
NSC	1.12%	2.66%	2.31%	2.48%
PSA	0.97%	2.31%	2.70%	1.75%
RHI	1.61%	6.41%	6.28%	9.99%
SFA	1.53%	3.47%	3.22%	3.21%
USB	0.79%	0.75%	0.74%	0.79%

Statistically, ESN performs the best in 57.03% cases, especially much better than others in some cases such as AHC, KMG, NBR and RHI. BPNN performs the best in 14.87% cases; Elman performs the best in 20.16% cases and RBFNN performs the best in 7.94% cases. Moreover, in 138 out of 211 stocks where ESN does not perform the best, it gives results very close to the optimal one (the gap is smaller than 0.5%).

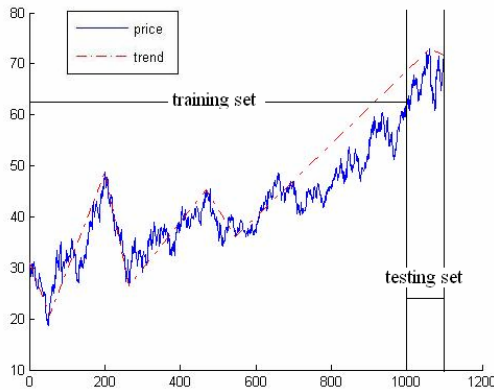


Fig. 2. The close price of NBR between Dec., 6th, 2001 and Nov. 25th, 2005

Only in a few cases (16/491), ESN gives very bad results. But by adjusting the parameters such as λ_{\max} , W^{back} and ν , the accuracy can be improved.

Qualitative observation finds that in many cases, if second half of the training samples approximately share the same trend with the first half of testing set (See Figure 2), ESN tends to outperform other networks. It coincides with the short-term memory ability of ESN.

Table 2 lists the average elapse time of every neural network to train and predict one stock on a Core 2 CPU T5500 1.0G computer. Obviously, it takes the most time for ESN to predict stock prices. But further experiments find that the process to calculate the Hurst exponent spends most of the time.

Table 2. Average Performance of Neural Networks

ESN (with Hurst exponent)	ESN (fixed transient)	BPNN	Elman	RBF
55.05s	6.86s	17.68s	23.91s	3.27s

4 Conclusions

In this paper, we applied Echo State Network in short-term stock data mining and compare its predictive accuracy with BPNN, Elman network and RBFNN. The Hurst exponent is used to guide transient selection before training an ESN. The experiments demonstrate that ESN is an effective model to predict stock time-series and outperform other conventional neural networks in most cases.

However, ESN is a young discrete model for chaotic time-series mining that needs further studies. First, which type of data ESN is suitable for is worth study. Our preliminary observation indicates that the price trend may influence prediction accuracy, but further exploration is required. Second, finding an optimal architecture and transfer functions with genetic algorithms (GAs) for each stock is under consideration. Finally, whether ESN has the ability of long-term stock data mining is still of interest.

References

1. Kim, H.-j., Shin, K.-s.: A hybrid approach based on neural networks and genetic algorithms for detecting temporal patterns in stock markets. *Applied Soft Computing* 7(2), 569–576 (2007)
2. Tan, T.Z., Quek, C., Geok See, N.G.: Biological brain-inspired genetic complementary learning for stock market and bank failure prediction. *Computational Intelligence* 23(2), 236–261 (2007)
3. Jaeger, H., Haas, H.: Harnessing Nonlinearity: Predicting Chaotic Systems and Saving Energy in Wireless Communications. *Science* 3 (2004)
4. Jaeger, H.: Short term memory in echo state networks GMD-Report 152. GMD-German National Research Institute for Computer Science (2002)

5. Ishii, K., van der Zant, T., Becanovic, V., Ploger, P.: Optimization of Parameters of Echo State Network and Its Application to Underwater Robot. In: SICE Annual Conference in Sapporo, vol. 3, pp. 2800–2805 (2004)
6. Jaeger, H., Lukosevicius, M., Popovici, D.: Optimization and applications of echo State networks with leaky integrator neurons. *Neural Networks* 20, 335–352 (2007)
7. Skowronski, M.D., Harris, J.G.: Automatic speech recognition using a predictive echo state network classifier. *Neural Networks* 20, 414–423 (2007)
8. Jaeger, H.: The Echo State Approach to Analyzing and Training Recurrent Neural Networks GMD-German National Research Institute for Computer Science, vol. GMD Report 148 (2001)
9. Hurst, H.E.: Long-term storage of reservoirs: an experimental study. *Trans Amer. Soc. Civil Engi.* 116, 770–799 (1951)
10. Qian, B., Rasheed, K.: Stock market prediction with multiple classifiers. *Applied Intelligence* 26(1), 25–33 (2007)

Text Categorization of Multilingual Web Pages in Specific Domain

Jicheng Liu and Chunyan Liang

North China Electric Power University
102206 Beijing, China
rjrj1999@gmail.com

Abstract. Compared to the traditional text categorization, automated categorization for domain-specific web pages poses new research challenges because of the noisy and diverse nature of the pages and the fine and complex category structure. For multilingual web pages, it also needs to be considered that how to extract the terms of different languages exactly. Using a dataset of hybrid Chinese-English chemical web pages, a new dictionary-based multilingual text categorization approach is proposed in this paper to try to classify the pages into a hierarchical topic structure more accurately. By using an automatic encoding detection and integration method, the approach can properly recognize and integrate the web page encodings. This makes the feature extraction more precise for the multilingual pages. The approach can also intensify the domain concepts in the web pages based on a chemistry dictionary. The experimental results show that the proposed approach has the better performance than the traditional categorization method when classifying the multilingual web pages in specific domain.

Keywords: Text categorization, Encoding detection and integration, Domain-related dictionary.

1 Introduction

In order to help users to find relevant information in Internet accurately and quickly, Text Categorization (TC) might be desirable to automatically classify web pages according to their topics and to show them in a category interface [1]. A wide range of statistical classification and machine learning techniques [2, 3, 4] have been applied to TC, including Rocchio classifiers, nearest neighbor classifiers, decision trees, Bayesian classifiers, support vector machines (SVMs), and so on. According to the research [3, 5, 6] comparing the performance of these techniques, the k -nearest neighbor (kNN) classifier is a learning method that is simple to implement, easy to scale up, relative robust and has good performance. In this paper, a modified kNN classifier is applied to classify the web pages of specific domain into a hierarchical topic structure.

Compared to the traditional text categorization, text categorization for domain-specific web pages poses new research challenges because of the diversity and

noisy nature of the web pages[2] and the fine and complex category structure. A dictionary is constructed to represent the domain knowledge that can be used to intensify the domain concept in the pages, and a new text categorization approach based on the dictionary is proposed to attempt to classify the domain-specific web pages more efficiently.

Amounts of Chinese web pages in Internet include English or other language words. For the hybrid multilingual web pages, it is essential to recognize the different language characters and extract the information accurately from the pages with diverse language encoding schemes. An automatic encoding detection and integration method is applied in the dictionary-based text categorization method to extract Chinese and English text information precisely from the hybrid bilingual Chinese-English pages.

2 Text Categorization

In this paper, we use the kNN classifier to classify the web pages for its robust and effective performance [5, 6]. Before classification, Feature extraction should be done to select the informative terms from the documents to form the vector space that will be used in the classifier. In this paper, both training and test documents are converted from the original format to the final vectors through the following steps. First, useless tags and stop words [7] are removed from the documents and word stemming is performed using the Porter stemmer. Then, term weights are computed [7] to represent the documents as term vectors. Since not every document contains all terms, the feature vector space is usually very large and sparse. Both feature selection and re-parameterization [8] are performed to reduce the original feature dimensionality.

Based on the vector space, kNN classifier is used to perform the multi-label classification. Given an test document, the kNN classifier ranks its nearest neighbours among training documents according to the corresponding cosine value, and uses the categories of the k top-ranking neighbours to predict the categories of the test. The cosine value of each neighbour is used as the weight of its categories, and the sum of category weights over the k nearest neighbours are used for category ranking. Those categories with a rank score higher than a threshold value are finally assigned to the test.

3 Dictionary-Based Multilingual Text Categorization

In spite of the fact that many Internet standard protocols designate Unicode as the default encoding, there still exist many encoding schemes on the Internet for a variety of reasons [9]. Without the right encoding information, the web pages are sometimes regarded as “garbage” or “unreadable” text during feature extraction by the classifier. To extract the features more exactly from the bilingual Chinese-English web pages, an automatic encoding detection and integration method is proposed to get the encodings of the different pages and integrate multiple encodings into a uniform encoding.

Compared with English characters, it is tougher to recognize Chinese characters because the commonly used encodings, such as GB2312, Big5 and UTF-8, have

different code-point ranges for them. To detect the language encoding of web pages, we try to get the encoding by extracting the charset information from the Meta HTML tags of the input pages. For the pages without explicit charset declaration, we apply a Character Distribution Method [9], which identifies the encoding based on the code point distribution statistics in the pages. Since different encoding schemes may have different code points for the same character, it is necessary to integrate them into a uniform encoding to make the pages with different encodings but same meaning have the same represented vector in the classifier. In this paper, we use GB2312 as the uniform encoding. GB2312 covers both the English and commonly used Chinese characters and it is easy to separate the bi-byte Chinese characters and single-byte English characters. After encoding detection and integration, all the Chinese-English pages with different encodings are integrated and can be represented accurately in the classifier.

For the domain-specific web pages, the semantic similarity between them brings more difficulties when retrieving the k nearest neighbors by identifying the differences among the documents to rank them, which makes the kNN classifier perform poor as the following experiments show. This motivates us to build a domain-related dictionary to represent the domain knowledge which could be helpful to extract the domain information more exactly from the web pages. Then based on the dictionary, a new text categorization approach is proposed.

In order to classify the chemical web pages more accurately in this paper, a machine-readable chemistry dictionary (ChemDict) is built. ChemDict has total 172,786 Chinese terms and 173,895 English terms. ChemDict includes large amounts of phrase Chinese and English terms consisting of more than one word. Longest substring matching algorithm is adopted to match the dictionary terms. Then these matched terms are inserted into the documents where they appear when indexing to intensify the chemical concepts and improve the semantic presentation.

In the dictionary-based multilingual text categorization system, encoding detection and integration described is first performed on the web pages, and then longest substring matching algorithm is run to extract dictionary terms for document expansion. After that, feature extraction as presented in section 2 is performed to produce the final vectors. In this process, the stop word list consists of 548 English terms and 1,068 Chinese terms and the stemming step is ignored for Chinese terms. Based on the training vector space, the kNN classifier finds the k nearest neighbors of the test document, and the relevant categories of the test are finally obtained.

4 Experiments

This section describes the experiments for comparing the dictionary-based multilingual text categorization approach with the traditional method in detail. Results are also provided and discussed.

To test the text categorization method on the real domain-related web pages, we constructed a labeled dataset from the chemical resources collected by ChIN [10] editors. ChIN is a comprehensive Chinese-English chemistry resource directory and

uses a hierarchical chemical categorization scheme adopted by Natural Science Foundation of China. It is a three level hierarchical topic structure and has total 341 categories. Each chemical resource in ChIN is manually classified into one or more categories of this structure. We used a subset of the ChIN resources to construct the ChIN-Page dataset. This dataset includes both Chinese and English chemical web pages collected based on the ChIN resource links from the Internet by a real-time crawler.

To compare the text categorization performance on the chemical web pages with that on the normal documents, we also use the popular Reuters-21578 [11] collection as the second dataset.

The category distributions in the datasets are shown in Table 1. Close observation will find that the datasets are all uneven or skewed multi-label datasets, while the category complexity and the number of categories per document of ChIN-Page are much larger than Reuters. Consequently, the categorization is more difficult to learn as the following experimental results show. Table 2 shows the document distributions in the datasets, where Reuters use the “ModApte” split [11] and ChIN-Page dataset is split according to the ChIN indexed date of each page. The documents that are unlabeled or have no text besides the tags are removed from both datasets. The distributions of Chinese documents in Chin-Page are put in parentheses.

Table 1. Category distributions in the datasets

Dataset	Hierarchical Level	Total Categories	Categories	Categories	Number of Categories per Document
			Having More than One Documents	Having More than 20 Documents	
Reuters	1	135	120	57	1.2
ChIN-Page	3	341	257	71	3.1

Table 2. Document distributions in the datasets

Dataset	Total Documents	Documents	Documents
		in Training Set	in Test Set
Reuters	9805	7063	2742
ChIN-Page	2337(295)	1635(255)	702(40)

For evaluating the average performance of a classifier over multiple categories, we followed the traditional macro-averaging and micro-averaging method [6]. Micro-averaged scores tend to be dominated by the performance of the system on common categories, while macro-averaged scores tend to be dominated by the performance on rare categories if the majority of categories in the task are rare [2]. For the uneven category distributions (Table 1.) in our tasks, providing both types of evaluation scores gives a clearer picture than considering either type alone. In the

following experiments, the micro-averaged F_1 and macro-averaged F_1 are used to evaluate the classifier average performance.

It is obvious that the classification performance is highly dependent on the dataset used. In the experiments, we first test the traditional text categorization method on the Reuters and the chemistry-related ChIN-Page dataset. Then the proposed dictionary-based multilingual text categorization approach is used to observe its effect on classification performance.

4.1 Effect of Dataset on Classification Performance

The traditional text categorization method described in section 2 is separately performed on the Reuters and ChIN-Page dataset. To show the effect of the category structure on the classification performance, we also classify the ChIN-Page web pages using different categorization schemes, that is only using the top-level categories (1-level), the top and the second level categories (1.2-level) and all the categories (all-level) in the 3-level hierarchical category structure. The experimental results are listed in the Table 3, where bold font marks the best performance of each column. The second column shows the actual total category number that the test documents are assigned in the experiments. For the top level categorization scheme of ChIN-Page, the performance of 10 most frequent categories is substituted by that of all the 7 categories.

The column-wise comparison in Table 3 shows that the performances on Reuters are often much better than that on ChIN-Page for Reuters has fewer noisy data and coarser classification scheme than ChIN-Page, while the macro-averaged F_1 of all the categories on Reuters is worse than that on ChIN-Page with the top-level scheme because Reuters has more and skewed category distributions. For the ChIN-Page, the more fine and complex categorization scheme induces the more inferior performance.

Table 3. Classification performances of the traditional text categorization on the datasets

Dataset	Category Number	All Categories		10 Most Frequent Categories	
		MicroAvg F_1	MacroAvg F_1	MicroAvg F_1	MacroAvg F_1
Reuters	94	0.7974	0.3636	0.8908	0.8058
	7 (1-level)	0.5709	0.5953	0.5709	0.5953
ChIN-Page	72 (1.2-level)	0.4418	0.2424	0.5479	0.5509
	206 (all-level)	0.3871	0.0526	0.5258	0.5074

From the row-wise comparison in Table 3, we can observe that the macro-averaged F_1 is often lower than the micro-averaged F_1 for the skewed category distribution in each dataset. The results also show that the averaged performances on the 10 most frequent categories are always superior to that on all the categories.

4.2 Effect of Dictionary-Based Multilingual Text Categorization on Classification Performance

The dictionary-based multilingual text categorization approach described in section 3 is performed on ChIN-Page with the 3-level hierarchical category structure to extract the chemistry information more exactly. The encoding detection and integration method without dictionary is also tested in the traditional text categorization to observe its effect. The results are shown in Table 4, where the second column shows the feature number in the training documents after feature extraction.

Table 4. Classification performances of the different text categorization methods on ChIN-Page dataset

Text Categorization Method	Feature Number	All Categories		10 Most Frequent Categories	
		MicroAvg F1	MacroAvg F1	MicroAvg F1	MacroAvg F1
Traditional	11,517	0.3871	0.0526	0.5258	0.5074
Encoding Detection and Integration	13,739	0.4166	0.0899	0.5598	0.5525
Dictionary-based multilingual	18,054	0.428	0.0904	0.5715	0.5502

From the results, we can see that the proposed encoding detection and integration method can extract precise and uniform Chinese features from web pages, and can improve the classification performance. We can also find that the dictionary-based multilingual text categorization approach can further improve the micro-averaging performance while not notably influence the macro-averaging performance. The reason may be that the classifier can't extract enough chemical terms to properly classify the test documents in the small categories that have scarce training documents.

From the performance comparison of the dictionary-based multilingual text categorization with the traditional method, we can observe that the proposed approach can notably improve the classification performance on the ChIN-Page web pages. For the micro-averaged F1 on all the categories, we can obtain about 11% improvement over the traditional method.

5 Conclusions

Text Categorization for domain-specific web pages poses new research challenges for the noisy and diverse nature of the pages and the fine and complex category structure of the specific domain. The proposed dictionary-based multilingual text categorization approach in this paper can effectively improve the categorization performance on the real chemistry-related web pages in the ChIN-Page dataset.

Future research is needed on the issues, such as strategy optimization of expanding a document by matched dictionary terms and utilizing the hyperlink and other web page characteristics to improve the classification performance for domain-specific web pages.

References

- [1] Dumais, S.T., Cutrell, E., Chen, H.: Optimizing search by showing results in context. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 277–284. ACM Press, Seattle, Washington, United States (2001)
- [2] Yang, Y., Slattery, S., Ghani, R.: A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems* 18, 219–241 (2002)
- [3] Lewis, D.D., Ringuette, M.: A comparison of two learning algorithms for text categorization. In: Proceedings of Third Annual Symposium on Document Analysis and Information Retrieval, Univ. of Nevada, Las Vegas, NV, pp. 81–93 (1994)
- [4] Joachims, T.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
- [5] Aas, K., Eikvil, L.: Text categorisation: a survey, Technical report, Norwegian Computing Center (1999)
- [6] Yang, Y.: An evaluation of statistical approaches to text categorization. *Information Retrieval* 1, 69–90 (1999)
- [7] Wang, Z.: Improving on latent semantic indexing for chemistry portal, Thesis, Institute of Process Engineering, Chinese Academy of Sciences (2003)
- [8] Berry, M.W., Dumais, S.T., O’Brien, G.W.: Using linear algebra for intelligent information retrieval. *SIAM Review* 37, 573–595 (1995)
- [9] Li, S., Momoi, K.: A Composite Approach to Language/Encoding Detection. In: Nineteenth International Unicode Conference, San Jose, California (2001)
- [10] ChIN: the chemical information network, Institute of Process Engineering, Chinese Academy of Sciences (2007), <http://chin.csd1.ac.cn/>
- [11] Lewis, D.D.: Reuters-21578 (2007), <http://www.daviddlewis.com/resources/testcollections/reuters21578/>

Efficient Joint Clustering Algorithms in Optimization and Geography Domains

Chia-Hao Lo and Wen-Chih Peng

Department of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, ROC
{fcamel,wcpeng}@gmail.com

Abstract. Prior works have elaborated on the problem of joint clustering in the optimization and geography domains. However, prior works neither clearly specify the connected constraint in the geography domain nor propose efficient algorithms. In this paper, we formulate the joint clustering problem in which a connected constraint and the number of clusters should be specified. We propose an algorithm K-means with Local Search (abbreviated as KLS) to solve the joint clustering problem with the connected constraint. Experimental results show that KLS can find correct clusters efficiently.

1 Introduction

A joint clustering problem over the geography domain and the optimization domain is that given a set of data objects with their attributes in both the geography domain and the optimization domain, we should partition objects into several groups such that objects in the same group are connected in the geography domain while minimizing the dissimilarity of the data objects in the optimization domain.

In this paper, we formulate a joint clustering problem with the connected constraint. Then, an algorithm KLS (standing for K-means with Local Search) is proposed. KLS consists of three phases: the transformation phase, the coarse clustering phase and the fine clustering phase. First, given the connected constraint required and the attributes of objects in the geography domain, grid-cells data structure is used to efficiently derive ConGraph (standing for CONnected Graph), where each vertex is a data object and an edge exists between two objects if their distance in the geography domain is within a given threshold. In light of ConGraph, we exploit the concept of K-means and local search to coarsely cluster objects into several groups. Based on the clustering results derived, we could further fine tune clusters to minimize the dissimilarity in the optimization domain. Our experimental evaluation demonstrates that algorithm KLS is indeed able to efficiently derive cluster results.

The joint clustering problem were proposed in [3,6,4]. Moreover, the clustering problem with constraints are addressed in [7,11,2]. Although prior works have

elaborated on the joint clustering problem, the connected constraint in the geography domain is not clearly defined, let alone proposing efficient algorithms for large scale of objects given. These features distinguish our study from others.

The rest of the paper is organized as follows. Preliminaries are given in Section 2. The proposed algorithm is presented in Section 3. Performance evaluation is conducted in Section 4. This paper concludes with Section 5.

2 Preliminaries

Objects considered in this paper have two domains of attributes. The two domains are the optimization domain and the geography domain. To facilitate the presentation of this paper, an object i is denoted as o_i . The corresponding set of attributes of o_i in the optimization domain is expressed by S_i , the j th attribute in S_i is represented as s_i^j , and the dimension of the optimization domain is d_S . Similarly, the definitions of L_i , l_i^j and d_L are respected to the geography domain. The Euclidean distance between two objects is used as the dissimilarity measurement. For two objects o_i and o_j , the distance in the optimization domain is formulated as: $dist_{opt}(o_i, o_j) = \sqrt{\sum_{k=1}^{d_S} (s_i^k - s_j^k)^2}$. Similarly, the one in the geography domain is denoted as $dist_{geo}(o_i, o_j)$.

Let a cluster C_j have a set of objects $(o_1, o_2, \dots, o_{|C_j|})$. The cost of cluster C_j in the optimization domain is formulated as:

$$g(C_j) = \frac{1}{|C_j|} \sum_{i=1}^{|C_j|} dist_{opt}(o_i, cen_j), \text{ where } cen_j \text{ is the centroid of } C_j.$$

Consequently, given a set of clusters $SC=(C_1, C_2, \dots, C_k)$, the average cost of SC is defined as: $f(SC) = \sum_{i=1}^k \frac{|C_i|}{n} g(C_i)$, where $n = \sum_{i=1}^k |C_i|$. The constraint in the geography domain is used to cluster objects such that their distance in the geography domain is within a threshold required such that objects in the same cluster are *connected*. The definition of the connected constraint is defined as:

Definition 1. (*Connected constraint on cluster*) Given a clusters C_t , where $|C_t| > 1$, and a threshold r , $\forall o_i, o_j \in C_t \wedge o_i \neq o_j, dist_{geo}(o_i, o_j) \leq r$ or there is a sequence of objects $o_{u1}, o_{u2}, \dots, o_{un} \in C_t$ such that $dist_{geo}(o_i, o_{u1}) \leq r, \dots$, and $dist_{geo}(o_{un}, o_j) \leq r$. C_t fits the constraint inherently when $|C_t| = 1$.

From the definitions above, the problem in this paper is that given the number of clusters k , a distance threshold r , n objects o_1, o_2, \dots, o_n with their attributes in the optimization and geography domain, the goal is to derive a set of clusters $SC = (C_1, C_2, \dots, C_k)$ such that (1) each object o_i belongs to only one cluster C_j , (2) objects in the same cluster are connected, and (3) the average cost (i.e., $f(SC)$) is minimized.

3 Algorithm KLS: K-Means with Local Search

We propose algorithm KLS consisting of three phases: the transformation phase, the coarse clustering phase, and the fine clustering phase. In the transformation

phase, ConGraph (standing for CONnected Graph) is derived for efficiently verifying the connected constraint. Then, in the coarse clustering phase, rough clusters are efficiently derived via ConGraph. The number of clusters is greater than or equal to k and the clusters may lose some qualities due to the efficiency consideration. Finally, the rough clusters are iteratively merged according to $f(SC)$ until the number of clusters is k .

3.1 Transformation Phase

In this phase, given a set of objects, the goal is to derive ConGraph that captures the connected features among objects in the geography domain. The definition of ConGraph is as follows:

Definition 2. (*ConGraph*) Given $O = \{o_1, \dots, o_n\}$ and a threshold r , a *ConGraph* is a graph $G = (V, E)$, where vertex v_i is object o_i and an edge $e(v_i, v_j)$ between v_i and v_j exists if $dist_{geo}(o_i, o_j) \leq r$.

We divide the geography domain into equaled sizes of cells. Objects are hashed into cells according to their attributes in the geography domain. Through cells, given an object o_i , we are able to quickly find out possible objects nearby o_i . Since the threshold of the connected constraint is set to r , the length of a cell size can be set to $2r$ such that only 2^{d_L} neighbor cells are required to explore when retrieving the neighbor nodes of a vertex. Therefore, the generation of ConGraph is efficient in that only a limited amount of objects are accessed.

3.2 Coarse Clustering Phase

Same as in K-means, we first select k vertices as initial centroids. Adapting the concept of local search [5], these vertices are used as the represented objects for their clusters. Then, those neighbors of these represented objects are extracted. The distance of these neighbors to the corresponding centroid in the optimization domain are calculated. Then, only the neighbor with the smallest distance value will be selected into the nearest cluster and the centroid of the corresponding cluster will be updated. Moreover, the represented object for the corresponding cluster is replaced by the new neighbor. The above procedure will be repeated until there is no unclustered neighbor. After this procedure, the unclustered objects are assigned to the nearest cluster if the connected constraint is not violated.

3.3 Fine Clustering Phase

Before explaining the fine clustering method, the connectivity of two clusters is defined as follows:

Definition 3. (*Connected constraint among clusters*) Two clusters C_i and C_j are connected if $\exists o_t \in C_i$ and $\exists o_u \in C_j$ such that $dist_{geo}(o_t, o_u) \leq r$.

The goal of this phase is to merge clusters until the number of clusters is k . To minimize the average cost, we adapt the agglomerative hierarchical clustering with the mean distance. The fine clustering phase is to recursively merge two connected clusters with the smallest distance between their centroids until the number of clusters is k .

4 Performance Study

We generate synthetic data with attributes in the two domains. The dimensions of the optimization and geography domain are three and two, respectively. Our generator requires parameters k and r , where k indicates how many clusters should be generated, and r is the threshold of the connected constraint. The shapes of objects are composed of mainly horizontal and vertical lines in the geography domain. After generating attributes in the geography domain, attribute in the optimization domain are then determined at random similar to [3]. We use $r - 1$ as the parameter of the data generator in order to make objects closer. The other parameter k , the number of clusters, is different among test scenarios.

For comparison purposes, we implement one naive algorithm Connected K-means (abbreviated as CK-means) which has three phases as KLS does. The only difference is that in the coarse clustering phase of CK-mean, objects are first partitioned by K-means with the attributes in the optimized domain only. After clusters OC are derived by K-means, we generate a new graph $G' = (V, E')$, where $E' = \{e(v_i, v_j) \mid e \in E \wedge v_i, v_j \in C_t, C_t \in OC\}$. We exploit BFS to traverse G' and get connected subgraphs $G_1, G_2, \dots, G_{k'}$. Then each connected subgraph is an equivalent cluster which fits the connected constraint. Therefore, $k' \geq k$ clusters are found.

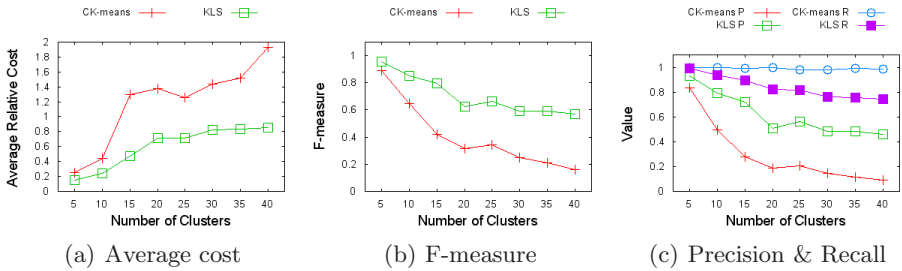


Fig. 1. Overall results between CK-means and KLS

Figure 1 shows the results of the experiments with our algorithms. We use $2k$ as the number of seeds in KLS instead of k . This small modification increases the precision of clusters found by KLS since it is hard to choose the correct seeds even a useful heuristic method is used. Then, the fine clustering phase iteratively merges clusters until the number of clusters is k .

According to Fig. 1(a) and 1(b), CK-means performs worse than KLS. Specially, when the number of clusters increases, CK-means performs much worse

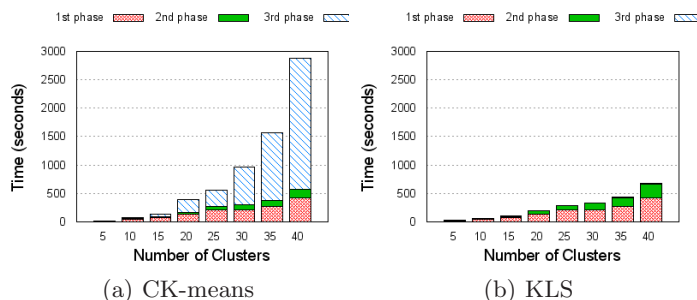


Fig. 2. Performance comparisons of CK-means and KLS

than KLS. Fig. 1(c) shows the corresponding value of the overall precision and recall that show CK-means performs worse due to its low precision.

As shown in Fig. 2, CK-means is slower than KLS. In Fig. 2(a), the execution time of the fine clustering phase dominates the overall time of CK-means since CK-means generated too many clusters in the coarse clustering phase. On the other hand, in Fig. 2(b), the transformation phase dominates the execution time.

5 Conclusion

In this paper, we formulated the joint clustering problem in which a connected constraint and the number of clusters should be specified. We propose algorithm KLS that consists of three phases: the transformation phase, the coarse clustering phase, and the fine clustering phase. In the transformation phase, we only consider the connected constraint and then derive ConGraph. Thus, in the coarse clustering phase, by exploring local search in ConGraph, rough cluster results are derived. In the fine clustering phase, these clusters are able to further merged to optimize the objective of the joint clustering. Experimental results show the effectiveness and efficiency of our proposed algorithm.

Acknowledgement

Wen-Chih Peng was supported in part by the National Science Council, Project No. NSC 95-2211-E-009-61-MY3 and NSC 97-2623-7-036-001-D, Taiwan, Republic of China.

References

1. Basu, S., Davidson, I.: Clustering with constraints: Theory and practice. In: ACM SIGKDD tutorial (2006)
2. Davidson, I., Ravi, S.S.: Clustering with constraints: Feasibility issues and the k-means algorithm. In: SDM (2005)

3. Lin, C.R., Liu, K.H., Chen, M.S.: Dual clustering: Integrating data clustering over optimization and constraint domains. *IEEE Trans. on Knowledge and Data Engineering* 17, 628–637 (2005)
4. Moser, F., Ge, R., Ester, M.: Joint cluster analysis of attribute and relationship data without-a-priori specification of the number of clusters. In: *ACM SIGKDD (2007)*
5. Russell, S.J., Norvig, P.: *Artificial Intelligence: A Modern Approach*, 2nd edn. Prentice Hall, Englewood Cliffs (2002)
6. Tai, C.H., Dai, B.R., Chen, M.S.: Incremental clustering in geography and optimization spaces. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) *PAKDD 2007. LNCS (LNAI)*, vol. 4426, pp. 272–283. Springer, Heidelberg (2007)
7. Wagstaff, K., Cardie, C.: Clustering with instance-level constraints. In: *ICML (2000)*

Active Learning with Misclassification Sampling Using Diverse Ensembles Enhanced by Unlabeled Instances

Jun Long, Jianping Yin, En Zhu, and Wentao Zhao

National University of Defense Technology, Changsha, Hunan 410073, China

Abstract. Active learners can significantly reduce the number of labeled training instances to learn a classification function by actively selecting only the most informative instances for labeling. Most existing methods try to select the instances which could halve the version space size after each sampling. In contrast to them, we try to reduce the volume of the version space more than half. Therefore, a sampling criterion of misclassification is presented. Furthermore, in each iteration of active learning, a strong classifier was introduced to estimate the target function for evaluation of the misclassification degree of an instance. We use a modified popular ensemble learning method DECORATE as the strong classifier which was enhanced by the unlabeled instances with high certainty by the current base classifier. The experiments show that the proposed method outperforms the traditional sampling methods on most selected datasets.

1 Introduction

The standard setting of supervised learning assumes that a previously labeled set of instances is available. However, in a large number of real world applications, obtaining labeled instances may be expensive or time-consuming. Therefore, reducing the number of labeled instances that are necessary to learn a classification function becomes important. Active learning methods [1] allow classifiers choose the most informative instances and ask the experts to label them. Thus the burden of labeling large number of instances could be alleviated.

The whole process of the pool-based active learning can be described as follows. Initially, the active learner has access to a pool of unlabeled instances and owns a set of labeled instances. Then, the active learner trains a base classifier on the set of labeled instances. Afterwards, an instance is sampled for labeling according to a certain criterion and is added into the labeled set. Then the active learner trains a new base classifier on the updated labeled set. The whole process runs repeatedly until the error rating of the current base classifier is below a preset value.

Depending on the criterion used to select instances for labeling, the current research falls under several categories: uncertainty reduction, expected-error minimization and version space reduction [2]. The uncertainty reduction approach

[13] selects the instances on which the current classifier has the least certainty of prediction. The expected-error minimization approach [4] samples the instances that minimize the future expected error rate on the test set. The version space reduction approach [5,6], including QBC [5], QBag [7], QBoost [7] and Active DECORATE [8], tries to select the instances that can reduce the volume of version space by half which based on the idea of binary searching. Query-by-Committee is a representative method of this approach that constructs a committee consisting of randomly selected hypotheses from the version space and selects the instances on which the disagreement within the committee is the greatest.

Choosing an efficient criterion for instance selection is the most important step in active learning. Most existing methods use the idea of binary searching in version space reduction process [5,7]. The binary searching idea assumes that all hypotheses in the version space have equal probability to be the target function. However, the assumption can not hold in most tasks.

We aim to accelerate the version space reduction process more than what binary searching does. We propose a sampling criterion which tries to keep only the most accurate hypotheses in the version space when sampling. Thus we propose a sampling method MSDEEUI (Misclassification Sampling Using Diverse Ensembles Enhanced by Unlabeled Instances) that tends to select the instances with the largest prediction difference between a strong classifier and the current base classifier. In this paper, the strong classifier is generated by the ensemble method DECORATE trained on the current labeled set and enhanced by the unlabeled instances with high certainty predicted by the current base classifier. The experiments show that the proposed method outperforms the traditional sampling methods on most selected datasets.

The rest of the paper is organized as follows. Section 2 introduces the basic notations. Section 3 presents the proposed active learning method MSDEEUI in details. Section 4 shows the experimental results of the MSDEEUI method as well as other methods on selected data sets. Section 5 draws the conclusion.

2 Preliminary

2.1 Notations

The instance space X is a nonempty set containing several instances. Each instance x_i is a feature vector. Let $Y = \{y_1, y_2, \dots, y_l\}$ be the set of possible labels. For simplification, we focus on 2-value classification problems in the paper. Thus $Y = \{0, 1\}$. The target function c to be learned is a function $c : X \rightarrow Y$ that classifies any $x \in X$ as a member of Y . The notion $\langle x, c(x) \rangle$ denotes a labeled instance, $\langle x, ? \rangle$ denotes an unlabeled instance where $? \in Y$ and D denotes a set of labeled instances for training. The hypothesis space H is a nonempty set containing functions that map from X to Y . Providing a set of labeled instances, the task of learning is searching a function f such that $\forall x \in X, f(x) = c(x)$ in the hypothesis space H . The version space VS_{HD} denotes the largest subset of hypotheses in H that satisfies $\forall h \in VS_{HD}, \forall x \in D, h(x) = c(x)$.

3 The Proposed MSDEEUI Method

3.1 Efficiency of Instances for Version Space Reduction

To visualize the process of version space reduction, we define hypothesis-instance matrix HIM_{mn} as a binary matrix whose rows are indexed by the hypotheses h_1, h_2, \dots, h_n from the hypothesis space H and whose columns are indexed by the instances x_1, x_2, \dots, x_m from the instance space X , and whose (i, j) entry is 1 if $h_i(x_j) = c(x_j)$ and otherwise 0. We define an operation $TRN(x_j)$ on the hypothesis-instance matrix HIM_{mn} as follows: when the operation $TRN(x_j)$ is executed, HIM_{mn} removes all the i th rows that satisfy the entry (i, j) being 0.

Then the process of active learning can be viewed as that providing a sequence of instances x_1, x_2, \dots, x_t , HIM_{mn} execute the operation of $TRN(x_1), TRN(x_2), \dots, TRN(x_t)$ one by one. Furthermore, the current version space can be denoted by the rows of the current HIM_{mn} (after executed several TRN operations).

When HIM_{mn} executes $TRN(x_i)$, many rows would be removed. The more rows are removed, the faster the version space size is reduced. Therefore we define the efficiency of $TRN(x_i)$ as

$$E_{x_i} = \frac{\#\{h_j | h_j \in VS_{HD}, h_j(x_i) \neq c(x_i)\}}{\#VS_{HD}} \quad (1)$$

E_{x_i} is the proportion of the hypotheses which misclassify x_i to the version space VS_{HD} and can also be viewed as the probability of the instance x_i being misclassified by the version space.

E_{x_i} indicates the ability of x_i to distinguish the hypotheses with high accuracy from the hypotheses with low accuracy. An instance x with high E_x implies that x is hard to be classified correctly and most of the hypotheses may misclassify x . Therefore if we choose the instance x with the highest E_x for labeling, the version space will decrease to several the most accurate hypotheses. Therefore E_{x_i} is a more efficient criterion than the criterion based on binary searching (e.g. uncertain sampling, QBC etc.).

3.2 Constructing the Strong Classifier Based on Modified DECORATE

The criterion E_{x_i} can not be calculated directly. Then we assume the base classifier could represent the whole version space. Thus E_{x_i} can be rewritten as: $E_{x_i} = P(h_D(x_i) \neq c(x_i))$ where h_D is the current base classifier trained on the labeled set. Then E_{x_i} is the probability of the unequal decisions on x_i made by h_D and c .

The target function c is unknown yet. Then we estimate it by constructing a strong classifier by using ensemble methods and enhance the strong classifier by unlabeled instances on which the current base classifier has high certainty.

An ensemble of classifiers is a set of classifiers whose individual decisions are combined to classify new instances [9]. As Hansen and Salamon had pointed out, an ensemble can be more accurate than its component classifiers when each

component classifier outputs independently and has an accuracy over $1/2$ [10]. An important property of a good ensemble for committee-based active learning is diversity [8].

We select DECORATE [11] to generate the ensemble. The DECORATE method generate the ensemble iteratively, learning one new classifier in each iteration and adding it to the current ensemble. It trains a classifier on the given data initially. In each iteration, artificial training instances called diversity instances are generated based on the data distribution. Class labels for these artificial instances are chosen so as to differ maximally from the current ensemble's predictions. Then it trains a new classifier on the union of the original training instances and the diversity instances. If adding this new classifier to the current ensemble increases the ensemble training error, then this classifier is rejected, else it is added to the current ensemble. This process repeated until the desired committee size is reached or a maximum number of iterations is exceeded.

However, errors would be introduced into the ensemble if the artificial instances are not consistent with the target function. In current research, unlabeled instances could augment classifiers trained on labeled instances. Such ideas inspired the research on semi-supervised learning. Thus we modified the standard DECORATE method by incorporating the unlabeled instances into the artificial training instances. In each iteration of DECORATE, a few unlabeled instances on which the current ensemble has high certainty are chosen, pre-labeled by the current ensemble, and added into diversity instances. Note that these pre-labeled instances are just used to train the stronger classifier in the current iteration and still stay in the unlabeled set with no label waiting for active sampling. Those pre-labeled instances could prevent the ensemble from overfitting problem and provide distribution information for the ensemble. Therefore, the DECORATE method could preserve its diversity and accuracy.

3.3 Sampling Criterion

Based on the efficiency of instances for version space reduction, we specify the sampling criterion as

$$R_{x_i} = - \sum_{y \in Y} \|P_D^*(y|x_i) - P_D(y|x_i)\| \log \|P_D^*(y|x_i) - P_D(y|x_i)\| \quad (2)$$

where $P_D^*(x_i)$ denotes the probability distribution of class label predicted by the modified DECORATE trained on unlabeled set D , $P_D(x_i)$ denotes the probability distribution of class label predicted by the current base classifier trained on D . Then R_{x_i} is the entropy of the difference between these two probability distributions.

Therefore, the process of MSDEEUI method is given in Algorithm [1].

4 Experiments

To evaluate the performance of our MSDEEUI method, we ran a series of experiments. Five different active learning algorithms were tested: the Random

Algorithm 1. the MSDEEUI method

Input: an initial labeled set L , an unlabeled set UL , a classifier I , a stopping criterion S , and an integer M which specify the number of instances sampled in each iteration.

Begin:

Train the classifier I on L ;

repeat

1. Generate the ensemble h_L^* using the modified DECORATE method on L Enhanced by high certain instances in UL ;

2. For each instance $x_i \in UL$ compute

$$L(P_L^*(x_i), P_L(x_i)) = - \sum_{y \in Y} \|P_L^*(y|x_i) - P_L(y|x_i)\| \log \|P_L^*(y|x_i) - P_L(y|x_i)\|$$

3. Select a subset A of size M from UL in which instances x_i have the largest $L(P_L^*(x_i), P_L(x_i))$;

4. Remove A from UL ;

5. Label all instances in A ;

6. Add A into L ;

7. Train the classifier I on L .

until the stopping criterion S is satisfied

End.

Output: The classifier I trained by the final labeled set L .

sampling, the Uncertainty sampling [1], QBC [5], Active DECORATE [8] and the MSDEEUI sampling. The experiments were done on 16 representative datasets from machine learning repository provided by UCI [12]. The committee size were set to 5. Naive bayes was selected to be the base classifier. 10-fold cross-validation was used to obtain the target accuracy of the base classifier on the 16 datasets. The target accuracy is defined as the accuracy obtained by the base learning method trained on the whole dataset. All results presented were averages of ten runs. For each dataset, we divided it into 10 equal partitions at random and each in turn is used for testing and the remainder was used as the sampling set. Before the test started, the sampling set was divided into two parts: one is the labeled set and another is the unlabeled set. The labeled set contains only one instance selected randomly and the unlabeled set contains all the rest. When the test started, the active learner sampled 1 instance from the unlabeled set for labeling in each iteration. While the active learner reached the target accuracy, the test stopped.

We summarized the data utilization of the different active learners in Table 1. We define data utilization as the number of instances an active learner requires to reach the target accurate rate. In Table 1, the least data utilization is marked in bold in each row and the number of wins is presented in the last row. In the head of Table 1, the Uncertainty method is denoted by UC, the Active DECORATE method is denoted by AD and Target Accuracy is denoted by TA.

According to Table 1, it shows that our MSDEEUI method has a superior performance than other sampling methods on most datasets. Based on these

Table 1. Data utilization of the different active learners

Data set	Random	UC	QBC	AD	MSDEEUI	TA(%)
car	148.9	83.8	134.0	87.6	74.5	80.11
mushroom	2598.1	38.1	28.3	34.9	37.4	95.00
vote	172.3	74.8	107.2	117.9	90.5	95.01
waveform-5000	42.0	84.5	32.7	39.0	47.0	76.08
nursery	98.7	71.9	221.5	166.7	70.7	84.74
anneal	254.7	271.4	243.8	233.4	197.5	84.28
balance-scale	51.6	283.6	53.7	57.0	37.2	80.89
colic	220.8	350.9	236.3	56.0	41.8	76.61
credit-g	157.2	393.2	211.6	254.9	217.3	75.02
kr-vs-kp	163.0	48.7	84.8	67.0	156.3	84.22
mfeat-fourier	192.0	153.1	138.6	147.3	121.7	72.10
mfeat-pixel	154.9	86.2	157.6	74.2	79.4	88.97
segment	142.1	291.3	92.0	89.2	126.4	77.63
soybean	262.0	229.1	206.4	298.3	241.9	91.47
splice	136.7	91.6	118.4	414.7	169.9	88.21
vowel	227.8	300.0	169.0	244.3	110.3	61.66
No. of Wins	1	3	3	2	7	

results, we may conclude that MSDEEUI is more likely to reduce the size of version space as much as possible.

Figure 1 and Figure 2 show the results on the datasets of *car* and *vowel*, respectively. In all these figures, the vertical axis shows the accuracy of the classifier and the horizontal axis shows the number of labels.

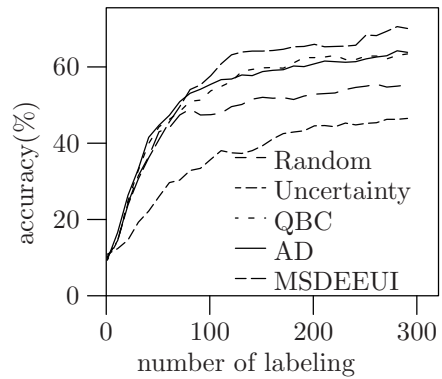
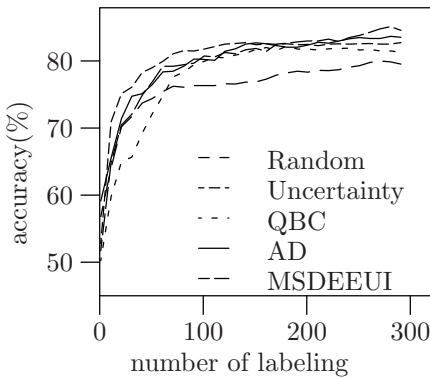


Fig. 1. Average testing accuracy on *car* **Fig. 2.** Average testing accuracy on *vowel*

In Figure 1, the Active DECORATE method achieves its maximal accuracy 83.67% at about the 291th sampling. Our MSDEEUI method requires 251 sampling to obtain the same accuracy. In Figure 2, the Active DECORATE method gets its highest accuracy, about 64.23%, after 291 sampling while the MSDEEUI method reaches the same accuracy at the 161th sampling. Furthermore, the other methods even can not get the accuracy of 64.23% before the 300th sampling.

5 Conclusion

Focusing on the sampling question in pool-based active learning, we visualize the process of version space reduction by proposing the Hypothesis-Instance matrix and its operation $TRN(x)$. Then we propose the MSDEEUI method, which samples the instances with the largest prediction difference between a strong classifier, generated by the DECORATE method enhanced by unlabeled instances with high prediction certainty, and the current base classifier. Experiments show that the MSDEEUI method is efficient and practical.

We would like to pursue following directions: finding a better function to obtain a more precise estimate of the target function and providing the theoretical proof of the converging speed of the version space using the MSDEEUI method.

Acknowledgments. This research was supported by the National Natural Science Foundation of China (No. 60603015, 60603062).

References

1. Lewis, D.D., Gale, W.A.: A sequential algorithm for training text classifiers. In: 17th ACM International Conference on Research and Development in Information Retrieval, pp. 3–12. Springer, Heidelberg (1994)
2. Muslea, I., Minton, S., Knoblock, C.A.: Active learning with multiple views. *Journal of Artificial Intelligence Research* 27, 203–233 (2006)
3. Campbell, C., Cristianini, N., Smola, A.: Query learning with large margin classifiers. In: Proc. 17th International Conf. on Machine Learning, Madison, pp. 111–118. Morgan Kaufmann, San Francisco (2000)
4. Roy, N., McCallum, A.: Toward optimal active learning through sampling estimation of error reduction. In: Proc. 18th International Conf. on Machine Learning, pp. 441–448. Morgan Kaufmann, San Francisco (2001)
5. Seung, H.S., Oppor, M., Sompolinsky, H.: Query by committee. In: Proceedings of the Fifth Workshop on Computational Learning Theory, San Mateo, pp. 287–294. Morgan Kaufmann, San Francisco (1992)
6. Freund, Y., Seung, H.S., Shamir, E., Tishby, N.: Selective sampling using the query by committee algorithm. *Machine Learning* 28, 133–168 (1997)
7. Abe, N., Mamitsuka, H.: Query learning using boosting and bagging. In: Proc. 15th International Conf on Machine Learning, Madison, pp. 1–10. Morgan Kaufmann, San Francisco (1998)
8. Melville, P., Mooney, R.J.: Diverse ensembles for active learning. In: Proc. 21th International Conf. on Machine Learning, Banff, CA, pp. 584–591. Morgan Kaufmann, San Francisco (2004)
9. Dietterich, T.G.: Machine-learning research: Four current directions. *The AI Magazine* 18(4), 97–136 (1998)
10. Hansen, L., Salamon, P.: Neural network ensembles. *IEEE Trans. Pattern Analysis and Machine Intell.* 12, 993–1001 (1990)
11. Melville, P., Mooney, R.J.: Constructing diverse classifier ensembles using artificial training examples. In: *IJCAI*, pp. 505–512 (2003)
12. Newman, D., Hettich, S., Blake, C.L., Merz, C.J.: Uci repository of machine learning databases (1998), <http://www.ics.uci.edu/~mllearn/mlrepository.html>

A New Model for Image Annotation

Sanparith Marukatat

Image Laboratory,
National Electronics and Computer Technology Center (NECTEC),
112 Thailand Science Park, Phahon Yothin Road,
Klong Luang, Pathumthani 12120, Thailand
`sanparith.marukatat@nectec.or.th`

Abstract. An approach to automatic image annotation is proposed. Generally, the relation between visual characteristics and the annotation label is estimated from the annotated corpus and is used to predict label for new test image. Unfortunately, when limited number of images are annotated, with possible multiple labels per image, this relation cannot be reliably estimated. To cope with this problem, we propose taking into account information derived directly from other images in the dataset. This method extends naturally to semi-supervised setting where un-annotated images are also used select annotation labels. Experiment shows that the proposed method yields promising results.

1 Introduction

The phrase “A picture is worth a thousand words.” clearly depicts the difficulty in image annotation. The objective of this work is to build a system that can automatically provide an *annotation* that describes the input image. Table 2 shows example of annotated images used in this work.

Common approach to solve the image annotation problem (e.g. [1,2,3,4,5]) consists in partitioning the whole image into regions. From this segmentation, a set of local descriptors is extracted from each region to describe its visual and texture characteristics. From a pool of local features extracted from all images, a clustering algorithm, usually K-means algorithm, is applied to select the set of features that will be used to represent the image. Indeed, each image will be represented by the set of these selected features or visterms in an analogous manner to the bag of words representation used in textual information retrieval.

Next, a learning technique is used to capture the relation between these blobs or visterms and the annotation label. Indeed, the annotation model computes the probability of a given label being present in the annotation, or *label posterior* for short, of an input image. In [1], the conditional probability of label given blob is trained using EM algorithm from all annotated images. We consider this as a *global approach*. When limited number of images is annotated the resulting model cannot efficiently capture all information. Moreover, not every object or region that appears in the image is annotated and on the other hand several labels may be used to annotate the same image. Annotated objects may also appear

on complex background. The probability estimated on this type of corpus is therefore noisy. As consequent, the annotation model should not rely on the global information solely.

The Cross-Media Relevance Model (denoted as CMRM hereafter) [2] and similar models [2,3,6], apply a smoothing technique to combines the global probability function with the probability function computed locally on each annotated image. Integrating this *local information* allows better estimation of the label posterior leading to a more accurate annotation.

This work also relies on the similar idea, but pushing a bit further. Indeed, we expand the computation of local information from each image, to include its neighborhoods images as well. In fact, we compute the label posterior of an image as a function of three quantities namely; the label posterior on other images in the dataset, the similarity between other images and the input image, and the probability of label being assigned to this image if we know that it has also been assigned (or not) to other images. This framework can take advantage of un-annotated images in the training corpus as described in the next section.

Besides, it is easier to collect new images rather than annotating them. Therefore, it is interesting to see how to exploit these un-annotated images in the annotation model. This idea is shared by several semi-supervised learning algorithms. Another objective of this work is then to integrate these un-annotated images into the annotation model and to investigate how this additional data can improve the performance of the system.

In the following, Section 2 presents our image annotation system. Section 3 and 4 presents the experimental results and the conclusion respectively.

2 Proposed Method

Given an un-annotated image, we want to automatically select a set of annotation labels from a set of known labels that best describe this image. An annotation model is constructed for each label independently. This allows integrating new label easily. We follow the common approach describe in previous section by first segment the image into regions, extract local descriptors and use bag-of-visual-words representation. Indeed, each image is represented by a normalized histogram of visual-words. This histogram along with the histogram intersection are used as a building block to compute different probabilities involved in our system.

2.1 Image Annotation with Global Information

In this subsection, we present a simple global image annotation model. The calculation used for this model is also used later in our proposed annotation model described in next subsection. For each label L , two histograms are computed from the set of annotated images namely L^{yes} from images having this label in the annotation and L^{no} from images without this label in the annotation. As these two histograms are computed from all annotated images, we consider it as global information.

For each image $I_i, i = 1, \dots, n$ in the training set, let L_i denote the presence of label L in the annotation of this image. For an image I_i , the posterior probability of having the label for I_i using this global information or $P_G(L_i = yes/I_i)$ can be computed by

$$P_G(L_i = yes/I_i) = \frac{P(L_i = yes, I_i)}{P(L_i = yes, I_i) + P(L_i = no, I_i)} \tag{1}$$

$$= \frac{L^{yes} \cap x_i}{(L^{yes} \cap x_i) + (L^{no} \cap x_i)} \tag{2}$$

where \cap denotes the histogram intersection operation.

2.2 Image Annotation Model Using Local Information

For each image $I_i, i = 1, \dots, n$ in the training set, let L_i denote the presence of label L in the annotation of this image. For an image I_i , the posterior probability of having the label for I_i or $P(L_i = yes/I_i)$ can be computed by

$$P(L_i = yes/I_u) = \sum_{j=1}^n \sum_{L_j \in \{yes, no\}} P(L_i = yes, I_j, L_j/I_i) \tag{3}$$

We further assume that the join probability $P(L_i = yes, I_j, L_j/I_i)$ can be factorized into

$$P(L_i = yes, I_j, L_j/I_i) = P(L_i = yes/I_j, L_j, I_i)P(L_j/I_j)P(I_j/I_i) \tag{4}$$

To simplify the notation, let

$$f_i = P(L_i = yes/I_i) \tag{5}$$

$$a_{ji} = P(I_j/I_i) \tag{6}$$

$$b_{ji}(yes) = P(L_i = yes/I_j, L_j = yes, I_i) \tag{7}$$

$$b_{ji}(no) = P(L_i = yes/I_j, L_j = no, I_i) \tag{8}$$

The equation 3 may be rewritten as

$$f_i = \sum_j a_{ji} (b_{ji}(yes)f_j + b_{ji}(no)(1 - f_j)) \tag{9}$$

The last equation is used to update the label posterior for un-annotated images iteratively. For an annotated image I_i , if the given label has been assigned to I_i then f_i is set to 1 and 0 otherwise. For an un-annotated images I_i the value f_i after convergence will be used to select the set of labels for I_i .

The transition probability (a_{ji} in the equation 6) can be computed using the simple histogram intersection operation, i.e.

$$a_{ji} = \frac{I_i \cap I_j}{\sum_k I_k \cap I_i} \tag{10}$$

To compute the *local posterior probability* (b_{ji} in equations 7 and 8), we first notice that $b_{ji}(yes) = P(L_i = yes/I_j, L_j = yes, I_i)$ can be interpreted in the following way; We know that I_j has label L and we want to compute the probability of I_i having label L too. If we consider the distribution of visterms for the class of images having the label, then around I_j this distribution should be more similar to the distribution that represents I_j rather than L^{yes} . Therefore, we propose computing $P(L_i = yes/I_j, L_j = yes, I_i)$ in an analogous manner to $P_G(L_i = yes/I_i)$ (equation 2), but with a distribution that is biased toward I_j . To this end, we define

$$L^{j,yes} = \alpha I_j + (1 - \alpha)L^{yes} \tag{11}$$

$$L^{j,no} = \alpha I_j + (1 - \alpha)L^{no} \tag{12}$$

where α is a trade-off coefficient between local and global information, and $b_{ji}(yes)$, $b_{ji}(no)$ are computed as follow:

$$b_{ji}(yes) = \frac{L^{j,yes} \cap I_i}{(I_i \cap L^{j,yes}) + (I_i \cap L^{no})} \tag{13}$$

$$b_{ji}(no) = \frac{L^{yes} \cap I_i}{(I_i \cap L^{yes}) + (I_i \cap L^{j,no})} \tag{14}$$

3 Experiments

3.1 Experimental Setup

To construct the dictionary of visterms, each image is first partitioned into 10x10 pixels regions. The mean and standard deviation of three color channels R, G, and B were computed from each region. Four Gabor filters were applied to the image, the mean and standard deviation of these response were also computed. In total 14 features were used in this work. In the quantization step, K-means algorithm was used with K=1000. The trade-off parameter α was experimentally set to 0.9. The average precision (AP) which is as the average value of maximum precision of this system at different recall rates is computed for each label. The mean of AP for every label is used to measure the performance of the annotation model.

The experiments were performed on a set of 2360 photography images¹. Every image is manually annotated. Fifty most frequent labels were retained for the experiment. Table 2 shows some images and their given annotations. This dataset is randomly split into a development set of 1860 images and a test set of 500 images. To investigate how un-annotated images may help improving the system’s accuracy, we randomly selected 20% of development set (372 images) as training data with annotation. Then we randomly add 30% and 80% more images from the development set into the training set, but without their annotation. This evaluation was repeated 10 times.

¹ <http://www.stat.psu.edu/~jjali/index.download.html>

3.2 Results

The median of the mean AP with the confidence interval from the CMRM model [2], the simple annotation model with only global information and the proposed model are shown in Figure 1. From this figure, we can see that the CMRM

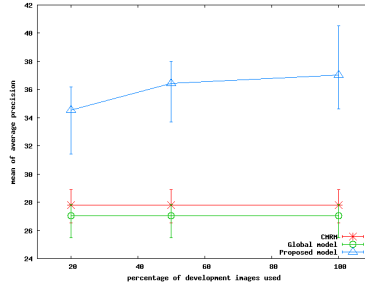




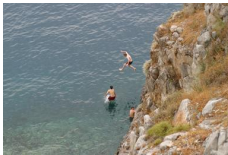



Fig. 1. Results from different models with 20% annotation, and different size of development set used in the construction of models (see text for more detail)

Table 1. Results from different model using all development set as training set

model	mean of average precision
CMRM	32.68%
simple global model	31.77%
proposed model	58.72%

Table 2. Example of images and its annotations both manual and automatically using the proposed model

			
manual	flower	leaves	tree, lawn, house
automatic	flower	plant	tree, lawn
			
manual	boat, reflexion, water	ocean, rock, people	water, tree, grass, boat
automatic	reflexion, water, fishing, house, tree	rock, ocean	tree, greece, boat

model outperforms the simple global model. We believe this is due to the use of local information in CMRM. The proposed model outperforms both the CMRM model and the simple global model. Moreover, the performance of the proposed model increases as the size of available training data increases, even without given annotation. The Table 1 presents the results of the these models trained with all development data as training data. These results also underline the superiority of our model compared to the two others.

To annotate an image using this model, let \mathcal{L} be the set of selected labels, its conditional probability given an input image I may be written as

$$P(\mathcal{L}/I) = \prod_{L \in \mathcal{L}} P(L = \text{yes}/I) \times \prod_{L \notin \mathcal{L}} (1 - P(L = \text{yes}/I)) \quad (15)$$

Using this equation, we may choose the set \mathcal{L} of labels with maximum probability as the annotation for input image I . This strategy determines automatically the length of the annotation. Table 2 shows some example of annotation provide by the proposed model.

4 Conclusion and Future Works

An image annotation model is proposed. The proposed model integrates local information extracted from each image and its neighborhood within a probabilistic framework. This model can be used to select appropriate labels for an input image. The evaluation of this method on a more standard database like the Corel dataset will be investigated in our future work.

References

1. Duygulu, P., Bernard, K., de Freitas, N., Forsyth, D.: Object recognition as machine translation: Learning a lexicon for a fixed image vocabulary. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002. LNCS, vol. 2353, pp. 97–112. Springer, Heidelberg (2002)
2. Jeon, J., Lavrenko, V., Manmatha, R.: Automatic image annotation and retrieval using cross-media relevance models. In: Proceedings of the ACM SIRGIR 2003 (2003)
3. Feng, S., Manmatha, R., Lavrenko, V.: Multiple bernoulli relevance models for image and video annotation. In: Proceedings of the CVPR 2004 (2004)
4. Jeon, J., Manmatha, R.: Using maximum entropy for automatic image annotation. In: Enser, P.G.B., Kompatsiaris, Y., O'Connor, N.E., Smeaton, A.F., Smeulders, A.W.M. (eds.) CIVR 2004. LNCS, vol. 3115, pp. 24–32. Springer, Heidelberg (2004)
5. Jin, R., Chai, J.Y., Si, L.: Effective automatic image annotation via a coherent language model and active learning. In: Proceedings of the 12th annual ACM international conference on Multimedia (2004)
6. Lavrenko, V., Manmatha, R., Jeon, J.: A model for learning the semantics of pictures. In: Advances in Neural Information Processing Systems (NIPS) (2003)

Unmixed Spectrum Clustering for Template Composition in Lung Sound Classification

Tomonari Masada, Senya Kiyasu, and Sueharu Miyahara

Nagasaki University, 1-14 Bunkyo-machi, Nagasaki 852-8521, Japan
{masada,kiyasu,miyahara}@cis.nagasaki-u.ac.jp

Abstract. In this paper, we propose a method for composing templates of lung sound classification. First, we obtain a sequence of power spectra by FFT for each given lung sound and compute a small number of component spectra by ICA for each of the overlapping sets of tens of consecutive power spectra. Second, we put component spectra obtained from various lung sounds into a single set and conduct clustering a large number of times. When component spectra belong to the same cluster in all clustering results, these spectra show robust similarity. Therefore, we can use such spectra to compose a template of lung sound classification.

1 Introduction

Real-world problems provide various spectral data. We often regard a given set of such spectra as a mixture of elementary spectra, which can be computed by some *spectral unmixing* method. In this paper, we apply independent component analysis (ICA) [5] to lung sound power spectra and approximate a set of tens of power spectra with a mixture of a small number of spectra which we call *component spectra*. Further, we filter out less effective component spectra by conducting clustering a large number of times on a set of component spectra obtained from various lung sounds. Component spectra belonging to the same cluster over all clustering results can be regarded as robustly similar with each other and may show an outstanding feature of a specific type of lung sounds. Therefore, we regard only such component spectra as efficient ones and use them to make templates for lung sound classification.

The rest of the paper is organized as follows. Section 2 provide the results of previous researches. Section 3 describes details of our method. In Section 4, we show the settings and the results of our evaluation experiment. Section 5 concludes this paper along with our future works.

2 Previous Work

We are experiencing a new wave of lung sound analysis due to the import of various machine learning techniques in recent years [1]. However, there seem no researches applying ICA not to lung sounds but to their power spectra obtained by FFT. Güler et al. [3] reduce the feature space dimension to two and provide

a visual classification of lung sounds drawn in two-dimensional space. Pelletier [9] regards each power spectrum as a histogram and provides a reduction of the number of bins with principal component analysis. In contrast, we do not reduce the dimension of power spectra. We assume that the number of component spectra, which are mixed to produce a given set of power spectra, is small and obtain a set of such component spectra by applying a spectral unmixing. Other research fields provide this type of approaches. In the field of remote sensing, we obtain a compact representation of a hyperspectral image by extracting component spectra, called *endmember spectra* [8]. Our research follows this line.

3 Details of Our Method

First of all, we obtain power spectra from a given lung sound by using FFT with Hanning window. In other sets of experiments, whose results are not included in this paper, we confirm that Hamming window also gives similar results, but that flattop window is not effective. We make any two consecutive windows share the half of their lengths. Let each window include $2T$ data points. When the sampling frequency is f Hz, each window corresponds to an interval of length $2T/f$ sec. In this paper, we set $T = 2048$. Two key modules are described below.

Spectral Unmixing. We call a set of N consecutive windows *frame*. Since two consecutive windows overlap by one-half the window length, each frame is of length $(N + 1)T/f$ sec. In this paper, we set $N = 32$. We make two consecutive frames share $3N/4$ windows. N power spectra obtained from each frame are approximated by a linear mixture of M ($M \ll N$) spectra, called *component spectra*. We use *fastICA* algorithm [5] to obtain component spectra. The update formula proposed in [4] is employed to achieve stable convergence. Since this paper is intended to reveal basic characteristics of our method, we choose the most simple set up for M . That is, we set $M = 2$. We call a pair of component spectra extracted from each frame *component pair*.

Component Spectra Clustering. In our setting, we can obtain many component pairs from a lung sound. We put component spectra taken from various lung sounds into a single set and conduct clustering over this set a large number of times. In this paper, we conduct k -means 100 times. We use no information about which two of component spectra come from the same component pair. If two component spectra from the same component pair belong to the same cluster in many of the 100 clustering results, we can conclude that the unmixing method we adopt is not effective, because a large number of executions of clustering provide a robust estimation of similarities between component spectra. We write $\mathbf{q} \sim \mathbf{q}'$ when two component spectra \mathbf{q} and \mathbf{q}' belong to the same cluster over all clustering results. We call this relation *coupling*. Let $(\mathbf{q}_1, \mathbf{q}_2)$ and $(\mathbf{q}'_1, \mathbf{q}'_2)$ be two component pairs obtained from different frames. $\mathbf{q}_1 \sim \mathbf{q}_2$ and $\mathbf{q}'_1 \sim \mathbf{q}'_2$ are undesirable, because component spectra from the same component pair are obtained by ICA and thus are expected to be dissimilar. We call this type of couplings *malicious*. When we use NMF algorithm in [6] in place of fastICA, a large number of couplings turn out to be malicious. This result shows the low quality

of this version of NMF unmixing. By excluding malicious couplings, we have the following cases of similarity between component pairs $(\mathbf{q}_1, \mathbf{q}_2)$ and $(\mathbf{q}'_1, \mathbf{q}'_2)$:

1. Both $\mathbf{q}_1 \sim \mathbf{q}'_1$ and $\mathbf{q}_2 \sim \mathbf{q}'_2$ hold, or both $\mathbf{q}_1 \sim \mathbf{q}'_2$ and $\mathbf{q}_2 \sim \mathbf{q}'_1$ hold.
2. Either $\mathbf{q}_1 \sim \mathbf{q}'_1$ or $\mathbf{q}_2 \sim \mathbf{q}'_2$ holds, or either $\mathbf{q}_1 \sim \mathbf{q}'_2$ or $\mathbf{q}_2 \sim \mathbf{q}'_1$ holds.

For Case 1, we say that both couplings are *perfect*. The couplings in Case 2 are called *imperfect*. Perfect couplings tell a strong similarity between the two frames corresponding to the two component pairs $(\mathbf{q}_1, \mathbf{q}_2)$ and $(\mathbf{q}'_1, \mathbf{q}'_2)$.

4 Experiment

Settings. We use lung sounds recorded in the CD accompanying a textbook for nurses [2]. After identifying different sound data corresponding to lung sounds of the same type, we have lung sounds splitted into 39 categories shown in Table 1. When we use each power spectrum in their full range, many malicious couplings are obtained. High-frequency part seems to have an undesirable effect to our method. Therefore, we eliminate high-frequency part and test the following settings for the range of spectra: 1~256, 1~512, 257~512, and 257~1024. Recall that $T = 2048$. As for the last two cases, we also eliminate low-frequency part. However, the third setting results in a few malicious couplings. This shows that the range 513~1024 has some importance when we discard the range 1~256.

At the initialization of k -means, we set the number of clusters to 100. In the course of the execution of a clustering, some clusters eventually get empty. For example, we obtain from 59 to 79 clusters and 69.7 clusters in average for 100 executions of k -means when we use 1st to 512th entries of power spectra.

Results. Table 1 includes the results when we use 1st to 512th entries of power spectra. Column A presents the number of frames obtained from the lung sounds of each category. The total number of frames is 3,357. Since ICA provides two component spectra for each frame, we obtain a set of 6,714 component spectra. The results of 100 executions of k -means induce perfect and imperfect couplings. Column B shows the number of frames giving imperfect or perfect couplings. Column C shows the number of frames giving perfect couplings. When the value in Column B is small, we have some trouble in processing the lung sounds of the corresponding category, because we can expect that component spectra from the lung sounds of the same category form at least imperfect couplings. For example, the category of ID 35 has a small value in Column B. This may be due to the fact that the sound is in small volume. It is, however, beyond our scope to propose preprocessing methods appropriate for our method.

Column D (resp. Column E) gives the number of imperfect (resp. perfect) couplings which include at least one component spectra from the corresponding category. All numbers in Column E are even, because every perfect coupling comes with another perfect coupling. When a coupling includes two component spectra from the same category, we call such a coupling *correct*. Otherwise, a coupling is called *incorrect*. For the category of ID 10, all of 10812 perfect couplings turn out to be correct. That is, each of these perfect couplings consists

Table 1. Experimental results obtained when we apply Hanning window in FFT and use 1st to 512th entries of power spectra

ID	lung sound type	A	B	C	D	E
01	normal vesicular breath sounds	525	391	210	17233	8748
02	normal bronchovesicular breath sounds	84	81	67	3522	716
03	normal tracheal breath sounds	44	42	18	896	206
04	decreased breath sounds (atelectasis)	129	114	63	4882	358
05	absent breath sounds (hemothorax)	134	59	34	973	220
06	increased breath sounds	154	148	108	8678	4280
07	bronchial sounds in abnormal locations	73	36	2	333	2
08	low pitched rhonchi	146	103	40	951	114
09	high pitched rhonchi	60	44	10	851	10
10	fine crackle	179	174	157	2423	10812
11	coarse crackle	201	192	118	7958	1988
12	spontaneous pneumothorax	29	15	0	151	0
13	normal vocal sounds (to be compared with 14)	27	22	10	169	58
14	spontaneous pneumothorax (vocal sounds)	30	25	17	123	80
15	pleural effusion (effusion side)	43	28	2	1021	2
16	pleural effusion (non-effusion side)	38	23	0	1239	0
17	asthma	136	133	88	3029	444
18	asthma (partial recovery)	56	36	5	412	8
19	asthma (nearly complete recovery)	45	28	6	704	10
20	pulmonary edema	100	82	52	967	938
21	pulmonary edema (nearly complete recovery)	33	27	8	290	24
22	chest drain sound	202	142	47	496	138
23	tracheal stenosis	56	41	9	183	12
24	tracheal stenosis (partial recovery)	75	39	15	123	56
25	pneumonia	115	93	22	1815	92
26	pneumonia (partial recovery)	59	46	19	802	106
27	pneumonia (nearly complete recovery)	43	26	5	200	8
28	pleuritis	118	115	101	2285	4796
29	congestive heart failure (early stage)	45	33	2	210	2
30	congestive heart failure (late stage)	31	15	0	156	0
31	congestive heart failure (late stage, mouth breathing)	24	22	12	87	30
32	congestive heart failure (recovered)	62	62	50	2511	992
33	congestive heart failure (recovered, mouth breathing)	32	21	7	164	20
34	sputum collection tube (before sputum collection)	31	15	5	246	20
35	sputum collection tube (after sputum collection)	34	6	0	5	0
36	air leak around tracheal tube (before adding air)	41	31	12	240	26
37	air leak around tracheal tube (after adding air)	23	21	19	71	142
38	subcutaneous emphysema	40	21	7	98	10
39	pneumothorax	60	28	2	182	2

of two component spectra taken from the lung sounds of this category. We can conclude that this category exhibits a strong self-similarity. In contrast, categories of ID 12, 16, 30 and 35 provide no perfect couplings. Among these four categories, the sounds of categories 12 and 35 are in small volume. Further, the

lung sounds of category 16 have an irregularly long respiration cycle. These categories require appropriate preprocessing methods, which is our important future work. The lung sounds of category 30 seem to show rapid changes in the same respiration cycle. We may need to try other settings for T or for the overlapping widths of windows. The category of ID 17 provides interesting results. While all 444 perfect couplings are correct, the number of imperfect couplings is 3,029, far larger than 444. Moreover, 2,330 among these 3,029 imperfect couplings are correct. We may explain these results by the fact that the lung sounds of this category show poor periodicity. Stable characteristics is reflected by one half of each component pair, and changeable characteristics is reflected by another.

Evaluation of Results. We can make equivalence classes by regarding perfect coupling as an equivalence relation. These equivalence classes are, in a sense, “meta-clusters” constructed based on multiple clustering results. Each equivalence class can provide a powerful clue to form templates useful in classification of unknown lung sounds. It is an important future work to devise a procedure of template composition for the categories giving no perfect couplings, e.g. categories of ID 12, 16, 30 and 35, because we can have no meta-clusters for these categories. Table 2 presents an evaluation of this meta-clustering. The number of categories which give at least two perfect couplings is shown in the column marked by “*” for each setting. The number in the column marked by “†” refers to the number of component spectra which are taken from the corresponding category and, at the same time, are included in at least one perfect couplings. Table 2 also presents an evaluation of meta-clustering. We use 39 categories presented in Table 1 as the ground truth. The evaluation measures are microaveraged/macroaveraged precision, microaveraged/macroaveraged recall and their harmonic mean. These measures are often used in the evaluation of clustering and are defined as follows. The *dominating category* of a cluster is the category providing the largest number of component spectra to that cluster. The *precision* of a cluster is equal to the number of component spectra from the dominating category divided by the cluster size, i.e., the number of component spectra included in the cluster. The *recall* of a cluster is the number of component spectra from the dominating category divided by the number of all component spectra from that category. We can summarize the precisions and the recalls of all obtained clusters by microaveraging or by macroaveraging. *Microaveraged precision* is equal to the sum of the numerators of all precisions divided by the sum of the denominators of all precisions. We can obtain *microaveraged recall* in the same manner. *Macroaveraged precision* (resp. *macroaveraged recall*) is computed simply as an arithmetic mean of all precisions (resp. all recalls).

In all settings, we have high precisions and low recalls, because the numbers of clusters, shown in the last column, are far larger than the number of categories. However, our method is designed for template composition in classification task. Therefore, each category can have several templates. The number of dominating categories is shown in the column “**”. When this number is smaller than that presented in the column “*”, there exist categories which can dominate no clusters. We have no meta-clusters also for such categories. Only when we use 1st

Table 2. Evaluation of “meta-clusters” induced by perfect couplings

range	*	†	precision		recall		harmonic mean		**	# of clusters
			micro	macro	micro	macro	micro	macro		
1-256	37	1276	0.9561	0.9473	0.0778	0.1500	0.1438	0.2590	36	228
1-512	35	1349	0.9711	0.9688	0.0680	0.1343	0.1272	0.2358	35	252
257-512	31	1017	0.5320	0.9365	0.0816	0.1519	0.1416	0.2614	28	130
257-1024	31	657	0.8706	0.9637	0.0546	0.1236	0.1028	0.2191	30	208

to 512th entries of power spectra, every category dominates at least one cluster. We also have the best precision, i.e., 0.9711, for this setting.

5 Conclusion and Future Work

In this paper, we propose a method for detecting robust similarities between short intervals taken from various lung sounds. The results of evaluation show that we can obtain equivalence classes of component spectra with high precision. However, some types of lung sounds do not provide meaningful similarities. We need to propose preprocessing methods for such lung sounds. Further, in the near future, we will provide the procedures for composing templates and will devise a method for comparing unknown lung sounds with the templates.

References

1. <http://www.rale.ca/pub/RRC.PDF>
2. Yonemaru, M., Sakurai, T.: Lung Sound Auscultation Training via CD for Nurses. Nanko-do, Tokyo (2001)
3. Güler, E.C., Sankur, B., Kahya, Y.P., Raudys, S.: Visual Classification of Medical Data Using MLP Mapping. *Computers in Biology and Medicine* 28, 275–287 (1998)
4. Hyvärinen, A.: Fast and Robust Fixed-Point Algorithms for Independent Component Analysis. *IEEE Transactions on Neural Networks* 10(3), 626–634 (1999)
5. Hyvärinen, A., Karhunen, J., Oja, E.: *Independent Component Analysis*. Wiley-Interscience, Chichester (2001)
6. Lee, D.D., Seung, H.S.: Algorithms for Non-negative Matrix Factorization. *Advances in Neural Information Processing Systems* 13, 556–562 (2001)
7. Marshall, A., Boussakta, S.: Signal Analysis of Medical Acoustic Sounds with Applications to Chest Medicine. *J. of the Franklin Inst.* 344(3-4), 230–242 (2007)
8. Miao, L., Qi, H., Szu, H.: Unsupervised Decomposition of Mixed Pixels Using the Maximum Entropy Principle. In: *Proceedings of ICPR 2006* (2006)
9. Pelletier, C.: *Classification des Sons Respiratoires en vue d’une Détection Automatique des Sibilants*. Master Thesis, Université du Québec à Rimouski (2006)

Forward Semi-supervised Feature Selection

Jiangtao Ren¹, Zhengyuan Qiu¹, Wei Fan², Hong Cheng³, and Philip S. Yu⁴

¹ Department of Computer Science, Sun Yat-Sen University, Guangzhou, China*

issrjt@mail.sysu.edu.cn, qzhengy@mail2.sysu.edu.cn

² IBM T.J.Watson Research, USA

weifan@us.ibm.com

³ Computer Science Department, UIUC, USA

hcheng3@uiuc.edu

⁴ Computer Science, University of Illinois at Chicago, USA

psyu@cs.uic.edu

Abstract. Traditionally, feature selection methods work directly on labeled examples. However, the availability of labeled examples cannot be taken for granted for many real world applications, such as medical diagnosis, forensic science, fraud detection, etc, where labeled examples are hard to find. This practical problem calls the need for “semi-supervised feature selection” to choose the optimal set of features given both labeled and unlabeled examples that return the most accurate classifier for a learning algorithm. In this paper, we introduce a “wrapper-type” forward semi-supervised feature selection framework. In essence, it uses unlabeled examples to extend the initial labeled training set. Extensive experiments on publicly available datasets shows that our proposed framework, generally, outperforms both traditional supervised and state-of-the-art “filter-type” semi-supervised feature selection algorithms [5] by 1% to 10% in accuracy.

Keywords: feature selection, semi-supervised learning.

1 Introduction

Feature selection is an important data processing step in high dimensional data learning tasks. Traditionally, feature selection methods use information from “labeled data” to find the most informative or most useful feature subsets [1,2], but the information in the “unlabeled” data is not used. When the size of the “labeled” data is limited, it is difficult to select an ideal feature subset only on “labeled” data. Recently, there have seen considerable interests in learning with labeled and unlabeled data [3]. In many learning tasks, the effectiveness of semi-supervised learning has been demonstrated [4]. Zhao and Liu [5] introduced a semi-supervised feature selection algorithm based on spectral analysis. Later, they exploited intrinsic properties underlying supervised and unsupervised feature selection algorithms, and proposed a unified framework for feature

* Supported by the National Natural Science Foundation of China under Grant No. 60703110.

selection based on spectral graph theory [6]. Yet these algorithms are “filter” models which 1) do not select feature subset for specific learning method and 2) sample selection bias is ignored. It is important if one can employ certain strategy to extend the initial training set with unlabeled data to overcome the biased distribution problem, and in the same time, perform a “wrapper type” feature selection for specific learning model.

In this paper, we introduce a “wrapper-type” forward semi-supervised feature selection framework. It uses the mechanism of random selection on unlabeled data to form new training sets, and the most frequently selected feature is added to the result feature subset in each iteration. With the introduction of randomly selected data with predicted labels, the sufficiency and diversity of the training sets can be improved, which in return helps to choose the most discriminative features.

In order to evaluate the effectiveness of our framework, we give formal analysis as well as conduct extensive experimental study on the algorithm. First, bipartite graph has been employed to formally show that unlabeled data is helpful in feature selection. Secondly, we have conducted extensive experiments with extremely few labeled instances (such as, only 6 labeled instances) which can reflect the scenario of data limitation. The results of these experiments show that the proposed “wrapper-type” framework, generally, outperforms the traditional feature selection method and state-of-the-art “filter-type” semi-supervised feature selection algorithms [5] by 1% to 10% in accuracy. It performs especially well when the size of the labeled data set is very small.

2 The Framework

Supervised sequential forward feature selection (SFFS) is one of the most widely used feature selection algorithms. Conceptually, it is an iterative process starting with an empty feature subset. In each iteration, one feature is chosen among the remaining features. To determine which feature to add, it tests the accuracy of a model built on the incremented feature subset. The feature that results in the highest accuracy is selected. Normally, the process terminates when no additional features could result in an improvement in accuracy or the feature subset already reaches a predefined size. Since this process can be easily implemented and is usually quite effective, it remains one of the widely adopted supervised feature selection methods. In this work, we extend it to take unlabeled data into account.

2.1 Our Approach

We propose a new framework of forward feature selection, which performs feature selection on both labeled and unlabeled data. Our algorithm uses SFFS and wrapper model to select $startfn$ features initially, then the $startfn$ features are used to train a classifier, the classifier is then used to predict the labels of the unlabeled data. Then the randomly selected $samplingRate\%$ unlabeled data with predicted labels is combined with labeled data to form a new training set. Afterwards, the new training dataset is used to select $fnstep$ features based on

```

Input:  $L, U, sizeFS, samplingRate, samplingTimes, maxIterations,$ 
          $startfn, fnstep$ 
Output:  $resultfs$ 
1 Perform feature selection on  $L$  using SFFS, select  $startfn$  features to
  form the current feature subset  $currentfs$ ;
2  $ReducedL \leftarrow L * currentfs$ ;
3  $ReducedU \leftarrow U * currentfs$ ;
4 for  $iteration \leftarrow 1$  to  $maxIterations$  do
5    $Predicted \leftarrow \text{classifier}(ReducedL, ReducedU)$ ;
6   for  $rand \leftarrow 1$  to  $samplingTimes$  do
7     Randomly select  $samplingRate\%$  of instances from  $Predicted$ ,
      and add it into  $L$  to form a new dataset  $NewDataset$ ;
8     Perform feature selection on  $NewDataset$  using SFFS, select
       $fnstep$  features to form feature subset  $fs[rand]$ ;
9   end
10  Count the frequency of every feature in  $fs$ , add the most frequent
    and not in  $currentfs$  feature into  $currentfs$ ;
11   $ReducedL \leftarrow L * currentfs$ ;
12   $ReducedU \leftarrow U * currentfs$ ;
13  if  $SIZE(currentfs) == sizeFS$  then break;
14 end
15  $resultfs \leftarrow currentfs$ ;

```

Fig. 1. Forward semi-supervised feature selection (FW-SemiFS)

SFFS and the learner. The random selection and feature selection process repeat $samplingTimes$ times and $samplingTimes$ groups of features are selected. And we count the frequency of every feature in the $samplingTimes$ groups of features, and the one with the most frequency is added to form a new feature subset. This process repeats until the size of the feature subset reaches a predefined number.

The algorithm is described in detail in Figure 1. L denotes the labeled data, U denotes the unlabeled data; $sizeFS$ denotes the predefined number of selected features; $samplingRate$ denotes the sampling rate according to the unlabeled data with predicted labels; $samplingTimes$ denotes the randomly sampling times; $maxIterations$ denotes the max iteration times; $startfn$ denotes the start feature number; $fnstep$ denotes the number of features selected in every step. $resultfs$ denotes the output feature subset. In our algorithm, “*” denotes the features reduction operator.

2.2 Method Analysis

The classifier “wrapped” in the feature selection algorithm is used to predict the labels of the unlabeled instances as well as to evaluate the effectiveness of the chosen feature subset. Obviously, a more accurate classifier can select more effective feature subsets to represent the target distribution. Next, we will demonstrate why the use of unlabeled data can improve the “accuracy” of the classifier at each step of feature selection.

Feature selection can be formulated as: $f_1(X') = f(X)$ subject to make X' as small as possible, where f_1 and f are the target functions on chosen feature subset X' and full feature set X , respectively. When unlabeled examples are used in feature selection, its process is similar to “co-training”, where X' and X are interpreted as the two “views” in co-training. The classifier constructed in one view is expected to be helpful for the other.

Now we can adopt the bipartite graph model to facilitate our interpretation on why “wrapper classifier” with unlabeled examples can improve the performance of feature selection. The training process can be regarded as a bipartite graph $G_D(X'; X)$ [7]. In Figure 2, each node on the left-hand side denotes one instance in X' view and the one on the right-hand side denotes the same instance in X view. Any two instances in the labeled dataset will be connected by an edge if they belong to the same class, and those edges are shown in solid lines. Let S be a finite dataset that consists of the labeled dataset, then G_S is a bipartite graph in S whose components show the concept distribution in G_S . In the case of unlabeled instances, each vertex on the left, depending on the its prediction on X' view, will be connected to a vertex on the right. Thus, it is connected to the most probable category.

Next, we analyze why unlabeled data can improve the generalization accuracy of the classifier on the platform of bipartite graph. Let G_D be the bipartite graph in the all-real-data distribution D . As we know, the components in G_D reflect the concept distribution on the real dataset, and we can achieve completely correct prediction if we get the components in G_D . But it is impossible to achieve this when the dataset D is infinite. The work of co-training model is to find components that are much “similar” to those in G_D by the use of unlabeled data. Given S , as the predictions on the unlabeled data increase, the edges will be added to the bipartite graph and the number of components will

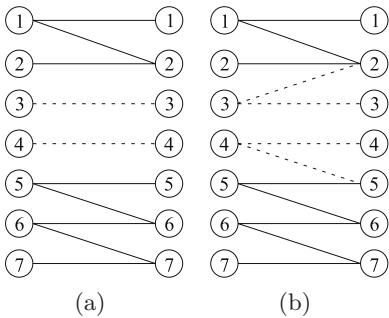


Fig. 2. Graphs G_D and G_S

drop as components merge together. Intuitively, the components in G_S are more similar to the components in G_D , leading to a more accurate prediction on unlabeled data. For example, as demonstrated in Figure 2, vertex 1 and 2 represent labeled instances with the same true labels, vertex 5, 6 and 7 represent labeled instances with the same true labels but different from 1 and 2. On the other hand, vertex 3 and 4 represent unlabeled instances. Before knowing the predicted labels of 3 and 4, there are four components in Figure 2(a). But by introducing the predicted labels of vertex 3 and 4 (for

example, vertex 3 has a predicted label the same as vertex 2, and vertex 4 has a predicted label the same as vertex 5), unlabeled vertex 3 can be connected to vertex 1 and 2 to form a new component, and unlabeled vertex 4 can be connected

to vertex 5, 6 and 7 to form another new component. Then the number of components is reduced to 2, as illustrated in Figure 2(b). Blum and Mitchell [7] had demonstrated that the components in G_S will be similar to those in G_D if the unlabeled set is large enough. Thus, they capture the real concept distribution of the real dataset and give the classifier higher prediction accuracy.

Based on the above analysis, it is clear that the accuracy of the wrapper classifier can be improved by the iteration process of FW-SemiFS. In other words, the effectiveness of FW-SemiFS can be improved by applying unlabeled data.

3 Experiment

In order to evaluate the effectiveness of our proposed algorithm, we have conducted extensive experiments on several datasets. Table 1 summarizes the information of the datasets that we used. In our experiment, the labeled data and unlabeled data are randomly selected from the whole dataset, and the left is used as testing data.

3.1 Experiment Settings and Evaluation Method

In FW-SemiFS experiments, the parameters *startfn*, *fnstep*, *samplingRate*, and *samplingTimes* are set to 5, 6, 50%, and 10, respectively. For comparison, SFFS and Semi-supervised Laplacian Score (called SLS for short) are also conducted. SFFS is a supervised feature selection algorithm described in the previous section; and SLS is a semi-supervised feature selection algorithm which is similar to Zhao and Liu’s algorithm [5], but it uses Laplacian score as its ranking criterion. We chose three machine-learning models, NaiveBayes, NNge, and k-NN, to evaluate the effectiveness of the algorithms. Specifically, NNge is the nearest neighbor like algorithm using non-nested generalized examples; k-NN is k nearest neighbor classifier whose parameter k is set to 5.

Table 1. Dataset summary

Dataset	#Labeled	#Unlabeled	#Testing	#Features	#Classes
German	6	100	294	20	2
Ionosphere	6	100	245	34	2
Mushroom	6	100	294	22	2
Sonar	6	100	102	60	2
Waveform	6	100	294	21	3
wdbc	6	100	463	30	2
ColonTumor	6	30	26	2000	2

For FW-SemiFS and SLS, we employ both the labeled dataset and unlabeled dataset to perform semi-supervised feature selection. But for SFFS approach, we only employ the labeled dataset to perform feature selection. After selecting the feature subset *ResultFS*, we construct the classifier only with the labeled dataset and *ResultFS*, and then the unseen testing dataset is employed to evaluate the classification accuracy.

3.2 Empirical Results

Table 2 shows the experiment results of the SFFS, SLS and FW-SemiFS methods when the size of the selected feature subset is 10, as well as the classification accuracy without feature selection (denoted as “Full” in the table). The numbers are shown in bold when it is the highest one among “Full”, “SFFS”, “SLS” and “SemiFS”. From table 2, we could find that the accuracies of FW-semiFS are higher than that of two other feature selection algorithms in 13 out of 21 cases.

For further view of the algorithm comparison, we conduct statistical analysis of the experiment results. For every dataset and every learning algorithm, we run SFFS, SLS and FW-SemiFS respectively according to 16 different feature subset sizes (from 5 to 20), and then get 16 groups of feature subsets and related accuracies. Each group has three feature subsets and three related classification accuracies according to SFFS, SLS and FW-SemiFS, respectively. After that, we calculate the mean and standard deviation of these 16 accuracies for each group, which are listed in table 3.

From Table 3, we can see that the mean of the classification accuracies of FW-SemiFS are higher than that of two others most of the time. On German

Table 2. The accuracy comparison between SFFS, SLS, and FW-SemiFS

Dataset	NaiveBayes				NNge				k-NN			
	Full	SFFS	SLS	SemiFS	Full	SFFS	SLS	SemiFS	Full	SFFS	SLS	SemiFS
German	52.72	54.08	51.59	55.44	67.01	65.65	58.84	70.41	64.63	63.61	60.20	67.01
Ionosphere	73.47	73.47	67.76	75.51	71.84	71.43	65.04	74.15	72.38	70.61	52.93	73.61
Mushroom	89.80	86.73	91.50	85.71	86.28	83.45	82.54	84.35	65.31	63.95	62.24	69.05
Sonar	52.94	52.61	48.37	59.15	68.63	70.59	70.59	70.59	56.86	61.76	59.80	63.73
Waveform	38.44	46.94	39.80	40.48	61.90	69.39	58.84	69.39	42.52	51.70	51.36	52.72
wdbc	82.72	77.97	87.90	82.51	90.28	92.87	91.79	91.58	92.22	89.85	85.31	87.04
ColonTumor	39.74	76.92	53.85	84.62	37.18	47.44	55.13	35.90	53.85	35.90	58.98	41.03

Table 3. Means and standard deviations of accuracies

Dataset	Method	NaiveBayes		NNge		k-NN	
		Mean	StDev	Mean	StDev	Mean	StDev
German	SFFS	54.49	2.16	66.14	1.43	62.35	4.98
	SLS	52.99	0.89	63.14	5.35	62.86	2.45
	SemiFS	55.93	2.22	69.26	2.77	63.82	3.92
Ionosphere	SFFS	72.32	1.48	72.50	1.08	71.51	1.07
	SLS	68.98	2.78	65.17	3.17	60.50	6.28
	SemiFS	74.64	0.99	74.46	1.00	73.92	0.53
Mushroom	SFFS	88.65	3.17	84.76	0.84	61.31	4.68
	SLS	88.80	2.44	82.41	3.78	63.39	5.53
	SemiFS	87.33	3.28	85.06	0.54	61.78	4.68
Sonar	SFFS	53.29	2.41	71.57	1.13	56.25	4.85
	SLS	50.51	2.27	68.20	2.91	58.09	3.07
	SemiFS	58.58	0.98	70.10	0.72	61.21	4.64
Waveform	SFFS	43.47	3.21	62.48	3.53	50.30	6.74
	SLS	40.18	1.22	60.89	3.65	51.79	3.84
	SemiFS	41.71	3.17	63.58	3.13	52.42	6.06
wdbc	SFFS	79.70	3.24	91.28	0.90	85.12	5.50
	SLS	85.81	3.34	91.20	0.60	86.00	0.73
	SemiFS	82.82	1.11	91.62	0.66	87.39	1.04
ColonTumor	SFFS	52.31	4.34	44.27	2.51	39.06	1.80
	SLS	43.42	3.14	57.78	3.00	61.45	2.89
	SemiFS	54.02	1.99	40.00	1.95	53.85	5.24

and Ionosphere datasets, the means of FW-SemiFS are the highest ones for NaiveBayes, NNge, and k-NN learners, along with small standard deviation; Although, for German dataset, the standard deviation is larger than that of two other algorithm, the mean of FW-SemiFS is much higher than that of two others. The similar phenomena can be also observed for Sonar and wdbc datasets.

4 Conclusion and Future Work

We have explored the use of unlabeled examples to facilitate “wrapper-type” forward semi-supervised feature selection. The proposed algorithm works in an iterative procedure. In each step, unlabeled examples receive labels from the classifier constructed on currently selected feature subset. Then a random sample of now “labeled” unlabeled examples is concatenated with the training set to form a “joint” dataset, where a wrapper-based feature selection is performed. Experiment results show that the proposed approach, generally, can obtain 1% to 10% higher accuracy than other supervised and semi-supervised feature selection algorithms.

Future Work. The work discussed in this paper represents techniques based on random selection mechanism. In the future, we plan to extend the techniques by using prediction confidence as a criterion to select those unlabeled data which is most probable to have correctly predicted labels.

References

1. Liu, H., Yu, L.: Toward Integrating Feature Selection Algorithms for Classification and Clustering. *IEEE Transactions on Knowledge and Data Engineering* 17(4), 491–502 (2005)
2. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. *The Journal of Machine Learning Research* 3, 1157–1182 (2003)
3. Seeger, M.: Learning with labeled and unlabeled data. Technical report (2000)
4. Zhu, X.: Semi-Supervised Learning Literature Survey. Computer Sciences Technical Report 1530, University of Wisconsin-Madison (2005)
5. Zhao, Z., Liu, H.: Semi-supervised Feature Selection via Spectral Analysis. In: *SIAM International Conference on Data Mining (SDM 2007)* (2007)
6. Zhao, Z., Liu, H.: Spectral Feature Selection for Supervised and Unsupervised Learning. In: *Proceeding of the 24th International Conference on Machine Learning (ICML 2007)* (2007)
7. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: *Proceedings of the 11th Annual Conference on Computational Learning Theory (COLT 1998)*, pp. 92–100 (1998)

Automatic Extraction of Basis Expressions That Indicate Economic Trends

Hiroki Sakaji, Hiroyuki Sakai, and Shigeru Masuyama

Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku-cho, Toyohashi-shi,
Aichi 441-8580, Japan

(sakaji,sakai,masuyama)smlab.tutkie.tut.ac.jp

Abstract. This paper proposes a method to automatically extract basis expressions that indicate economic trends from newspaper articles by using a statistical method. We also propose a method to classify them into positive expressions that indicate upbeat, and negative expressions that indicate downturn in economy, respectively. It is important for companies, governments and investors to predict economic trends in order to forecast revenue, sales of products, prices of commodities and stock prices. We considered that basis expressions are useful for the companies, governments and investors to forecast economic trends. We extracted basis expressions, and classified them into positive expressions or negative expressions as information to forecast economic trends. Our method used a bootstrap method that was minimally a supervised algorithm for extracting basis expressions. Moreover, our method classified basis expressions into positive expressions or negative ones without dictionaries.

1 Introduction

It is important for companies, governments and investors to predict the economic trends in order to forecast revenue, sales of products, prices of commodities and stock prices. The diffusion index^[1] is one of indices concerning economic trends, and is computed every three months, and provides economic trends during prior period. However, it is difficult to forecast the business performance accurately by using diffusion indices, as it can not indicate current economic trends.

These indices are computed using numeric data. However, some qualitative language data that reflect economic trends may not be quantified straightforwardly. For example, an opinion “*Economy seems to recover*” in a newspaper article is hard to be quantified, as “*Economy seems to recover*” is a sense of the writer.

Nakajima et al. [2] proposed a method for extracting articles concerning economic trends from newspaper articles and classifying them into positive articles

¹ The diffusion index is a summary measures designed to facilitate the analysis and forecast of business cycles by combining the behavior of a group of economic indicators that represent widely differing activities of the economy, such as production and employment, and that correspond closely to turning points.

<http://www.esri.cao.go.jp/en/stat/di/di2e.html>

that indicate upbeat in economy and negative ones that indicate downturn in economy. However, Nakajima's method can not classify articles having two different opinions. For example, an article that indicates economy in Aichi prefecture is upbeat while that in Gifu prefecture is downturn, includes two different opinions about different areas, and can not be treated by Nakajima's method.

We propose a method to extract basis expressions that indicate economic trends from newspaper articles concerning economic trends and to classify basis expressions into positive or negative expressions. We considered that opinions concerning economic trends can be extracted by using basis expressions, which enable us to distinguish two different types of opinions in the same articles. Our method used a bootstrap method that was minimally supervised algorithm for extracting basis expressions. Moreover, our method classified basis expressions into positive expressions or negative ones without dictionaries.

2 Related Work

As related work for extracting phrases that have a particular meaning, Kanayama et al. proposed a method for extracting a set of sentiment units by using transfer-based machine translation engine replacing the translation patterns with sentiment patterns[5]. However, to construct a complete list of complex rules or patterns manually, which is the case of the above methods, is a time-consuming and costly task. In contrast, our method uses statistical information and only one initial clue phrase as an initial input. The domain-specific dictionaries, predetermined patterns, complex rules made by hand are not needed.

Wilson et al. proposed a method for determining whether an expression is neutral or polar[6]. In their research, the expressions are extracted manually and the method needs dictionaries. In contrast, our method automatically extracts expressions and does not need dictionaries.

Sakai et al. proposed a method for extracting cause information from Japanese financial articles concerning business performance[3]. Their work is probably most closely related to ours. However, our method extracts basis expression concerning not performance of each company but economic trends. Moreover, our method also classifies basis expressions into positive and negative ones.

3 Extraction of Basis Expressions

As a preprocessing, our method extracts articles concerning economic trends from newspaper corpus by using Support Vector Machine(SVM)[4]. We applied a method proposed by sakai et al.[2] for extracting them. As a result, 10,027 newspaper articles concerning economic trends were extracted from Nikkei newspapers published from 1990 to 2005.

Here, a basis expression is a part of a sentence consisting of some "*bunsetu*'s" (a *bunsetu* is a basic block in Japanese composed of several words). Our method extracts basis expressions by using clue phrases, i.e. phrases frequently modified by basis expressions. For example, a basis expression frequently modifies clue

phrase "(*no eikyou*: influenced by)" in Japanese. Our method extracts an expression that consists of a clue phrase and a phrase that modifies it as a basis expression. Hence, if many clue phrases effective for extracting basis expressions are acquirable, basis expressions are extracted automatically. However, it is hard to acquire sufficient clue phrases effective for extracting basis expressions manually. Hence, our method also acquires such clue phrases automatically from a set of articles concerning economic trends.

Our method for extracting basis expressions is as follows.

- Step 1:** Input an initial clue phrase "(*no eikyou*: influenced by)" and acquire phrases that modify them.
- Step 2:** Extract phrases appearing frequently in a set of the phrases acquired in Step 1 (e.g. (*sekai keizai*: world economy)). In this paper, such a phrase extracted in Step 2 is defined as a "frequent phrase".
- Step 3:** Acquire new clue phrases modified by the frequent phrases.
- Step 4:** Extract new frequent phrases from a set of phrases that modify the new clue phrases acquired in Step 3. This step is the same as Step 2.
- Step 5:** Repeat Steps 3 and 4 until they are executed predetermined times or neither new clue phrases nor new frequent phrases are extracted.
- Step 6:** Extract basis expressions by using extracted frequent phrases and acquired clue phrases.

3.1 Extraction of Frequent Phrases

The method for extracting "frequent phrases" from a set of phrases that modify clue phrases is described below.

- Step 1:** Acquire a *bunsetu* modifying a clue phrase and eliminate a case particle from the *bunsetu*. Here, the *bunsetu* is denoted by c .
- Step 2:** Acquire frequent phrase candidates by adding *bunsetu* modifying c to c . (See Figure 1)
- Step 3:** Calculate score $S_f(e, c)$ of frequent phrase candidate e containing c by the following Formula 1.
- Step 4:** Adopt e assigned the best score $S_f(e, c)$ among the set of frequent phrase candidates containing c as a frequent phrase.

Score $S_f(e, c)$ is calculated by the following Formula 1.

$$S_f(e, c) = -f_e(e, c)f_p(e) \log_2 P(e, c), \quad (1)$$

where $P(e, c)$ is the probability that frequent phrase candidate e containing c appears in the set of articles concerning economic trends. $f_e(e, c)$ is the number of frequent phrase candidate e 's containing c in the set of articles concerning economic trends. $f_p(e)$ is the number of *bunsetu*'s that compose e . $P(e, c)$ is calculated by the following Formula 2.

$$P(e, c) = \frac{f_e(e, c)}{N_e(c)}, \quad (2)$$

where $N_e(c)$ is the total number of frequent phrase candidates containing c in the set of articles concerning economic trends.

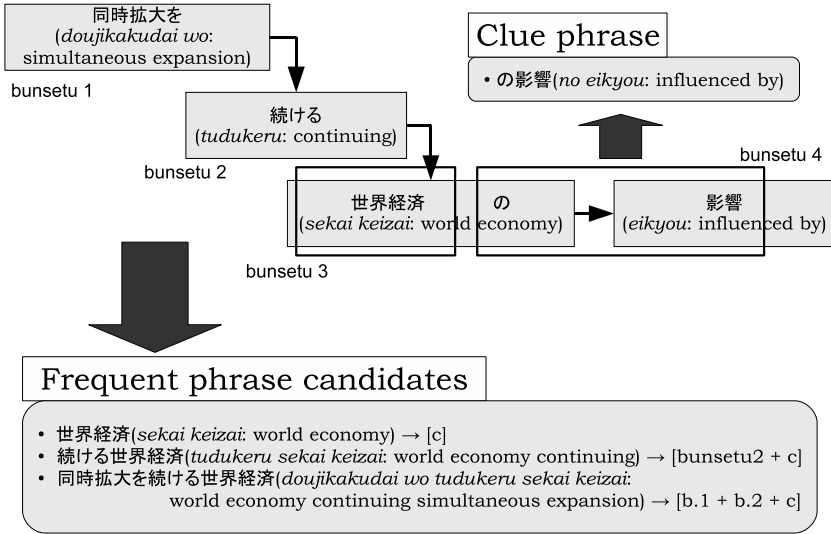


Fig. 1. Examples of frequent phrase candidates

3.2 Selection of Frequent Phrases

The frequent phrases extracted from a set of phrases that modify clue phrases may contain inappropriate ones. Hence, our method selects appropriate frequent phrases from them. Here, our method calculates entropy $H(e)$ based on $P(e, s)$ and selects frequent phrases assigned entropy $H(e)$ larger than a threshold value calculated by Formula 5. $P(e, s)$ is the probability that frequent phrase e modifies clue phrase s . Entropy $H(e)$ is used for reflecting “variety of clue phrases modified by frequent phrase e ”. If entropy $H(e)$ is large, frequent phrase e modifies various kinds of clue phrases and such a frequent phrase is an appropriate frequent phrase. Entropy $H(e)$ is calculated by the following Formula 3.

$$H(e) = - \sum_{s \in S(e)} P(e, s) \log_2 P(e, s), \tag{3}$$

where

$$P(e, s) = \frac{f(e, s)}{\sum_{s' \in S(e)} f(e, s')}. \tag{4}$$

Here, $S(e)$ is the set of clue phrases modified by frequent phrase e . $f(e, s)$ is the number of frequent phrase e 's that modifies clue phrase s in the set of articles concerning economic trends. The threshold value is calculated by the following Formula 5.

$$T_e = \alpha \log_2 |N_s|, \tag{5}$$

where N_s is the set of clue phrases used for extracting frequent phrases and α is a constant ($0 < \alpha < 1$).

3.3 Acquisition of Clue Phrases

The method for acquiring new clue phrases from frequent phrases is as follows.

Step 1: Extract a *bunsetsu* modified by frequent phrase e .

Step 2: Acquire clue phrase s by adding a case particle contained in the frequent phrase e to the *bunsetsu*.

Step 3: Calculate entropy $H(s)$ based on the probability $P(s, e)$ that clue phrase s is modified by frequent phrase e .

Step 4: Select clue phrase s assigned entropy $H(s)$ larger than a threshold value calculated by Formula 7.

Here, entropy $H(s)$ is introduced for selecting appropriate clue phrases and is calculated by the following Formula 6.

$$H(s) = - \sum_{e \in E(s)} P(s, e) \log_2 P(s, e), \quad P(s, e) = \frac{f(s, e)}{\sum_{e' \in E(s)} f(s, e')}. \quad (6)$$

Here, $E(s)$ is the set of frequent phrases that modify clue phrase s , and $f(s, e)$ is the number of clue phrase s 's modified by frequent phrase e in the set of articles concerning economic trends. The threshold value is calculated by the following Formula 7.

$$T_s = \alpha \log_2 |N_e|. \quad (7)$$

Here, N_e is the set of frequent phrases used for extracting clue phrases. α is the same constant that in Formula 5.

3.4 Extraction of Basis Expressions by Using Frequent Phrases and Clue Phrases

Finally, our method extracts basis expressions by using frequent phrases and clue phrases. A basis expression consists of a phrase that modifies the clue phrase. Moreover, the phrase that modifies the clue phrase contains some frequent phrases. For example, “(*yusyutu no gennsyou wo haikai ni*: under decreasing export)” is a basis expression since phrase “(*yusyutu no gennsyou*: decreasing export)” modifies clue phrase “(*wo haikai ni*: under)” and the phrase contains frequent phrase “(*gennsyou*: decreasing)”.

4 Classification of Basis Expressions

Our method classifies extracted basis expressions into positive expressions and negative expressions. However, extracted basis expressions contain some of inappropriate basis expressions. As a result, our method extracted basis expressions into positive expressions, negative expressions and other expressions. Other expressions are extracted basis expressions that are neither positive nor negative expressions.

For example, “(*doujidakudai wo tudokeru sekaikeizai*: world economy continuing simultaneous expansion)” is a positive expression. Thus positive expressions indicate that Japanese economy is upbeat. For example, “(*setubitoushi ya kojinsyouhi no donka*: slowdown of business investment and personal consumption)” is a negative expression. Thus negative expressions indicate that Japanese economy is downturn. For example, “(*tyousataisyouhenkou*: change of objective for survey)” and “(*keiki no nobinayami*: stagnation of economy)” are other expressions. We define expressions that cite Japanese economy are inappropriate as basis expressions, because our goal is extraction of basis expressions.

We develop two classifiers by using one-versus-rest method and Support Vector Machine(SVM)^[4]. The one classifies extracted basis expressions into positive expressions and the others. Here, a positive expression is defined as a *correct expression* and the other is defined as an *incorrect expression*. The other one classifies extracted basis expressions into negative expressions and the others. Here, negative expression is defined as a *correct expression* and the other is defined as an *incorrect expression*. The classifiers use *character N-gram* and *word N-gram* as features.

5 Evaluation

In this section, we evaluate our method. Our method extracted basis expressions from 10,027 newspaper articles concerning economic trends and classify them into positive and negative.

First, we evaluated our method for extracting basis expressions. We employ CaboCha^[2] as a Japanese parser. We manually extracted 75 basis expressions from 100 articles concerning economic trends performance as a correct data set. Moreover, we extracted basis expressions by our method from the same 100 articles and calculated precision and recall. Here, a basis expression extracted by our method is correct if it contains a basis expression extracted as the correct data set. The precision, recall and F-measure^[3] calculated by the following formulas.

$$Precision = \frac{|Sb \cap Ab|}{|Sb \cap Nb|}, \quad Recall = \frac{|Sb \cap Ab|}{|Ab|},$$

where Sb is the set of basis expressions extracted by our method from 100 articles concerning economic trends. Ab is the set of basis expressions contained in the correct data set. Nb is the set of expressions modifying clue phrases in the 100 articles concerning economic trends. The results are shown in Tabel ^[1].

Next, we evaluated our method for classifying basis expressions. We employ ChaSen^[4] as a Japanese morphological analyzer, and SVM^{light} ^[5] as an implementation of SVM. We extracted 1620 basis expressions by our method with α 0.6 and iteration count 3. 1620 basis expressions were manually annotated with

² <http://chasen.org/~taku/software/cabochoa/>

³ $F - measure = (2 \times Precision \times Recall) / (Precision + Recall)$.

⁴ <http://chasen.aist-nara.ac.jp/hiki/ChaSen/>

⁵ <http://svmlight.joachims.org>

Table 1. Precision, Recall and F-measure of basis expressions extraction

α	Precision	Recall	F-measure	num. of basis expression
0.9	1.000	0.160	0.276	650
0.6	0.714	0.333	0.455	1620
0.5	0.042	0.573	0.078	49293

Table 2. Precision, recall and F-measure of basis expression classification with *character N-gram feature*

	num. of frequent features	Precision	Recall	F-measure
Positive	9021	0.800	0.615	0.695
Negative	9021	0.843	0.855	0.849

“positive”, “negative” or “others”. The annotated basis expressions were divided into two sets. The first (1120 expressions) were a training data, used for feature selection and modeling. We used the second set (500 expressions) as a test data set. We calculated precision, recall and F-measure from the test data set. The precision and recall are calculated by the following formulas.

$$Precision = \frac{|E \cap C|}{|E|}, \quad Recall = \frac{|E \cap C|}{|C|}.$$

Here, E is the set of basis expressions annotated with *correct expressions* in the test data set. C is the set of *correct expressions* contained in the test data set. The results are shown in Tables 2 and 3.

6 Discussions

In Table 1, precision rises from α 0.5 to 0.6, while recall drops. When low α value is assigned, inappropriate clue phrases and frequent phrases were found in a set of extracted clue phrases and extracted frequent phrases. Furthermore, new inappropriate ones are extracted by extracted inappropriate phrases. As a result, our method acquires many inappropriate ones. This happens when α is between 0.5 and 0.6.

In Tables 2 and 3, focusing on recall, it is interesting to note that *negative* classifiers perform better than *positive* ones. This is due to the fact that 1620

Table 3. Precision, recall and F-measure of basis expression classification with *word N-gram feature*

	num. of frequent features	Precision	Recall	F-measure
Positive	6450	0.769	0.545	0.638
Negative	6450	0.833	0.867	0.845

basis expressions contain negative expressions far more than positive ones. We consider that cause for few positive expressions is due to recession in Japan during the period of the corpus.

In Tables 2 and 3, focusing on precision, *character N-gram feature* classifier is the highest of all. This is due to the fact that characters play a key role in classifying expressions in Japanese, because Chinese characters that are one of the Japanese characters have meanings.

7 Conclusion

We proposed a method for extracting basis expressions that indicate economic trends from Japanese newspaper articles concerning economic trends. First, our method extracts basis expressions from them by using statistical information and initial clue phrases. Next, our method classifies basis expressions into *positive expressions*, *negative expressions* and *other expressions*. This method can also be applied to other tasks such as extracting reputations for specific items.

Acknowledgment

This work was supported in part by Global COE Program “Frontiers of Intelligent Sensing”, MEXT, Japan.

References

1. Nakajima, T., Sakai, H., Masuyama, S.: A classification method based on the view of the author of each newspaper article on economics. IPSJ SIG Notes 2003(51)(20030522), 175–180 (2003) (in Japanese)
2. Sakai, H., Umemura, S., Masuyama, S.: Extraction of Expressions concerning Accident Cause contained in Articles on Traffic Accidents. Journal of Natural Language Processing 13(4), 99–124 (2006) (in Japanese)
3. Sakai, H., Masuyama, S.: Extraction of Cause Information from Newspaper Articles Concerning Business Performance. In: Proc. of the 4th IFIP Conference on Artificial Intelligence Applications & Innovations (AIAI 2007), pp. 205–212 (2007)
4. Vapnik, V.N.: Statistical Learning Theory. John Wiley & Sons, Chichester (1999)
5. Kanayama, H., Nasukawa, T., Watanabe, H.: Deeper sentiment analysis using machine translation technology. In: Proceedings of the 20th COLING, pp. 494–500 (2004)
6. Wilson, T., Wiebe, J., Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. In: Proceedings of HLT/EMNLP-2005, pp. 347–354 (2005)

A New Framework for Taxonomy Discovery from Text

Ahmad El Sayed¹, Hakim Hacid², and Djamel Zighed¹

¹ University of Lyon 2
Bron 69676, France

{asayed, dzighed}@eric.univ-lyon2.fr

² University of New South Wales
Sydney NSW 2052, Australia
hakimh@cse.unsw.edu.au

Abstract. Ontology learning from text is considered as an appealing and a challenging approach to address the shortcomings of the hand-crafted ontologies. In this paper, we present OLEA, a new framework for ontology learning from text. The proposal is a hybrid approach combining the pattern-based and the distributional approaches. It addresses key issues in the area of ontology learning: low recall of the pattern-based approach, low precision of the distributional approach, and finally ontology evolution. Preliminary experiments performed at each stage of the learning process show the pros and cons of the proposal.

1 Introduction

Given the many difficulties related to the encoding of “semantic” ontologies today, an appealing and challenging approach is to build such ontologies automatically from wealthy resources like texts. This led to the emergence of the field of *ontology learning from text* [6]. In this paper, we present OLEA (Ontology LEARNING), a new framework for ontology learning from text [1]. The general architecture of OLEA is illustrated in Figure 1. The proposal is a hybrid approach that aims to deal with key issues in the area:

On Low Recall of the Pattern-Based Approach. The pattern-based approach [5], though yielding “acceptable” precision, suffers from very low recall since detecting relations depends on the appearance of a set of *rigid* lexicosyntactic patterns (e.g., *NP such as {NP, NP..}*). Our framework deals with this drawback, and proposes a technique able to capture and match more “flexible” patterns in text.

On Low Precision of the Distributional Approach. This approach consisting mainly of clustering terms basing on their similarities, lacks generally

¹ The accomplished work concerns though the concepts and the concepts hierarchies learning, so we will rather refer to the task as *taxonomy learning*.

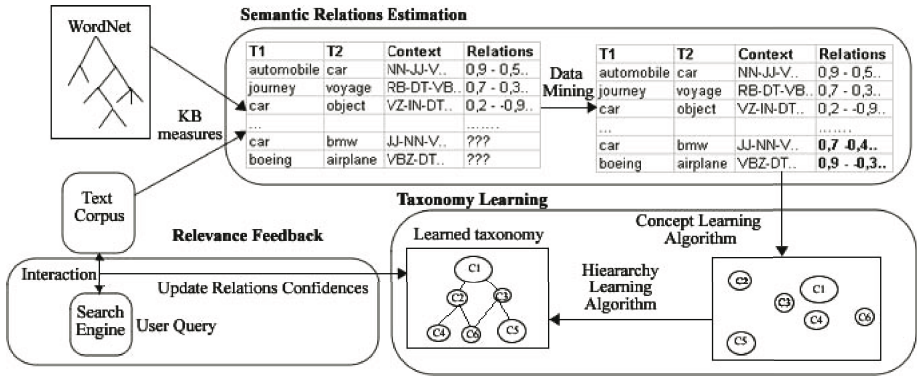


Fig. 1. OLea: General Architecture

from low precision. This is due to two main reasons: (1) The commonly used hierarchical methods are not quiet adaptive [24] since they provide binary trees of crisp clusters. (2) Methods lack of reliability since they rely, in most cases, on a single semantic relation (e.g., synonymy). That is, we present a learning procedure involving more semantic relations, and thus supplying us with more reliable decisions while building the concepts hierarchy.

On Ontology Evolution. It is known that an ontology should be subject of continuous refinements in order to adapt it to new users’ requirements. However, existing approaches either ignore this issue, or require regular human interventions, which is a tedious task [3,7]. That is, we propose a preliminary approach that places the learned taxonomy at the core of a search engine, in order to adapt the taxonomy to users’ vision over text, without any manual effort.

2 Estimating Semantic Relations

The overall technique for estimating relations more “flexibly” is described as follows. Each pair of terms occurring in a corpus is represented by a set of lexico-syntactic features. Pairs that could be matched in WordNet will be augmented by confidence rates for each of their semantic relation. This will construct the learning base that will serve to predict the semantic relation rates between pairs uncovered by WordNet.

Calculating Relations between Concepts. For pairs of terms that could be matched in WordNet (concepts), we calculate a confidence rate for each of their semantic relations basing on the semantic structure of the taxonomy. What we are seeking at the end, is statements assessing, for instance, that “object” and “car” are 0.1-synonyms, 0.8-hypernyms, and 0-meronyms. The calculation of such rates depends on the target relation. While hypernymy confidence relies on the edges count along the shortest path separating two concepts, confidences for antonymy and meronymy are boolean, depending simply on the

presence/absence of such relations. Synonymy relations are calculated by means of our semantic distance measure proposed in [10].

Mining Relations between Terms. The obtained rates from the previous step are used as a “reference” for predicting semantic relations between the uncovered pairs in WordNet[2]. The assumption is that terms pairs appearing in similar contexts tend to have similar semantic relations. Relations’ confidence rates for an uncovered pair P are calculated by means of the confidence rates of its K Nearest Neighbors (KNN). Each context is characterized by a set of lexico-syntactic features (e.g., head word, partial path, path length). In order to compare two contexts, distances between the different features can be either a simple integer/string comparison, or based on the Waterman alignment algorithm [11] (for path features).

Consider a relation r for a pair P . Finding the best K confidence rates depends on how much we can “optimize” the distance between a pair of contexts. These distances can be optimized when reaching a maximal correlation with distances between pairs of semantic relations (response variables). That is, we applied a multiple linear regression model to find the coefficients (weights) that optimize the correlation between them. Then, we apply the optimal coefficients on the previous equation in order to find more accurately the KNN. A relation’s confidence is finally calculated by means of the weighted average of the K -nearest relations confidences.

Evaluation and Results. Our experiments was carried out on a benchmark composed of 1000 documents picked from the Reuters corpus[3] along with the WordNet taxonomy. The goal is to check how far can semantic relations between terms approach the “gold standard” semantic relations between concepts. For this, we divided the set of concept pairs into 80% for the training set, and 20% for the test set. Obtained results illustrated in Figures 2[3] show that K has no significant effect on performance, depending more on the obtained R^2 . Without using the linear model we obtained a best correlation ratio of 0.32 for synonymy. However, when incorporating the regression model with KNN, we could dramatically increase correlation, attaining an interesting rate of 0.82 for synonymy.

3 Taxonomy Learning

In this section, we present a two-phases procedure that takes as an input the semantic relations rates, and provides as an output a hierarchy of concepts. It includes concepts learning, and concepts hierarchy learning.

Concepts Learning. The goal here is to group terms into a set of sense-bearing units which will be regarded as concepts. Hence, we define a soft hierarchical-based clustering algorithm able to deal with polysemous words (see Algorithm

² Each semantic relation is treated separately.

³ Reuters corpus, volume 1, English language, release date: 2000-11-03.

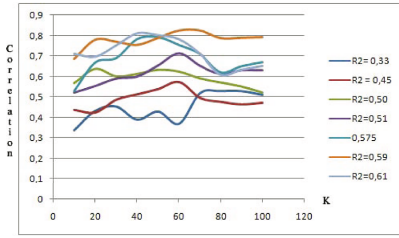


Fig. 2. Effect of K variation on the final correlation rate for each regression model

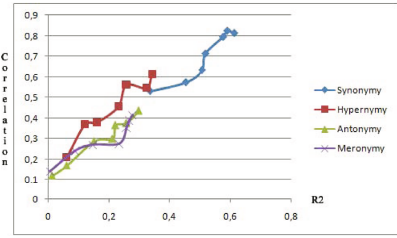


Fig. 3. Effect of R^2 on the final correlation rate for each semantic relation

□4. Rather than clustering terms by relying solely on semantic similarity which is error-prone, our algorithm offers more reliable decisions by taking into account a larger set of relations. The point is that two related clusters will be merged only if they are found “purely” synonyms, therefore do not have any other relation with a confidence rate greater than a specified threshold.

Algorithm 1. Concept Learning Process

Require: parameters θ_1 and θ_2

- 1: Initialize each term t_i in the set T as a cluster c_i in the set C
- 2: **repeat**
- 3: Identify the closest pair P of clusters c_p and c_k in C having synonymy exceeding a threshold θ_1 and having no other relation exceeding a threshold θ_2
- 4: Create a new cluster c_n containing the instances of c_p and c_k
- 5: **if** c_p is not basically a term **then**
- 6: Remove c_p from C
- 7: **end if**
- 8: **if** c_k is not basically a term **then**
- 9: Remove c_k from C
- 10: **end if**
- 11: **for all** other clusters oc in C **do**
- 12: **if** $oc_i \subset c_n$ OR $c_n \subset oc_i$ **then**
- 13: continue
- 14: **end if**
- 15: Compute relations R_i between c_n and oc_i
- 16: **if** $R_i(\text{synonymy})$ is above θ_1 and all other R_i are below θ_2 **then**
- 17: Merge oc_i instances in c_n
- 18: **if** oc_i is not basically a term **then**
- 19: Remove oc_i from C
- 20: **end if**
- 21: **end if**
- 22: **end for**
- 23: Mark c_p and c_k as a considered pair (to not be considered again)
- 24: **until** P is empty
- 25: **return** the set of created clusters

Concepts Hierarchy Learning. Following concepts learning, we present Algorithm 2 which aims to learn taxonomic *is-a* relations. As hypernyms occur rarely between pairs of terms, lot of concepts will remain unlinked. To overcome

⁴ An experimental comparison between different soft algorithms is out of scope of this paper.

this shortcoming, we defined a measure that aims at finding the most appropriate place for an unlinked concept in a given hierarchy.

At the end of this phase, we obtain a fuzzy taxonomy in the sense that related terms within a concept are assigned a synonymy confidence between each others, and that concepts are related to each others by an 'is-a' relation being assigned a hypernymy confidence as well.

Algorithm 2. Taxonomic-Relations Learning Process

Require: parameters α_1 and α_2

- 1: Let P be the set of concepts pairs with their relations confidence obtained from the previous phase
- 2: Define direct-hypernymy confidence $dirhyp(cp_i)$ for a concept pair cp_i in P as $synonymy(cp_i) * hypernymy(cp_i)$
- 3: **repeat**
- 4: Identify the concept pair cp_k with the highest $absolute(dirhyp(cp_k))$ that must be above a threshold α_1 and having the other relations confidence (except synonymy and hypernymy) below a threshold α_2
- 5: Create a hypernymy link for cp_k
- 6: **until** cp_k is empty
- 7: **for** each remaining concept c_r sharing no link with any other concept **do**
- 8: find the K closest concepts for c_r by means of synonymy
- 9: **for** each close concept c_i **do**
- 10: calculate a score s_i as a function of $synonymy(c_r, c_i)$ and $synonymy(c_r, hypernyms(c_i))$
- 11: **end for**
- 12: create a hypernym link between c_r and the c_i with the highest score $MAX(c_i)$
- 13: **end for**

Evaluation and Results. Actually, ontology learning community lacks common frameworks for evaluation and comparison. Concerning our work, we performed a preliminary evaluation against a “reference” taxonomy. Typically, after specifying the actual context of *newspapers*, we asked a human subject to group and organize in one or many trees a set of 50 terms. Finally, we compare the human-made tree with our learned tree in terms of precision and recall by considering the number of correct *vs* incorrect learned relations. Concerning concepts learning, since precision and recall depends on θ_1 and θ_2 , we altered θ_1 in the range of [0.88, 0.97], while fixing θ_2 to 0.05. As we can see in Figure 4, while precision tends to drop dramatically when reducing θ_1 , recall tends to be somehow stable. Concerning taxonomic-relations learning, we consider the number of correct *vs* incorrect links between validated concepts by the user. We fixed the parameter θ_1 at 0.95, since it gave the optimal trade-off between precision and recall. Then, we applied Algorithm 2 by altering α_1 in the interval [0.1, 0.2] and fixing α_2 at 0.05. Results illustrated in Figure 5 show interesting performance, especially from a recall point of view.

4 Improving Taxonomy with Relevance Feedback

Involving human subjects in the learning process, although extremely benefic, can be a very tedious and time-consuming task [3,7]. What we propose here is to add supervision to the learning process without any manual effort: Since our taxonomy seeks essentially to integrate a IR environment, we placed our learned

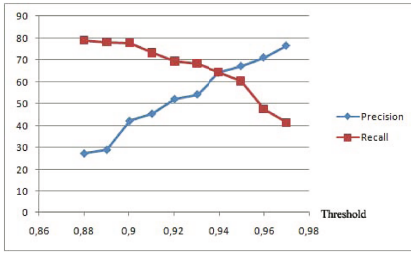


Fig. 4. Concept learning performance in terms of precision and recall with different parameter values

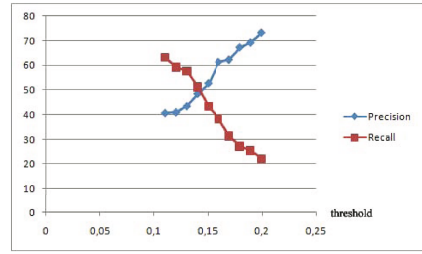


Fig. 5. Taxonomic-relations learning performance in terms of precision and recall with different parameter values

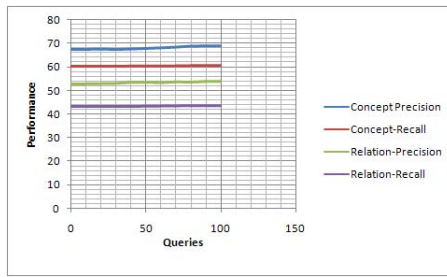


Fig. 6. Performance evolution along queries using Relevance Feedback

taxonomy at the core of our IR system [9]. Keywords queries will be expanded to other related terms by means of the synonymy and hypernymy relations. Then, users interactions with the system are taken into account to update the taxonomy by means of a relevance feedback mechanism [8]. For instance, given a query term q expanded with another term t . As a respond, a document d was presented to the user by its outline o . if o contains both q and t , and d was clicked by the user, the relation between q and t will be strengthen by a parameter β that we define. Such feedbacks will enable the system to take more subjective decisions about accepting or rejecting a specific expansion term in future queries.

Evaluation and Results. To evaluate the effect of relevance feedback on taxonomy learning, we took as a starting point the results given by the learned taxonomy obtained using the optimal parameters. Next, 100 keywords queries (related to the selected hierarchy for evaluation) are sent consecutively to the system. At the end of session of each query, clicked and unclicked documents are considered for the feedback. Taxonomy is updated at the end of each set of 20 queries in order to be reevaluated against the hand-built taxonomy. Figure 6 shows the precision and recall values for both concepts and relations learning along the 100 queries. We can notice the slight but sure improvement in the final results (especially in precision). Yet, we argue that the improvement can be seen more clearly with larger set of queries.

5 Conclusion and Future Works

To wrap up, we presented in this paper OLea, a framework for learning ontology from a text corpus. It has the advantage of addressing the main drawbacks of the pattern-based and distributional approaches. However, a comparison with other methods is still needed to assess the added-value of our proposal. This is not an easy task though. We argue that a better evaluation is task-oriented. That is, we are intending to perform other evaluations in environments like Information Retrieval and Document Clustering.

References

1. Caraballo, S.A.: Automatic construction of a hypernym-labeled noun hierarchy from text. In: ACL (1999)
2. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. *J. Artif. Intell. Res (JAIR)* 24, 305–339 (2005)
3. Faure, D., Poibeau, T.: First experiments of using semantic knowledge learned by asium for information extraction task using intex (2000)
4. Grefenstette, G.: Explorations in automatic thesaurus construction. Kluwer, Dordrecht (1994)
5. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. Number S2K-92-09, 8 (1992)
6. Maedche, A., Staab, S.: Ontology learning for the semantic web. *IEEE Intelligent Systems* 16(2), 72–79 (2001)
7. Moldovan, D.I., Girju, R.: An interactive tool for the rapid development of knowledge bases, 65–86 (2001)
8. Ruthven, I., Lalmas, M.: A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev.* 18(2), 95–145 (2003)
9. Sayed, A.E., Hacid, H., Zighed, D.: Combining text and image for content-based information retrieval. In: Proceedings of the International Conference on Information and Knowledge Engineering IKE 2007 (2007)
10. Sayed, A.E., Hacid, H., Zighed, D.: A multisource context-dependent approach for semantic distance between concepts. In: Wagner, R., Revell, N., Pernul, G. (eds.) DEXA 2007. LNCS, vol. 4653, pp. 54–63. Springer, Heidelberg (2007)
11. Smith, T., Waterman, M.: Identification of common molecular subsequences 195–197 (1981)

R-Map: Mapping Categorical Data for Clustering and Visualization Based on Reference Sets

Zhi-Yong Shen^{1,*}, Jun Sun^{1,*}, Yi-Dong Shen¹, and Ming Li²

¹ Lab. of Computer Science, Institute of Software
Chinese Academy of Sciences, China

{zyshen, junsun, ydshen}@ios.ac.cn

² Department of Epidemiology
Michigan State University, USA

liming@msu.edu

Abstract. In this paper, we propose a framework that maps categorical data into a numerical data space via a reference set, aiming to make the existing numerical clustering algorithms directly applicable on the generated image data set as well as to visualize the data. Using statistics theories, we analyze our framework and give the conditions under which the data mapping is efficient and yet preserves a flexible property of the original data, i.e. the data points within the same cluster are more similar. The algorithm is simple and has good effectiveness under some conditions. The experimental evaluation on numerous categorical data sets shows that it not only outperforms the related data mapping approaches but also beats some categorical clustering algorithms in terms of effectiveness and efficiency.

Keywords: Clustering, Data mapping, Categorical data.

1 Introduction

Clustering is to partition data points into groups, which makes data points in the same group more similar to one another than to those out of the group. Clustering algorithms are categorized by the data types they fit for. There are roughly two data types including numerical and categorical, and a data set may have mixed types. We take the internet traffic data as an example: protocols and operation systems are categorical data, while packet sizes and packet numbers are numerical data. Researchers have designed various clustering algorithms for these two data types. The categorical clustering algorithms are mostly based on pair-wise similarities or information theory etc. such as those proposed in [4,9], most of which suffer from quadratic complexity or combinational explosion problems. Clustering algorithms for numerical data [5,6] are quite different from those for categorical data. In numerical data space, coordinates based linear reference frame is available so that many concepts, like means, and mean square distance etc., could be defined to compress the numerical data and make the clustering algorithms more efficient. Moreover, data visualization, which helps to understand the data intuitively, is easier to be implemented in the numerical space than in the categorical space.

* Graduated University, Chinese Academy of Sciences.

Some applications such as internet traffic data based intrusion detection need an efficient clustering strategy which can handle the mixed data types and integrated with data visualization. Intuitively, mapping the categorical data sets into numerical ones can cover these requirements, since the mixed-typed data are naturally turned into numerical data when their categorical parts become numerical ones. It can also employ some efficient numerical clustering algorithms such as K-means and can be easily visualized. There're some traditional approaches [8,10] that can map data into a new space based on the pair-wise similarity matrix. The most important issue of data mapping is what data property has been preserved during the mapping. Traditional techniques concern different data properties such as the distance between each pair of data points [10] or the neighborhood of each data point [8]. These algorithms employ complex processes on the similarity matrix to preserve these strict properties.

We hereby propose a model to establish a data mapping for clustering and visualizing only for categorical data since in the proposed strategy handling the mixed-type data is essentially handling the categorical data. The mapping is based on calculating similarities between data points and a sample set called reference set. Mapping data via calculating similarity is not new and nor is the concept of reference points. Our most significant contribution is that we propose to preserve a more flexible data property specific to the goal of clustering, i.e. the expectation of intra-cluster similarities is higher than that of inter-cluster similarities. We find that under some conditions, which are easily to satisfy in practice, this property can be preserved by directly treating the similarities with points in the reference set as the image data. This simplicity makes our mapping algorithm efficient and the preservation of the clustering property makes it effective for clustering categorical data. Because of the existence of the reference set, we'll call this mapping framework *R-map*.

1.1 Related Works

Mapping data into a new numerical data space by calculating a similarity matrix¹ is not a new approach. Traditional techniques include Multidimensional Scaling (MDS) [1] and Locally Linear Embedding (LLE) [8] among which Landmark MDS (LMDS) [10] is most closed to the model proposed in this paper. The input of MDS is a distance matrix between objects. MDS produces a coordinate vector for each object in a new numerical space whose dimension is user-specified. This process is called *embedding*. The goal of MDS is to minimize a quadratic cost function between the new embedding coordinates and the original coordinates that created the distance matrix. Visualization and clustering can be performed on the resulting embedding coordinates. The original MDS technique is not appropriate for large data sets nowadays. The author of [7] compared some scalable MDS approaches and the experimental results show that is more accurate than the others with roughly the same computational cost. LMDS is efficient because it only calculates distances to a set of data points which are called landmarks.

However, we found the operation after generating a reference set can be simpler while keeping the effectiveness. In this paper, we propose a mapping framework that

¹ Some techniques calculate a distance matrix and we won't differentiate similarity or distance matrix in statements since there is no essential difference between them, i.e. high similarities are equivalent with small distances.

directly treats the similarity matrix as the image data. The key novelty of our framework is that we propose a new data property to preserve and demonstrate the effectiveness of model based on a statistical foundation. Our embedding process preserves some clustering property, which is more flexible than to preserve the distance or neighborhood. This makes our algorithm much simpler and more efficient than the traditional ones while the effectiveness of clustering after the mapping isn't worse but even better under some conditions which will be analyzed in Section 2.1

2 The R-Map Framework

The main steps of R-map are very simple:

Given a categorical data set and a similarity measure:

Step 1: Randomly sample r data points as a reference set.

Step 2: Calculate the similarity between each data point and reference point to compose a similarity matrix.

Step 3: Apply PCA on the similarity matrix (optional), then clustering and visualization.

Some issues should be resolved to accomplish this algorithm such as what kind of categorical data is appropriate for this framework, how does the measure selection impact the framework and what is the detailed sampling strategy? We'll discuss these three issues in the following subsection.

2.1 Analysis of the Framework

Some notations used in this paper are summarized as follows.

SYMBOL	DESCRIPTION	SYMBOL	DESCRIPTION
A_j/A'_j	the j -th attribute/image attribute	n	number of data tuples
t_i/t'_i	the i -th data object/image data object	m	number of attributes (dimension)
D_k	domain of attribute A_k	d_{max}	maximum domain size
d_{kl}	the l -th value of domain D_k	c	number of clusters
C^j	the j -th cluster	p	number of major principal components
D	the categorical data space		

Issue 1: Definition of Categorical Cluster. The R-map framework is based on a specific cluster definition. The clusters C^1, C^2, \dots, C^c in categorical data are defined in a generative way, i.e. we assume the data points in the categorical space are generated by a set of discrete distributions. Formally, let $M_k(\mathbf{p}_k)$ denote a discrete distribution over D_k . $\mathbf{p}_k = (p_{k1}, p_{k2}, \dots, p_{k|D_k|})$ are the probabilities of taking the values in D_k and we have $\sum_{l=1}^{|D_k|} p_{kl} = 1$, where $p_{kl} \geq 0$.

Definition 2.1 (Cluster). A cluster C^j is generated from a set of discrete distributions: $M^j = (M_1^j(\mathbf{p}_1^j), M_2^j(\mathbf{p}_2^j), \dots, M_m^j(\mathbf{p}_m^j))$, ($j = 1, 2, \dots, c$). Therefore, the distribution of the whole data space is a mixture of $M^j: c_1M^1 + c_2M^2 + \dots + c_cM^c$, where c_k is the mixture coefficient means the probability of a data object belonging to the k -th cluster.

Note that the discrete distribution is general and it can be any specific discrete distribution such as multinomial distribution. Moreover, the distributions on each attribute needn't be independent to each other. Therefore, most categorical data that have clusters satisfy this assumption. Suppose data objects t_1, t_2 are generated from M^i and t_3 is generated from M^j . Following Definition 2.1 on the k -th dimension t_{1k}, t_{2k} can be viewed as generated from a subspace cluster $M_k^i(\mathbf{p}_k^i)$ and t_{3k} from $M_k^j(\mathbf{p}_k^j)$.

Issue 2: The Property Preserved in the Mapping. As mentioned in Section 1, a data object should be more similar to the objects in the same cluster than to those out of this cluster. We formulate this clustering property as an inequation between the *expectation* of intra-cluster and inter-cluster similarities (denoted as $s(\bullet)$).

Definition 2.2 (Clustering Property)

For $\forall C^i, C^j, i \neq j, t_1, t_2 \in C_i, \text{ and } t_3 \in C_j, \text{ we have}$

$$E[s(t_1, t_2)] \geq E[s(t_1, t_3)] \tag{1}$$

This definition is a reasonable assumption because it is nearly impossible to do distance/similarity based clustering analysis if the intra-cluster and inter-cluster distances/similarities can not be differentiated. We claim that this property can be preserved by the mapping of our framework, as is formalized in Lemma 1 if we use the *Simple-Match (SM)* similarity defined as

$$SM(t_1, t_2) := \sum_{k=1}^m (\delta(t_{1k}, t_{2k})), \text{ where } \delta(a, b) := \begin{cases} 1 & a=b \\ 0 & a \neq b \end{cases}.$$

Lemma 1. *Suppose o_x is a reference object each attribute of which is uniformly and independently generated from the attribute domains respectively. The definition of $t_{1,2,3}$ follows Definition 2.2. Denote random variable $X_1 = SM(t_1, o_x), X_2 = SM(t_2, o_x)$ and $Y = SM(t_3, o_x)$. We have $E[(X_1 - X_2)^2] \leq E[(X_1 - Y)^2]$, when $i \neq j$.*

Proof: We omit the proof due to lack of space.

Note that the squared Euclidian distance between two objects t'_i, t'_j in IDM is $\sum_{k=1}^r [IDM(i, k) - IDM(j, k)]^2$. Followed by Kolmogorov's Strong law of Large Numbers, it is consistent to $E(X_i - X_j)^2$ as r increases, where X_i and X_j denote the random variables that generates $t_i, t_j \in D$. Lemma 1 deems that the objects in the same cluster with high SM similarities in the original data will have small squared Euclidian distances in the image data, i.e. the clustering property defined in Definition 2.2 is preserved.

Issue 3: Sampling Strategy and Similarity Measure. As analyzed above, the key point of the R-map is the inequality between the expectation of intra-cluster similarity and inter-cluster similarity and we've demonstrate this inequality will be preserved using a simple distance definition and a reference set uniformly sampled from the background data space. Now we come to discuss the impact of different sampling strategies and different similarity measures.

If we directly sample the reference set from the original data points. Such sampling will cause dependence on the attributes of the reference points. But the impact of this

² The pdf file of this proof can be found in the web-page:
http://myfreefilehosting.com/f/b909a376b9_0.04MB

dependence is not negative. Actually, under most conditions, the dependence among the attributes makes the SM values between intra-cluster data points smaller and those between inter-cluster data points larger. The inequation in Lemma 1 will not be significantly impacted. Other similarity measures such as normalized simple match or Jaccard coefficient will not change the distance orders between objects, so the inequality can also be preserved. We have practically examined these discussions and the clustering effectiveness is similar to these different settings. We sample the reference set directly from the data points and use the SM measure in our experiments. Other sampling strategies and similarity measures are viewed as optional settings of our model.

3 Experimental Evaluation

The experiments are performed on a PC platform, with a Pentium(R) 4 3.2GHz CPU, and 1Gb Ram. The procedure of our framework's execution environment is Matlab(R) R14. We use numerous clustering evaluation measures for the clustering results including Accuracy (AC), Precision (PR), Recall (RE), Information Gain (IG), Normalized Mutual Information (NMI) and Adjusted Rand Index (ARI).

3.1 Experiments for Effectiveness Comparison with Related Works

In this part we experiment with real life categorical data sets taken from the UCI machine learning repository [3] for a overview of the effectiveness and efficiency comparison. There are totally 22 categorical data sets in the MLDB of the UCI machine learning repository. We remove some redundant ones and some have more than 1000 data objects for simplicity. On the remaining 14 data sets, we apply K-means clustering after various data mapping strategies including R-map, LMDS and LLE. We also compare our framework with K-modes clustering algorithm [4] and spectral clustering [2]. Since K-modes algorithm simulates K-means in the categorical data and spectral clustering is a clustering strategy based on pairwise distance matrices, these comparisons make sense.

The results of this part are shown in Figure 1 and Figure 2 using scatter plots. Figure 1 reveals the comparison of clustering effectiveness. Each sub-figure reveals the comparison between our framework and a related work (labeled on the Y-axis). Different marker types represent different clustering evaluation measures and each marker type has 14 instances which represent the results on 14 data sets. R-map outperforms LLE, spectral clustering and K-modes on these data sets and obtains similar effectiveness to LMDS. Figure 2 compares the efficiency of algorithms. The values are logarithmic because some relate works cost so long time that the scatter points assemble onto the Y-axis. In this scatter figure, different marker types represent different related algorithms and each point with the same marker represents a data set. R-map outperforms all the related algorithms on efficiency. It seems that only LMDS is comparable to R-map on clustering these data sets on effectiveness, so we'll pay more attention to this algorithm in the following parts.

3.2 Experiments for Scalability on Synthetic Data Sets

Since our model is most related to LMDS and the computational complexity with these two works can not be exactly compared, we empirically compare their scalability

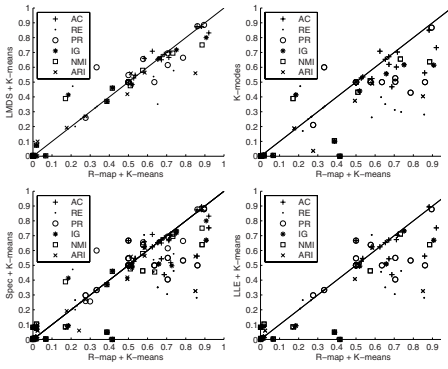


Fig. 1. Clustering effectiveness comparison:spec represents spectral embedding

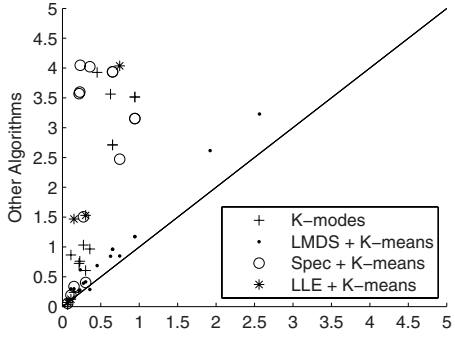


Fig. 2. Time (logarithmic) cost comparison

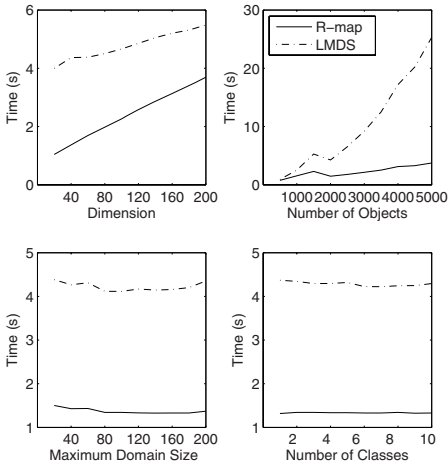


Fig. 3. Scalability evaluation

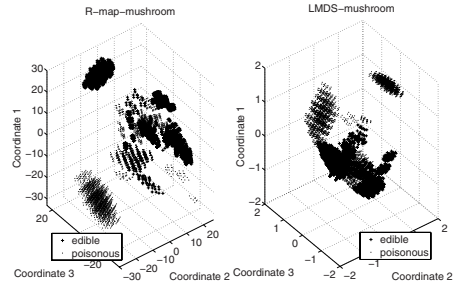


Fig. 4. Visualization for Mushroom. Note that the coordinates values are directly generated from the procedures and normalization on them doesn't significantly affect the visualization effectiveness

using synthetic data. We generate the synthetic data sets by a data generator implemented by ourselves, which can generate categorical data sets with optional n, m, d_{max} and c . The generation of clusters follows Definition 2.1. We compare the scalability of the mapping process between R-map and LMDS. Since the synthetic data sets can have optional n, m, d_{max} and c . We fix three of them and increase the remaining one. The average time costs of 50 rounds on each step are recorded to exam the scalability of our framework on each data scale. The results are shown in Figure 3, where each subfigure is the scalability performance on one data scale (labeled on X-axis). The real line

represents our framework and the dashed line represents the LMDS as a comparison. We find R-map outperforms LMDS especially on the number of data objects.

3.3 Experiment for Visualization

Data visualization uses the first three Principle components of the image data space and the experiment data is the well-known Mushroom data set from MLDB. The left plot of Figure 4 is the visualization of Mushroom data using R-map. We can see there are about 20 clusters from this visualization. This number is exactly the cluster number on which most distance based clustering approaches get excellent results. Moreover, the cluster sizes also accord with the results proposed by those previous clustering approaches such as reported in [9]. Compared with the right plot of Figure 4 which is visualization based on LMDS, R-map has better cluster aspects.

4 Conclusions

In this paper we introduce a simple and effective framework to map categorical data into a numerical space for visualization and clustering, which has theoretical guarantee and empirical demonstration. The key point of our model is a flexible data property which is preserved during the mapping, i.e. the difference between the inter-cluster similarities and the intra-cluster similarities. The theoretical analysis demonstrates that this property will be preserved if the data is categorical data, the clusters at which can be modeled by a set of discrete distributions.

Acknowledgments. This work is supported in part by NSFC grants 60673103 and 60421001.

References

1. Cox, T.F., Cox, M.A. (eds.): *Multidimensional scaling*. Chapman and Hall, London (1995)
2. Ding, C.: Spectral clustering. *icml2004 tutorial* (2004)
3. Newman, D.J., Hettich, S., Blake, C.L., Merz, C.: *UCI repository of machine learning databases* (1998), <http://www.ics.uci.edu/~mllearn/mlrepository.html>
4. Huang, Z.X.: A fast clustering algorithm to cluster very large categorical data sets in data mining. In: *Proceedings ACM SIGMOD International Conference on Management of Data*, ACM Press, New York (1997)
5. Kaufman, L., Rousseeuw, P. (eds.): *Finding Groups in Data: An Introduction to Cluster Analysis*. Wiley, New York (1990)
6. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: *Proc. 5th Symposium at Mathematical Statistics and Probability* (1965)
7. Platt, J.: Fastmap, metricmap, and landmark mds are all nystrom algorithms. In: *Proc. 10th International Workshop on Artificial Intelligence and Statistics*, pp. 261–268 (2005)
8. Roweis, S., Lawrence, S.: Nonlinear dimensionality reduction by locally linear embedding. *Science*
9. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes. *Information Systems* 25, 345–366 (2000)
10. Silva, V., Tenenbaum, J.B.: Global versus local methods in nonlinear dimensionality reduction. In: *Proc. NIPS 2003*, pp. 721–728 (2003)

Mining Changes in Patent Trends for Competitive Intelligence

Meng-Jung Shih, Duen-Ren Liu, and Ming-Li Hsu

Institute of Information Management, National Chiao Tung University, Hsinchu, Taiwan
{mj_shih, dliu}@iim.nctu.edu.tw

Abstract. Obtaining sufficient competitive intelligence is a critical factor in helping business managers gain and maintain competitive advantages. Patent data is an important source of competitive intelligence that enterprises can use to gain a strategic advantage. Under existing approaches, to detect changes in patent trends, business managers must rely on comparing two patent analysis charts of different time periods, it requires human effort and time. In this paper, we propose a patent trend change mining (PTCM) technique that can identify changes in patent trends without the need for specialist knowledge. We apply the PTCM approach to Taiwan's semiconductor industry to discover changes in four types of patent trends: the R&D activities of a company, the R&D activities of the industry, company activities in the industry and industry activities generally. The change mining approach generates competitive intelligence to help managers develop appropriate business strategies based on their findings.

Keywords: change mining, patent trend, competitive intelligence.

1 Introduction

Technological innovation is one of the critical success factors in business today. By analyzing patent data, managers can evaluate and understand trends in the development of technologies and plan suitable strategies [10]. Competitive intelligence helps enterprises measure the competition's potential, technological capabilities, and innovation performance in order to develop strategies for increasing revenue.

Changes in patent trends represent movements in the direction of technological innovation. It is important for business managers to be sensitive to changes in patent trends. There has been a great deal of research on patent data, and several applications, have been developed [1, 2, 3, 4, 5, 8]. Most of these studies/tools use statistical methods to analyze patent data in a specific period, and represent patent trends by visualization graphs and tables. However, these tools cannot express changes in patent trends over two time periods. In practice, experts usually identify changes in patent trends by comparing charts/tables for different periods, this requires human effort and time.

To capture changes in patent trends in different periods more efficiently and effectively, we propose a novel technology that can identify such changes without the need for specialist knowledge, and report the changes to analysts by ranking the degree of change. We combine association rule change mining with patent indicators to develop a technique called patent trend change mining (PTCM), which transforms patent

documents into a rule format and then identifies frequent rules among the rules. The frequent rules represent a patent trend in a specific period; thus, we can observe changes in patent trends by comparing the frequent rules of two time periods. In this study, we divide patent trends into four levels for analysis, and mine changes in different levels to help managers develop appropriate business strategies.

The remainder of this paper is organized as follows. In the next section, we review literature relevant to this research. Section 3 provides an overview of our PTCM technique. In Section 4, we describe the methods for mining changes in patent trends. In Section 5, we investigate changes in patent trends in Taiwan’s semiconductor industry. Then, in Section 6, we present our conclusions.

2 Background and Related Work

- Association rule mining

Association rule mining is a data mining technique used in various applications, such as market basket analysis. The technique searches for interesting associations or relationships among items in a large data set. Different association rules express different regularities that exist in a dataset; and two measures, support and confidence, are used to determine whether a mined rule is a regular pattern [6]. In this work, we apply association rule mining to patent data to find patent patterns (rule patterns).

- Patent analysis

Several software tools and services have been developed in the patent field [1, 5]. These tools analyze patents by classification, clustering, and statistical methods to find the relationships between patents with similar content. The results of patent analysis are usually presented as graphs or tables. Although existing patent analysis tools can provide various results, analysts still need to compare the results of two periods to identify changes over time. The motivation of this study is to discover changes in the patent trends of different time periods without the need for expert knowledge, and report changes to business managers by ranking the degree of change.

- Patent Indicators

Since the value of patents is rarely observable, scholars and research organizations have defined a number of patent indicators to determine the value of patents [2, 4, 8, 11]. The common patent indicators are described below [2, 4, 8, 11]:

Patent age: the age of a patent.

Originality: it indicates the diversity of cited patents, i.e., the patents cited by the target patent. The measure is based on the distribution (ratio) of cited patents over classes.

$$Originality = 1 - \sum_{j \in S_B} B_j^2 \tag{1}$$

$B_j = \frac{\text{Number of cited patents belonging to Class } j}{\text{Number of cited patents}}, S_B : \text{the set of classes of cited patents.}$

Generality: it indicates the diversity of citing patents, i.e., the patents that cite the target patent. The measure is based on the distribution (ratio) of citing patents over classes.

$$\begin{aligned}
 \text{Generality} &= 1 - \sum_{j \in S_F} F_j^2 \\
 F_j &= \frac{\text{Number of citing patents belonging to Class } j}{\text{Number of citing patents}}, \quad S_F = \text{the set of classes of citing patents.}
 \end{aligned}
 \tag{2}$$

Technology Cycle Time (TCT): it is the median age of the patents cited by the target patent. It is a measure of technological progress.

3 Patent Trend Change Mining Technique

As shown in Fig. 1, the proposed patent trend change mining (PTCM) technique comprises four components: a patent fetcher, a patent transformer, a patent indicator calculator, and a change detection module.

Patent fetcher: The patent fetcher module uses a keyword search strategy to retrieve patents for analysis. Patent fetcher acquires patent documents from the patent website and stores them into the patent document pool.

Patent transformer: This module transforms the raw patent document from HTML format into a text format, stores it in the database, filters out irrelevant content, and extracts required patent content, including the patent number, International Classification (IPC) et al. The extracted content is stored in the database for further processing to compute patent indicators.

Patent indicator calculator: This module calculates the patent indicators for each patent to determine the patent’s value. In this study, we use the citation index (CI), originality, generality and technology cycle time (TCT) as indicators to analyze patent documents.

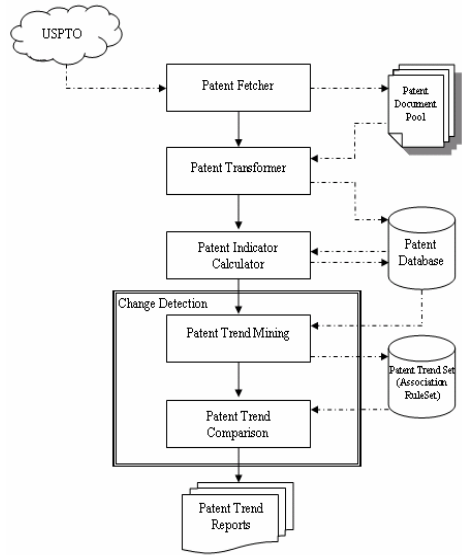


Fig. 1. An overview of the PTCM technique

4 Detecting Changes in Patent Trends

4.1 Patent Trend Mining

Before describing the patent trend mining module, we define four kinds of patent trends and classify them into two levels for analysis: company-level and industry-level trends.

- ◆ company-level patent trends: these trends provide information about a company’s technological development.

Trends in the R&D activities of a company: these changes can be determined by comparing the relations between technological fields (IPC) and four patent indicators over two time periods.

◆ industry-level patent trends: these trends provide information about the technological development of an industry.

Trends in the R&D activities of an industry: these changes can be determined by comparing the relations between the technological fields (IPC) and four patent indicators over two time periods.

Trends in the technological competitiveness of companies: we identify these changes by comparing the relations between a patent's assignee (company) and the four patent indicators over two time periods; the patent indicators reflect the technological competitiveness of a company.

Trends in the technological competitiveness of companies in a specific technological field: these changes can be observed by comparing the relations between both a patent's assignee and technological fields (IPC) and four patent indicators over two time periods.

We apply association rule mining to patent data to identify patent trends (frequent association rule patterns). The mined frequent patterns can be regarded as trends extracted from patent documents.

4.2 Patent Trend Comparison

After the patent trends of different time periods have been discovered, the trends (in rule format) are compared to identify changes.

Before we describe the rule matching process used to discover changes in patent trends, the types of change should be clarified.

Types of change:

Based on previous research [7, 9], we define four types of change in patent trends.

- (1) Emerging patent trends: an emerging patent trend is a rule pattern whose support increases significantly from one dataset to another.
- (2) Unexpected changes in patent trends: unexpected changes in patent trends can be found in newly discovered patent trends whose consequent parts of the rule patterns are different from those of the previous patent trend.
- (3) Added patent trends: an added patent trend is a new rule, i.e., a rule not found in previous rule patterns.
- (4) Perished patent trends: a perished patent trend is the opposite of an added rule, as it is only found in previous rule patents.

Rule matching:

We use rule matching method to compare the patent trends of different time periods [7, 9].

Identifying the type of change:

Table 1 shows the measures used to determine each type of event change; the measurements are adopted from [7, 9]. The four types of event change can be classified according to the two judged factors, i.e., the similarity measure S_{ij} and the

difference measure ∂_{ij} , and three predefined thresholds: θ_{em} for emerging patterns, θ_{un} for unexpected changes, and $\theta_{a/p}$ for added and perished rules ($\theta_{em} > \theta_{un} > \theta_{a/p}$). The process of identifying the types of changes follows a pre-determined sequence.

4.3 Evaluating the Degree of Change

As a large number of changes occur in a competitive business environment, managers need to focus on the most important changes. To do this, it is necessary to evaluate the degree of change, and rank the changed rules according to their importance.

Table 1 also shows the simple formulations for measuring the degree of change. The formulations, which are adopted from [7], measure the degree of change. After calculating the degrees of change, the most important changes are reported to business managers, who then analyze the changes in patent trends over different time periods and use the information to understand the changing business environment and plan appropriate strategies.

Table 1. Measurement and degree for each type of change

Type of Change (r_i^t, r_j^{t+k})	Measurement	Degree of Change
Emerging Pattern	$S_{ij} \geq \theta_{em}$ ($S_{ij} = C_{ij} \times Q_{ij}$), C_{ij} : similarity degree of the conditional parts Q_{ij} : similarity degree of the consequent parts.	$\frac{Support^{t+k}(r_j) - Support^t(r_i)}{Support^t(r_i)}$
Unexpected Change in patent trends	$Max(\zeta_i, \zeta_j) < \theta_{em}, \partial_{ij} > \theta_{un}$ $(\partial_{ij} = C_{ij} - Q_{ij})$	$\frac{Support^t(r_i) - Support^{t+k}(r_i)}{Support^t(r_i)} \times Support^{t+k}(r_j)$
Added Patent trend	$\zeta_j < \theta_{a/p}$ ($\zeta_j = \max_i S_{ij}$)	$(1 - \zeta_j) \times Support^{t+k}(r_j)$
Perished Patent trend	$\zeta_i < \theta_{a/p}$ ($\zeta_i = \max_j S_{ij}$)	$(1 - \zeta_i) \times Support^t(r_i)$

5 Changes in Patent Trends in Taiwan’s Semiconductor Industry

We used a keyword-based approach to select a subset of the Taiwan semiconductor-related patents from the USPTO patent database for the period 2001-2004. The dataset contains 4,310 unique patents is divided into two periods. The first part contains 2,352 patent documents for the period 2001 to 2002, while the second part contains 1,958 patent documents for the period 2003 to 2004. Table 2 lists some changes in patent trends in Taiwan’s Semiconductor Industry between 2001 and 2004.

- Changes in the R&D activities of Taiwan Semiconductor Manufacturing Co. Ltd:
 From patent trend (1) in Table 2, we observe the rapid growth (57%) of the company in terms of high originality in H01L29/788. This information shows that, during the period under study, TSMC exhibited a high degree of inventiveness in the technological field H01L29/788.

- R&D activities of Taiwan's semiconductor industry:

In Table 2, the emerging patent trend (2) shows that companies in the industry invested in H01L29/76 consistently throughout the period under study. The high growth rate (131%) indicates that companies focused their R&D activities on the technological field. However, the low CI indicates that the companies lacked pioneer patents and basic patents in these technological fields.

- Technological competitiveness of companies in Taiwan's semiconductor industry:

The added patent trends (4) shows new assignee of semiconductor patents, which means that new company (AOC) entered the semiconductor industry during 2003-2004.

- Technological competitiveness of companies in specific technological fields:

The perished patent trend (5) shows that UMC's technological competitiveness with medium CI in H01L21/336 declined, which may imply a change in UMC's innovative activities.

Table 2. Some changes in patent trends in Taiwan's Semiconductor Industry (2001-2004)

	Patent trend	Change degree
(1) Emerging patent trends	IPC=H01L29/788 → Originality= High	0.57
(2) Emerging patent trends	IPC=H01L29/76 → CI= Low	1.31
(3) Unexpected changes in patent trends	2001-2002: Assignee=Siliconware Precision Industries Co., Ltd. → Originality= High	0.03
	2003-2004: Assignee=Siliconware Precision Industries Co., Ltd. → Originality= Low	
(4) Added patent trends	Assignee=Au Optonics Corp. → CI= Low	0.04
(5) Perished patent trends	IPC=H01L21/336, Assignee= United Microelectronics Corp. → CI= Mid	0.02

6 Conclusions

Conventional patent analysis approaches and tools are based on statistical methods and analyze patent data in a given time period. Patent analysts discover changes in patent trends by comparing two patent analysis charts belonging to different periods. The comparison requires human effort and time. Moreover, the degrees of change can not be discovered intuitively; they must be calculated and ranked by analysts. We have proposed a patent trend change mining (PTCM) technique that captures changes in patent trends without the need for specialist knowledge and reports changes to business managers by ranking the degrees of change. We applied the proposed PTCM to Taiwan's semiconductor industry for the period 2001-2004 to discover changes in four types of patent trends. Competitive intelligence about business is derived by an automatic change mining approach that business managers can modify and develop appropriate strategies according to their findings.

Acknowledgement

This research was supported in part by the National Science Council of the Taiwan (Republic of China) under the grant NSC 96-2416-H-009-007.

References

1. Breitzman, A.F., Mogee, M.E.: The many applications of patent analysis. *Journal of Information Sciences* 28, 187–205 (2002)
2. Brockhoff, K.K.: Indicators of firm patent activities. In: *Technology Management: the New International language*, pp. 476–481 (1991)
3. Chen, M.C., Chiu, A.L., Chang, H.H.: Mining changes in customer behavior in retail marketing. *Expert Systems with Applications*, 773–781 (2005)
4. CHI-Research, <http://www.chiresearch.com>
5. Dou, H., Leveillé, V., Manullang, S., Dou, J.J.: Patent analysis for competitive technical intelligence and innovative thinking. *Data Science Journal* 4, 209–237 (2005)
6. Han, J., Kamber, M.: *Mining association rules in large databases. Data Mining-Concepts and Techniques*. Morgan Kaufmann Publishers, San Francisco (2001)
7. Liu, D.R., Shih, M.J., Liau, C.J., Lai, C.H.: Mining the change of event trends for decision support in environmental scanning. *Expert Systems with Applications* (2007), doi:10.1016/j.eswa.2007.10.016
8. Reitzig, M.: Improving patent valuations for management purposes- validating new indicators by analyzing application rationales. *Research Policy* 33, 939–957 (2004)
9. Song, H.S., Kim, J.K., Kim, S.H.: Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications* 21, 157–168 (2001)
10. Stemberge, B.: Sorting the wheat from the chaff- the use of patent analysis in evaluating portfolios (2005), <http://www.scientific.thomson.com/newsletter>
11. Tuomo, N., Hermans, R., Kulvik, M.: Patent citations indicating present value of the biotechnology business, <http://www.etla.fi/>

Seeing Several Stars: A Rating Inference Task for a Document Containing Several Evaluation Criteria

Kazutaka Shimada and Tsutomu Endo

Department of Artificial Intelligence, Kyushu Institute of Technology
680-4 Iizuka Fukuoka 820-8502 Japan
{shimada, endo}@pluto.ai.kyutech.ac.jp

Abstract. In this paper we address a novel sentiment analysis task of rating inference. Previous rating inference tasks, which are sometimes referred to as “seeing stars”, estimate only one rating in a document. However reviewers judge not only the overall polarity for a product but also details for it. A document in this new task contains several ratings for a product. Furthermore the range of the ratings is zero to six points (i.e., stars). In other words this task denotes “seeing several stars in a document”. If significant words or phrases for evaluation criteria and their strength as positive or negative opinions are extracted, a system with the knowledge can recommend products for users appropriately. For example, the system can output a detailed summary from review documents. In this paper we compare several methods to infer the ratings in a document and discuss a feature selection approach for the methods. The experimental results are useful for new researchers who try this new task.

Keywords: Sentiment analysis, Rating inference, Review mining.

1 Introduction

As the World Wide Web rapidly grows, a huge number of online documents are easily accessible on the Web. Finding information relevant to user needs has become increasingly important. The most important information on the Web is usually contained in the text. We obtain a huge number of review documents that include user’s opinions for products. Buying products, users usually survey the product reviews. More precise and effective methods for evaluating the products are useful for users. Many researchers have recently studied extraction and classification of opinions [6, 10, 11, 12, 14, 15].

There are many research areas for sentiment analysis; extraction of sentiment expressions, identification of sentiment polarity of sentences, classification of review documents and so on. In this paper we address a new sentiment analysis task of review documents. Most of existing studies for classification of review documents have handled two polarities: positive and negative opinions [10, 12]. On the other hand, several researchers have challenged not only p/n classification but also rating inference, namely seeing stars in a review document [8, 9]. We also handle a rating inference task in this paper.

The previous studies, p/n classification and rating inference, contain a problem; a document includes only one polarity (or stars). They did not discuss a task handling several polarities in a document. However, reviewers judge not only the overall polarity for a product but also details for it. For example, they are “performance”, “user-friendliness” and “portability” for laptop PCs and “script”, “casting” and “music” for movies.

In this paper we deal with a document containing several sentiment polarities. It is a new task for sentiment analysis: seeing *several* stars in a document. This is a primary experiment for the task. To estimate several ratings in a document is beneficial for users. Furthermore it is important for sentiment analysis tasks to extract words or phrases that relate to each polarity (evaluation criteria). Zhuang et al. have reported a method of movie review mining and summarization using the discovered p/n information [15]. If significant words or phrases for an evaluation criteria and their strength as positive or negative opinions are extracted, a system with knowledge that consists of them can recommend products for users appropriately. For example, the system can output a detailed summary from review documents: it generates not only a simple summary “This movie is good”, but also a more detailed summary “The story of this movie is excellent (five stars), but the music might be substandard (two stars)”.

In this paper we compare several methods for the rating inference task. Also we compare some feature sets for SVR in this task and discuss solutions for the improvement of accuracy. The experimental results are useful for new researchers who try this new task.

2 Task

There are many review documents of various products on the Web. In this paper we handle review documents about game softwares. Figure 1 shows an example of a review document. The review documents consist of evaluation criteria, their ratings, positive opinions, negative opinions and comments for a product. The number of evaluation criteria is 7: “Originality”, “Graphics”, “Music”, “Addiction”, “Satisfaction”, “Comfort”, and “Difficulty”. The range of the ratings, e.g. stars, is zero to six points.

We extract review documents from a Web site¹. The site establishes a guideline for contributions of reviews. In addition, the reviews are checked by the administrator of the site. As a result, the reviews unfitting for the guideline are rejected. Therefore the documents on the site are good quality reviews.

3 The Methods and Features

3.1 The Methods

In this section we describe 4 methods, which are SVM, SVR, Maximum entropy and a similarity based method, for inferring the ratings in a document.

¹ <http://ndsmk2.net>

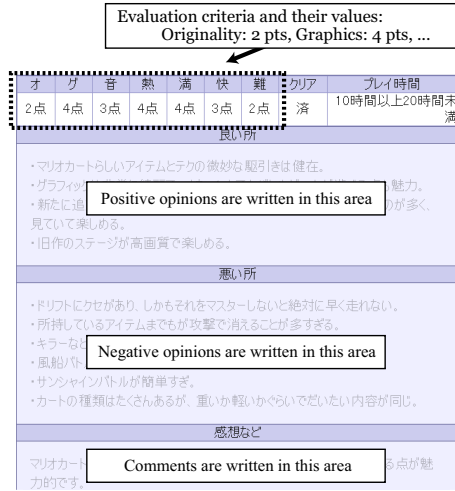


Fig. 1. An example of a review document

SVM and SVR. SVMs are a machine learning algorithm that was introduced by [13]. We expand the binary SVMs into a multi-classifier by using one-versus-one methods. Also we employ linear support vector regression (SVR). This is one of straightforward methods for this task. Related studies also used SVR for the rating inference task. We use the SVM^{light} package² for training and testing, with all parameters set to their default values [4].

ME. Maximum entropy modeling (ME) is one of the best techniques for natural language processing [1]. In this paper we use Amis³, which is a parameter estimator for maximum entropy models. We estimate parameters by using the generalized iterative scaling algorithm.

SIM. The 4th method is based on a similarity measure. We use the cos measure for the similarity calculation as follows:

$$sim(tr_x, te_y) = \frac{\sum_{i=1}^N x_i \cdot y_i}{\sqrt{\sum_{i=1}^N x_i^2 \times \sum_{i=1}^N y_i^2}} \tag{1}$$

where tr and te are a document in training data and a document in test data respectively. x_i and y_i are the value of a word i in tr and tr respectively. Next we extract documents of which the similarity exceeds a threshold. For the extracted documents, we compare the average values of each evaluation criterion. Finally we output the values as the result of the method.

² <http://svmlight.joachims.org>

³ <http://www-tsuji.is.s.u-tokyo.ac.jp/amis/index.html>

3.2 Feature Selection

For the features of the methods, we use words appearing in positive and negative opinions in review documents. We do not use words in comment areas because the accuracy with them in a preliminary experiment was lower than that without them. Here we distinguish words in the positive opinion areas and the negative opinion areas. In other words, for a word w_i , the word in the positive opinion areas is w_i^p and the word in the negative opinion areas is w_i^n . A vector of an evaluation criterion y for a document d_x is as follows:

$$d_{xy} = \{w_1^p, w_2^p, \dots, w_j^p, w_1^n, w_2^n, \dots, w_j^n\}$$

where j is the number of words appearing in review documents. We select words belonging to “noun”, “verb”, “adjective” and “adverb”. We use ChaSen for the morphological analysis⁴. The value of the features is based on the word frequency.

Next we consider two extensions for the feature selection. One approach is to use more complex information. In this paper, we use a word sequential pattern between two words in each sentence, namely cooccurrence. In the pattern extraction, we allow a skip between words. We extract word pairs within a length that we define. For example, we obtain the patterns “Fighting::WiFi, Fighting::excited, Fighting::me, WiFi::excited, WiFi::me, excited::me” from a sentence “Fighting with WiFi excited me.”

Another approach for improvement of the accuracy of a classifier is to select effective and significant features for the feature space. Furthermore it seems unlikely that all words in a document contribute to all evaluation criteria. In other words some words that are significant to estimate the rating of an evaluation criterion exist in a review document. To extract the words, we compute a confidence measure of each word. The confidence measure in this paper is variance of words concerning each evaluation criterion. We measure whether a word appears frequently with the same point regarding an evaluation criterion. It is computed as follows:

$$var(w_{c_j}) = \frac{1}{m} \sum_{i=0, w \in d_i}^n (real(d_i, c_j) - ave(w_{c_j}))^2 \quad (2)$$

where c_j is an evaluation criterion. m and n are the document frequency (df) of a word w (or a word pair) and the number of documents respectively. $real(d_i, c_j)$ and $ave(w_{c_j})$ are the actual rating of c_j in d_i and the average score of w for c_j . We use w of which the var is a threshold or less.

Furthermore we apply two conditions to the feature selection.

Frequency (F). The frequency of a word is n times or more.

Evaluation value (E). If a word w appears in “positive opinion area”, the actual rating of the evaluation criterion have to be 3 points or more. If a word w appears in “negative opinion area”, the actual rating of the evaluation criterion have to be 3 points or less.

⁴ <http://chasen.naist.jp/hiki/ChaSen/>

4 Experiment

In this section, we explain datasets and criteria for the experiment first. Then we evaluate our method with a dataset and discuss the experimental results.

4.1 Dataset and Criteria for the Experiment

We evaluated this new sentiment analysis task with a dataset that consists of 1114 review documents that consist of different kinds of game softwares such as RPGs and action games of Nintendo DS, namely a mixed dataset. In this experiment we evaluated the dataset with 5-fold cross-validation.

In this experiment, we evaluated the outputs of each method with the following criteria: the mean squared error (MSE) between actual ratings and outputs of each method, the standard deviation (SD) of the MSE, and the accuracy . The mean squared error (MSE) is computed as follows:

$$MSE_j = \frac{1}{n} \sum_{i=1}^n (out(d_{ij}) - real(d_{ij}))^2 \quad (3)$$

where i and j denote a review document and an evaluation criterion in the document respectively. *out* and *real* are the output of a method and the actual rating in a document respectively. We converted the outputs of the SVR and the similarity based method into integral value with half adjust because it was continuous. The MSE is one of important criteria for the rating inference task because not all mistakes of estimation with the methods are equal. For example, assume that the actual rating of a criterion is 4. In this situation, the mistake of estimating it as 3 is better than the mistake of estimating it as 1.

In this experiment, we used two types of accuracy. The first accuracy is simple accuracy, that is to say the correspondence between real ratings and outputs. The second one is PNN accuracy (Positive-Neutral-Negative). For the PNN accuracy, we defined 4 and 5 points as “Positive”, 3 points as “Neutral” and 0, 1, 2 points as “Negative”.

4.2 Results

First we compared the methods with bag-of-words (Bows) features only. We ran the SVR and SVM with all default parameters in this experiment. For the Maximum entropy we estimate parameters by using the generalized iterative scaling algorithm.

Table 1 shows the result. “All-3” in the table is the MSE in the assumption that the ratings of all criteria are 3. “Ave” is the MSE computed from actual ratings and average values of each evaluation criterion in the training data. The average values are discretized for the MSE computation. These MSEs are baselines for this task. As you can see, all methods outperformed the baselines 2.

⁵ We evaluated the Naive Bayes classifier and C4.5 with the same dataset. However, the MSEs of them were larger than the average-based baseline.

Table 1. Comparison with baselines

	All-3	Ave	SVR	SVM	ME	SIM	
MSE	Originality	1.26	1.54	0.88	0.91	0.98	1.03
	Graphics	1.03	0.85	0.74	0.78	0.82	0.84
	Music	1.21	0.79	0.70	0.69	0.75	0.77
	Addiction	1.89	1.89	1.21	1.54	1.44	1.45
	Satisfaction	1.97	1.77	1.22	1.54	1.57	1.42
	Comfort	1.29	1.29	1.13	1.24	1.35	1.27
	Difficulty	1.74	1.17	1.22	1.23	1.35	1.18
	Average	1.48	1.33	1.02	1.13	1.18	1.14
SD	0.17	0.24	0.12	0.19	0.19	0.20	
Accuracy	26.60	37.69	41.37	41.76	40.23	39.47	
PNN Accuracy	26.60	51.98	57.41	58.43	57.05	55.71	

Table 2. The effectiveness of *var*

<i>var</i>	0.25	0.5	0.75
MSE (Ave)	0.99	0.99	0.99
MSE (SD)	0.12	0.12	0.12
Accuracy	41.40	41.59	41.56
PNN Acc	57.49	57.49	57.46

In this experiment, the SVMs produced the best accuracy. However the MSE of the SVR was the smallest of them. The SD of the SVR was also small. As a result, we arrived at the conclusion that the SVR was the most suitable in this experiment because the MSE is the most important criterion in this task.

Next we compared the results concerning the extensions for the feature selection, namely word sequential patterns and a confidence measure *var* based on the variance. In this experiment, we used the SVR only for the evaluation. Here we applied the extension with *var* to word sequential patterns only. Table 2 shows the comparison of the value of *var*. In this experiment, the length for the pattern extraction was 4. The value of the condition of the frequency (F) in Section 3.2 was 16. Table 3 shows the comparison of the length for the pattern extraction. The value of the *var* was 0.5. As you can see, there is no difference in the MSE and the accuracy.

Here we need to discuss a problem for this task. In this task, there is a possibility that humans even can not infer a rating in a document because a document contains many evaluation criteria. In other words, words or phrases for an evaluation criterion do not exist in a document occasionally. Therefore we inquired into 30 documents selected from review documents randomly. We judged whether we could infer each criterion in the documents or not. The criterion of the judgment was whether the document contained words or phrases for an evaluation

⁶ Although we compared several conditions of the frequency (F) in this experiment, there is no difference in the MSE and the accuracy.

Table 3. The effectiveness of the patterns

Length	1	2	6
MSE (Ave)	1.02	1.00	0.99
MSE (SD)	0.12	0.12	0.13
Accuracy	41.39	41.39	41.46
PNN Acc	57.08	57.34	57.26

criterion or not⁷. As a result, approximately 75% of all criteria could be inferred by humans. We think that this is one reason that the accuracy was low. However, the judgment of the possibility of inference was examined by one test subject only. We need to discuss the reliability of the judgment process with some test subjects by using a concordance rate such as the Kappa coefficient [2].

4.3 Discussion

In this section we discuss this task on the basis of the experimental results. The accuracy in the experiment was insufficient; approximately 41% for the 5-fold cross-validation. These results show the difficulty of this “seeing several stars” task (6 grades for 7 criteria). We need to discuss the improvement of the accuracy and the MSE. We think that dictionaries obtained from opinion extraction or word polarity estimation tasks [5, 6, 14] are useful to infer the ratings in our task.

In this experiment, we used SVR to estimate the ratings in a document. The SVR is often utilized in rating inference tasks [8, 9]. However Koppel and Schler [7] have discussed a problem of use of regression for multi-class classification tasks and proposed a method based on optimal stacks of binary classifiers. Pang et al. [9] have proposed a method based on a metric labeling formulation for the rating inference problem. The results of these studies denote that SVR is not always the best classifier for this task. We need to consider other methods for the improvement of the accuracy. We have proposed high accuracy classifiers for a p/n classification task [11]. The method incorporated three classifiers: SVMs, Maximum Entropy and score calculation. In the movie review classification task [10], this multiple classifier improved the accuracy as compared with the single classifiers. Applying this method to this task is one of our future work.

The size of the dataset in this experiment was not large: 1114 documents. To generate a high accuracy classifier, we need a large amount of training data. Goldberg and Zhu [3] have argued the significance of training data acquisition from unlabeled data. As an additional experiment, we evaluated the SVR-based method with bows and patterns based on the value of *var* computed from 1114

⁷ Here we did not consider the correctness of ratings estimated by us. For example, if we could infer an evaluation criterion by reading the positive opinion area in the case that the rating was 4 or 5, we judged that the evaluation criterion could be inferred.

review documents⁸. As a result, the accuracy increased by 11%⁹. We think that one reason for the improvement is the increase of training data for the *var* calculation. Therefore, we need to consider a training data extraction method.

5 Conclusion

In this paper we described a novel sentiment analysis task of rating inference. The documents in this task include 7 evaluation criteria that contain 6 rating points: seeing several stars in a document. As a primary experiment for this task we inferred the ratings in each document and compared some machine learning techniques. As a result, the support vector regression (SVR) produced the best performance. We also explained the feature selection based on variance of words and the use of word sequential patterns. The experimental results show that this is a difficult task of sentiment analysis and we need more training data. Future work includes (1) extraction of more effective features for a classifier, (2) evaluation with other classification methods.

References

- [1] Berger, A.L., Della Pietra, S.A., Della Pietra, V.J.: A maximum entropy approach to natural language processing. *Computational Linguistics* 22(1), 39–71 (1996)
- [2] Cohen, J.: A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20(1), 37–46 (1960)
- [3] Goldberg, A.B., Zhu, X.: Seeing stars when there aren't many stars: Graph-based semi-supervised learning for sentiment categorization. In: *HLT-NAACL 2006 Workshop on Textgraphs: Graph-based Algorithms for Natural Language Processing* (2006)
- [4] Joachims, T.: Transductive inference for text classification using support vector machines. In: *Proceedings of the Sixteenth International Conference on Machine Learning*, pp. 200–209 (1999)
- [5] Kawano, Y., Shimada, K., Endo, T.: Sentence polarity classification based on a scoring method (in Japanese). In: *HINOKUNI Symposium 2007 CD-ROM A-3-4* (2007)
- [6] Kobayashi, N., Iida, R., Inui, K., Matsumoto, Y.: Opinion extraction using a learning-based anaphora resolution technique. In: *Proceedings of the Second International Joint Conference on Natural Language Processing (IJCNLP-2005)*, pp. 175–180 (2005)
- [7] Koppel, M., Schler, J.: The importance of neutral examples in learning sentiment. *Computational Intelligence* 22(2), 100–109 (2006)
- [8] Okanohara, D., Tsujii, J.: Assigning polarity scores to reviews using machine learning techniques. In: *Proceedings of the 2nd International Joint Conference on Natural Language Processing (IJCNLP)*, pp. 314–325 (2005)

⁸ Conditions: *var* = 0.25, the frequency ≥ 2 for Bows and *var* = 0.25, the frequency ≥ 2 , the length = 4 for patterns.

⁹ However, the method with the conditions could not estimate the ratings for documents of 15% of the test data because zero vectors are often generated owing to the condition of the value of *var*. Moreover, this is a close experiment.

- [9] Pang, B., Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 115–124 (2005)
- [10] 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), pp. 79–86 (2002)
- [11] Tsutsumi, K., Shimada, K., Endo, T.: Movie review classification based on a multiple classifier. In: The 21th Pacific Asia Conference on Language, Information and Computation (PACLIC) (2007)
- [12] Turney, P.D.: Thumbs up? or thumbs down? semantic orientation applied to unsupervised classification of reviews. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, pp. 417–424 (2002)
- [13] Vapnik, V.N.: Statistical Learning Theory. Wiley, Chichester (1999)
- [14] Wilson, T., Wiebe, J., Hwa, R.: Just how mad are you? finding strong and weak opinion clauses. In: AAAI Spring Symposium on Exploring Attitude and Affect in Text: Theories and Applications (2004)
- [15] Zhuang, L., Jing, F., Zhul, X.-Y.: Movie review mining and summarization. In: Proceedings of the ACM 15th Conference on Information and Knowledge Management (CIKM-2006), pp. 43–50 (2006)

Structure-Based Hierarchical Transformations for Interactive Visual Exploration of Social Networks

Lisa Singh, Mitchell Beard, Brian Gopalan, and Gregory Nelson

Georgetown University, Washington DC 20057, USA

Abstract. In this paper, we propose hierarchical transformations of traditional social networks based on structural expansion values of nodes in the network. The hierarchical visualization clusters or groups nodes with similar structural expansion values in the network. It is a complement to traditional network visualization and gives users the ability to quickly understand how structure is distributed throughout the network. After describing our approach, we analyze a real world social network, highlighting the benefit of a network structure-based hierarchical transformation for visual exploration of this network.

1 Introduction

It can be difficult to understand and interpret data mining results. One of the goals of visual mining is to combine visual and analytic approaches to give users the ability to manipulate the data and better understand the data space. When investigating large social networks, a need exists to identify common structures in the network. For example, if we know that an individual in the network has an important role in the network, we may be interested in finding others that play a similar role in the network. The role may be based on relationships to others in the network (one or more graph invariants), as well as on features of the individual.

In this paper, we propose an alternative view of traditional social networks based on hierarchies that support the exploration goals of visual data mining. While hierarchies have been used extensively for visualization of 'top down' semantic relationships, they have not been used to identify structural similarity based on node expansion within social networks. The hierarchical visualization clusters or group together nodes with similar structural properties in the network. It is a complement to traditional network visualization and gives users the ability to quickly understand how structure is distributed throughout the network. Further, the structural similarity is based on a node's view of the *entire* network, not just its relationship to its *immediate* neighbors. We will illustrate that determining this structural similarity using the traditional graph view is a complicated visual exploration task, particularly if the network contains more than a handful of nodes. By considering a complementary hierarchical view of the network, the similarities are more apparent to the user.

There are a number of important benefits to these hierarchies including a easily interpretable clustering based on expansion vectors of structural properties associated with network nodes, a simple construction algorithm, and interactive exploration using both a hierarchical and a traditional graph representation of the social network. Finally, the contributions of this paper are twofold. First, we introduce a novel network structure-based hierarchical transformation of a graph that is based on expansion vectors. Second, we demonstrate the utility of this transformation by using it within a visual mining tool to explore real world social networks.

The paper is organized as follows. Section 2 begins with a motivating example and background that describes when our hierarchical transformation is beneficial for analysis. We describe the hierarchy semantics and clusters in section 3. Section 4 presents a brief visual mining case study using the Invenio network mining software. Finally, Section 5 presents conclusions and final observations.

2 Social Network Background and Motivation

2.1 Social Network Graph Models

When analyzing social networks, both attribute data and relationship data are important for gaining insight about the dynamics or patterns within the network. Examples of social networks include blog networks, email networks, disease transmission networks, and communication networks. Typically, a social network is modeled as a graph, $G = (V, E)$. Here, the actors are represented as a set of n vertices or nodes, $V = \{v_1, v_2, \dots, v_n\}$, and the relationships between actors are represented as m edges or links between the nodes, $E = \{(v_i, v_j) \mid v_i, v_j \in V, i \neq j, i \leq n, j \leq n\}$. While we can use hierarchical transformations on multi-mode graphs with multiple node types, we will focus our discussion on uni-mode networks with a single node type.

We further extend this representation to include attributes or features associated with each actor or relationship. If we consider V to be a relation and each node in V to be an instance of a tuple, then we can specify the relation as $V(v_{id}, a_1, a_2, \dots, a_k)$, where v_{id} is the node id and $a_1 \dots a_k$ is the set of attributes associated with each node in V . We will refer to this attribute data as *semantic content* since it is domain specific. Similarly, we can specify a set of attributes for the edges E , where E is an associative relation. Here, $E = \{e_{id}, b_2, \dots, b_l\}$ and each edge is an instance of a tuple in E .

We will refer to social network measures or graph invariants that are calculated using the topology of the network as *structural properties*. Some well known centrality structural properties include: (1) Degree $degree(a_{ego})$ - the number of nodes directly connected to a_{ego} . (2) Betweenness $between(a_{ego})$ - the number of shortest geodesic paths that pass through a_{ego} . (3) Eigenvector $ev(a_{ego})$ - The number of 'important' nodes connected to a_{ego} is used to measure the importance of a_{ego} . We refer you to Wassermann and Faust for a detailed discussion of these and other centrality social network metrics [5].

2.2 Expansion Values for Centrality Measures

One extension of any centrality measure is to take the average value of the measure for all the nodes in the network. Instead, we investigate a different extension we call *expansion*. Informally, expansion is a node's view of the network at different distances. The view is based on the calculation of some centrality measure of a node and its neighbors. The node has a value for the measure, the node's neighbors have a value of the measure, the neighbor's neighbors have a value of the measure, etc. Expansion aggregates this measure for all the neighbors at a particular distance from the node to create a vector of centrality values.

To make our discussion more concrete, we will focus on a particular expansion value, *hop expansion*, an expansion vector for node degree. The centrality measures previously described give little insight into the connectivity patterns or landscape of subgraphs in the network. The landscape can be seen more easily using the hop expansion measure because it is not a single scalar value, but a vector of values. For hop expansion, it is a measure of the number of nodes at different distances from a particular node, a_{ego} . We refer to the evaluation of a centrality measure at different distances from each node in the network as an *expansion vector*.

Definition 1. *The ϕ -hop level of a node a_{ego} is the set of nodes $N_\phi(a_{ego}) = \{v_1, \dots, v_j\}$ for which the shortest path between any $v_i \in N_\phi(a_{ego})$ and node a_{ego} has length ϕ . Formally,*

$$\begin{aligned} N_\phi(a_{ego}) &= \{v_i \mid \text{distance}(a_{ego}, v_i) = \phi, \\ &1 \leq \phi \leq \text{diam}(G), \\ &a_{ego} \neq v_i, v_i \in V\} \end{aligned}$$

We refer to the size of this set as a node's hop degree, $H_\phi(a_{ego})$, where $H_\phi(a_{ego}) = |N_\phi(a_{ego})|$. We then define $\mathbf{H}(a_{ego})$ as the set of ordered pairs $(\phi, H_\phi(a_{ego}))$ for all hop levels up to the diameter of the network:

$$\mathbf{H}(a_{ego}) = \{(\phi, H_\phi(a_{ego}))\} \quad \forall \phi \leq \text{diam}(G)$$

This measure shows us the hop expansion for a node in the network. Each element in the vector is a pair of values, the hop level and the hop degree. Hop expansion is an extension of degree and gives insight about the network connectivity of the entire graph from a single nodes perspective. This measure captures the influence of other nodes on a_{ego} . Is a_{ego} surrounded by a tightly bound network (characterized by early large values) or does it go through several medium increases?

The vector for each node contains up to $\text{diam}(G)$ elements, the diameter of the network. It has been shown that the degree distribution of social networks is not random. Instead, many large graphs, including the web, follow a power law distribution and have a small diameter ($\lesssim 50$) [1]. Therefore, the size of $\mathbf{H}(a_{ego})$ is small relative to the size of the network. Also, the calculation is straightforward and can be completed by calculating all-pairs shortest path. We are investigating alternative heuristics for this sparse data set.

Definition 2. *The hierarchical transformation approximation G^* is based on the set of hop expansion vectors for every node in the network, G . Formally, $G^* = \{\mathbf{H}(v_i) \forall v_i \in V\}$.*

G^* is the set of hop expansion values for every node in G . This transformation is potentially very powerful. We will use this new representation as a way to create hierarchies of the original network. We note, that while we have illustrated the concept of 'expansion' using 'hop expansion', any centrality measure can be used as the basis for the expansion vector. If the centrality measure is a real value, then rounding and/or binning ranges of values is an option.

2.3 Motivation

Traditional visual analytics tools let us filter this network based on attribute values or centrality measure, e.g. display nodes with degree greater than 3. While this provides some insight, we are still not able to readily identify nodes with structural similarity based on a node's 'view' of the other nodes in the network. If instead we had a rooted graph approximation that contained centrality aggregate values for neighborhoods a particular distance away, nodes with the same expansion vectors could be represented as the leaves of the same branches of a tree. With this structural similarity information clearly illustrated, social scientists can investigate questions such as:

- How do nodes with the most influence compare structurally and semantically?
- How does information propagate through the network?
- Do nodes with a similar structural position have similar attribute values?
- How diverse is the structural landscape of the network?

Therefore, in this paper, we propose giving users the ability to create *hierarchical transformations* of the original network based on structural properties of the nodes in the network. Each level of the hierarchy approximates a level of neighborhood structural information. For example, suppose that we build a hierarchy based on the hop expansion values of each node in the network. If two nodes have the same hop expansion vector, then their overall view of the network is similar, e.g. the two nodes have the same number of neighbors; their neighbors have the same number of neighbors; their neighbor's neighbors have the same number of neighbors, etc. Using this information, sociologists can then compare these nodes semantically by coloring based on attribute value(s). They can also use this information to help identify potential similar community clusters in the network.

3 Semantics of Structure-Based Hierarchical Transformation

When investigating large social networks, a need exists to identify common structures in the network. We accomplish this by transforming the traditional node

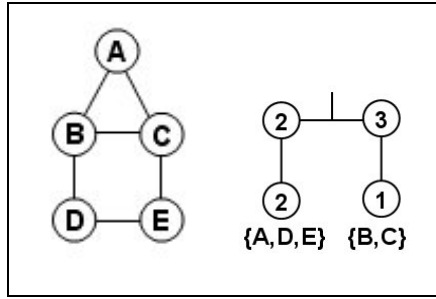


Fig. 1. Example - Left: social network; Right: hierarchical transformation

and edge graph representation to another meaningful structure, hierarchies that approximate a structural property in the network. However, unlike the hierarchies proposed in previous literature [3], these structural hierarchies take a traditional network structural property, e.g. degree, and show the expansion of that property across the network, where expansion is defined in Section 2. These hierarchies can be viewed as path prefix trees containing ego network structure vectors for each node in the network.

Once the hierarchy is built, nodes with similar structural properties are children of the same branches. Each level of the hierarchy maps to a distance level ϕ for the actors in G . Each node label in the hierarchy corresponds to a hop degree, H_ϕ , of one or more nodes in G . The value associated with each key in the tree is the number of nodes in the social network with the prefix.

We will now go through an example for hop expansion of the toy social network illustrated in Figure 1. In that network, there are 5 nodes. The hop expansion values for node A are $H_1(A) = 2$, $H_2(A) = 2$, and $\mathbf{H}(A) = \{(1, 2), (2, 2)\}$. Figure 1 shows the hierarchy for our example graph. For clarity of the example, we show the node label mapping to the hop expansion vectors in the hierarchy at the leaf nodes. The nodes have two structural paths $P_1(2_1, 2_2)$ and $P_2(3_1, 1_2)$ in the hierarchy, $p = 2$. Each node in the hierarchy contains the number of nodes with the prefix hop expansion vector. Also, since the network contains a single connected component, the sum of key values for each path from the root to a leaf node is $n - 1$. For our example, the sum of each path is 4.

For this example, nodes A , D , and E have the same structural hop expansion vector. This means that nodes A , D , and E connect to the same number of nodes and their respective neighbors also connect to the same number of nodes. Therefore, A , D , and E are said to have the same 'hop expansion' position in the network.

4 Visual Mining Case Study

For this case study, we used a coauthorship network of scientists studying networks. The data set was created in 2006 and contains 1589 scientists and 2742

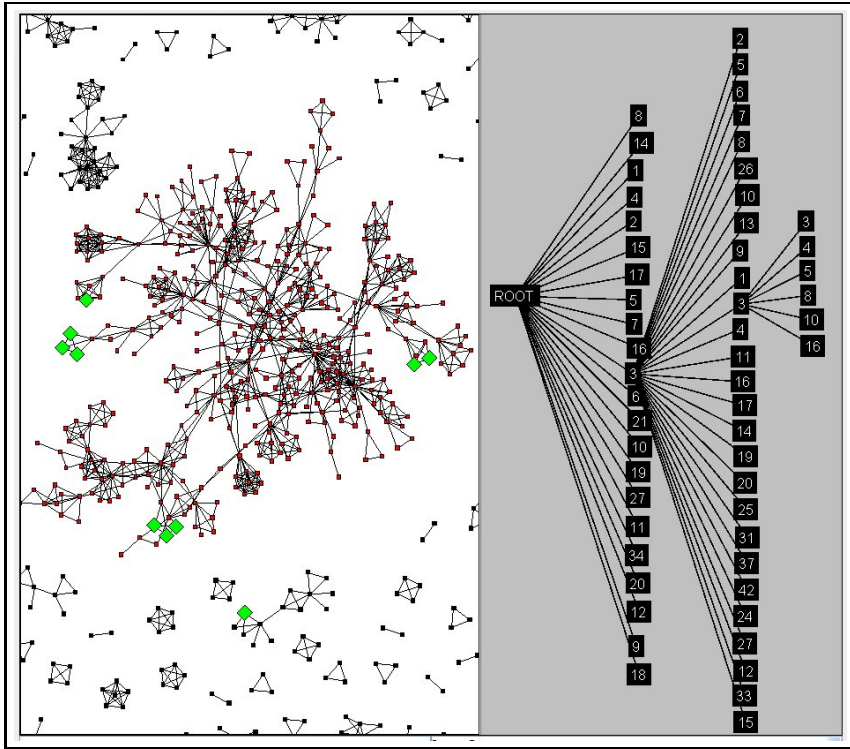


Fig. 2. Interactive mapping between hierarchy and social network

edges [2]. The network is shown in Figure 3 on the left size of the window using the Invenio visual mining interface [4]. Invenio is a visual mining tool for interactive exploration of social networks. To incorporate the hierarchical transformation, we implemented a dual screen that allows a user to explore the hierarchical representation and see the results of the exploration on the full social network.

Using the traditional social network graph layout, we can highlight nodes with certain degree values. We then select the nodes of interest to us of a particular degree. At that stage, we can look at the neighborhoods of our neighbors and see how they compare. This process can continue iteratively until we feel that the view of the network is similar for the nodes being analyzed.

The process of analysis using hierarchical transforms begins with the user selecting an option to build a hierarchical transform of the graph using expansion values. The user then selects the structural measure that will be used for the transformation and the maximum distance of interest. The tool then uses the selections, generates an expansion vector and places each node in the hierarchy based on the vector values. We use a classical tree layout where children nodes are positioned below their common ancestor.

The screen is then split so that both the hierarchy and the original network can be seen. Initially, only the root of the structural hierarchy is expanded. Then

the user can focus on different parts of the hierarchy by selecting them and seeing where the nodes are in the original graph. For this example, we follow one branch multiple levels and then highlight the nodes in the original graph as shown in Figure 2. The larger nodes in the original graph correspond to the nodes with the shown expansion subbranch. We see that the nodes with expansion vector 3 and 3 in G^* appear in different parts of the full network G . A sociologist can use this as evidence that like regions or structural redundancy exist in different areas on the network. Various 'macro' level pieces of information can be gathered from the hierarchy. For example, if the different branches of the hierarchy have similar hop expansion values, then information flow is relatively even throughout the network.

5 Conclusions and Future Directions

Hierarchies have been used to analyze networks containing parent child relationships. Here, we consider using hierarchies to understand the structural relationship that exists among actors in traditional social networks. We build the hierarchy based on graph invariants of different actors or nodes in the network. We are then able to easily identify like structures across a network.

There are a number of future extensions including merging branches next to each other to create larger bins of similar nodes and incorporating attribute semantics into the hierarchy expansion process.

References

1. Faloutsos, M., Faloutsos, P., Faloutsos, C.: On power-law relationships of the internet topology. In: SIGCOMM 1999: Proceedings of the conference on Applications, technologies, architectures, and protocols for computer communication, pp. 251–262. ACM Press, New York (1999)
2. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices (2006)
3. Plaisant, C., Grosjean, J., Bederson, B.B.: Spacetime: Supporting exploration in large node link tree, design evolution and empirical evaluation. In: INFOVIS 2002: Proceedings of the IEEE Symposium on Information Visualization (InfoVis 2002), Washington, DC, USA, p. 57. IEEE Computer Society Press, Los Alamitos (2002)
4. Singh, L., Beard, M., Getoor, L., Blake, M.B.: Visual mining of multi-modal social networks at different abstraction levels. In: IV 2007: Proceedings of the 11th International Conference Information Visualization, Washington, DC, USA, pp. 672–679. IEEE Computer Society Press, Los Alamitos (2007)
5. Wasserman, S., Faust, K.: Social network analysis: methods and applications. Cambridge University Press, Cambridge (1994)

CP-Tree: A Tree Structure for Single-Pass Frequent Pattern Mining

Syed Khairuzzaman Tanbeer, Chowdhury Farhan Ahmed,
Byeong-Soo Jeong, and Young-Koo Lee

Department of Computer Engineering, Kyung Hee University
1 Seochun-dong, Kihung-gu, Youngin-si, Kyunggi-do, 446-701, Republic of Korea
{tanbeer, farhan, jeong, yklee}@khu.ac.kr

Abstract. FP-growth algorithm using FP-tree has been widely studied for frequent pattern mining because it can give a great performance improvement compared to the candidate generation-and-test paradigm of *Apriori*. However, it still requires two database scans which are not applicable to processing data streams. In this paper, we present a novel tree structure, called CP-tree (Compact Pattern tree), that captures database information with one scan (*Insertion phase*) and provides the same mining performance as the FP-growth method (*Restructuring phase*) by dynamic tree restructuring process. Moreover, CP-tree can give full functionalities for interactive and incremental mining. Extensive experimental results show that the CP-tree is efficient for frequent pattern mining, interactive, and incremental mining with single database scan.

Keywords: Data mining, data stream, frequent pattern, association rule.

1 Introduction

Finding frequent patterns (or *itemsets*) plays an essential role in data mining and knowledge discovery techniques, such as association rules, classification, clustering, etc. A large number of research works [1], [7], [5], [3] have been published presenting new algorithms or improvements on existing algorithms to solve the frequent pattern mining problem more efficiently. FP-tree based FP-growth mining technique proposed by Han et. al. [5] has been found one of the efficient algorithms using the prefix-tree data structure. The performance gain achieved by FP-growth is predominantly based on the highly compact nature of FP-tree, where it stores only the frequent items in a frequency-descending order. During mining this item arrangement not only enables it to avoid global infrequent node deletion process from each conditional tree but also reduces the search space to find next frequent item in item list to one item. However, construction of such FP-tree requires two database scans and prior knowledge about support threshold, which are the key limitations of applying FP-tree in data stream environment, incremental, and interactive mining.

The prefix-tree based approach may suffer from the limitation of memory size when it tries to hold whole database information. However, as the currently available memory size becomes more than GBytes, several prefix-tree data structures capturing partial (with an error bound) [4] or whole [6], [3] database information have been

proposed for mining frequent patterns. AFPIM [4] algorithm performs incremental mining mainly by adjusting the FP-tree structure. Therefore, it requires two database scans. CATS tree [6] is a single-pass solution but it still suffers from complex tree construction process. The above two limitations are well-addressed in CanTree [3] that captures the complete information in a canonical order of items from database into a prefix-tree structure in order to facilitate it for incremental and interactive mining using FP-growth mining technique. Although CanTree offers a simple single-pass construction process, it usually yields poor compaction in tree size compared to FP-tree. Therefore, it is storage and runtime inefficient causing higher mining time since the items in the tree are not stored in frequency-descending order.

In this paper, we propose a novel tree structure, called CP-tree (Compact Pattern tree), that constructs a compact prefix-tree structure with one database scan and provides the same mining performance as the FP-growth technique by efficient tree restructuring process. Our comprehensive experimental results on both real-life and synthetic datasets show that frequent patterns mining, interactive and incremental mining with our CP-tree outperforms the state-of-the-art algorithms in terms of both execution time and memory requirements.

The rest of the paper is organized as follows. Section 2 describes the structure and restructuring process of CP-tree. We report our experimental results in Section 3. Finally, Section 4 concludes the paper.

2 Overview of CP-Tree: Construction and Performance Issues

Let $L = \{i_1, i_2, \dots, i_n\}$ be a set of literals, called items that have ever been used as a unit information of an application domain. A set $X = \{i_j, \dots, i_k\} \subseteq L$, ($j \leq k$ and $1 \leq j, k \leq n$) is called a pattern. A transaction $T = (tid, Y)$ is a couple where tid is a transaction-id and Y is a pattern. If $X \subseteq Y$, it is said that T contains X or X occurs in T . A transactional database DB over L is a set of transactions and $|DB|$ be the size of DB , i.e. total number of transactions in DB . The support of a pattern X in DB is the number of transactions in DB that contains X . A pattern is called frequent if its support is no less than a user given support threshold min_sup, δ , with $0 \leq \delta \leq |DB|$. Given δ and a DB , discovering the complete set of frequent patterns in DB , say F_{DB} is called the frequent pattern mining problem.

We discuss the preliminaries and step-by-step construction mechanism of our CP-tree here. In general, CP-tree achieves a frequency-descending structure by capturing part-by-part data from the database and dynamically restructuring itself after each part by using efficient tree restructuring mechanism. Like FP-tree, to facilitate the tree traversal it maintains an item list, say, I-list. The construction operation mainly consists of two phases: *Insertion phase*, that inserts (similar to FP-tree technique) transaction(s) into CP-tree according to current sort order of I-list and updates frequency count of respective items in I-list; and *Restructuring phase*, that rearranges the I-list according to frequency-descending order of items and restructures the tree nodes according to new I-list. These two phases are executed alternatively; starting with *Insertion phase* (with the first part of DB) and finishing with *Restructuring phase* (after the last insertion) at the end of DB .

Fig. 1 shows a transaction database and step-by-step construction procedure of CP-tree. For the simplicity of description we assume that the *Restructuring phase* is executed after inserting every three transactions and the first *Insertion phase* will follow item-appearance order of items. For simplicity of figures we do not show the node traverse pointers in tree, however, they are maintained in a fashion like FP-tree does.

Fig. 1(b) shows the exact structures of the tree and I-list after inserting transactions 10, 20, and 30 in item-appearance order. Since the tree will be restructured after every three transactions, the first *Insertion phase* ends here initiating the first *Restructuring phase*. The *Restructuring phase*, at first, rearranges the items in the I-list in frequency-descending order then, restructures the tree according to that order as shown in Fig. 1(c). It can be noted that items having higher count value are arranged at the upper most portion of the tree; therefore, CP-tree at this stage is a frequency-descending tree. The next *Insertion phase* (for transactions 40, 50, 60) will follow the I-list order of {a, b, d, c, e, f} instead of previous order of {c, a, e, b, d, f}. Fig. 1(d) and Fig. 1(e) respectively present the trees after second *Insertion phase* and *Restructuring phase*. The final frequency-descending CP-tree we get by performing the *Insertion phase* and *Restructuring phase* for last three transactions as shown in Fig. 1(g).

Fig.1(h) shows a lexicographic CanTree containing more nodes with respect to CP-tree for the same dataset. Usually databases share common prefix patterns among the transactions; therefore, the size of CP-tree is usually much smaller than its *DB* and bounded by the size of *DB*. Since CanTree does not guarantee of a frequency-descending tree, generally the size of CP-tree will be smaller than that of CanTree. Once CP-tree is constructed, using FP-growth mining technique F_{DB} can be mined for any value of support threshold δ by starting from the bottom most item in I-list having count value $\geq \delta$.

One of the two primary factors to affect the performance of CP-tree is effectively switching to *Restructuring phase*. Too much or too few restructuring operations both

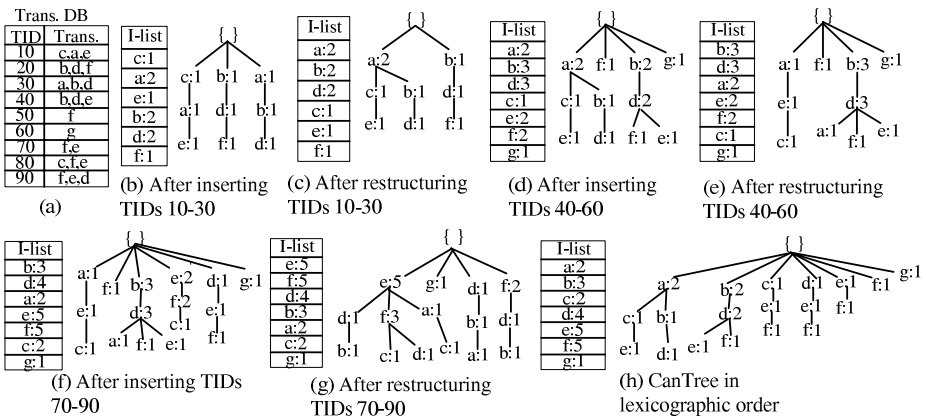


Fig. 1. Construction of CP-tree and CanTree

may lead to poor performance. Therefore, it can be initiated (i) after each user-given fixed sized slot, or (ii) when combined displacement of top- K items in I-list exceeds a given threshold.

2.1 Tree Restructuring

The other performance factor is tree restructuring mechanism. Existing Path adjusting method (PAM), proposed in [4], sorts nodes of a prefix-tree by using bubble sort technique. Any node may be split when it needs to be swapped with any child node having count smaller than that node. Otherwise, simple exchange operation between them is performed.

We propose a new tree restructuring technique called Branch sorting method (BSM) that, unlike PAM, restructures by sorting unsorted paths in the tree one after another and the I-list in frequency-descending order. We revisit the prefix-tree of Fig. 1(b) constructed based on first three transactions of Fig. 1(a), where I-list order $\{c:1, a:2, e:1, b:2, d:2, f:1\}$ is not in frequency-descendent order. To restructure the tree to such order, the I-list is sorted first to $\{a:2, b:2, d:2, c:1, e:1, f:1\}$ order. Secondly, tree restructuring starts with the first path in the first branch say, $\{c:1 \rightarrow a:1 \rightarrow e:1\}$. Since the path is not sorted according to new I-list order, it is removed from the tree, sorted (using merge sort technique) into a temporary array and then again inserted into tree in $\{a:1 \rightarrow c:1 \rightarrow e:1\}$ order. All unsorted paths in other remaining branches are processed using the same technique. If any path is found sorted (e.g., the path of the last branch), it is not sorted, rather merged with previously processed common sorted path (if any). Thus, with the processing of the last path the restructuring of the tree is completed and we get the frequency-descending tree of Fig. 1(c).

The performance of PAM largely depends on *degree of displacement* (DD) among items between two I-lists, since swapping two nodes takes bubble sort cost of $O(n^2)$, where n is the number of nodes between them. On the other hand, BSM uses merge sort approach with a complexity of $O(n \log_2 n)$ (n being the number of items in path), therefore, the DD is immaterial on its performance. Hence, it is not suitable to use PAM when the DD is reasonably high. However, BSM might be a better candidate in such cases, since it performs almost evenly on variations of DD. Moreover, its sorted path handling feature reduces not only the number of sorting operations but also the size of data to be sorted. In summary, during tree restructuring a somewhat dynamic manner can initiate the switching between two methods based on the value of DD.

3 Experimental Results

We performed comprehensive experimental analysis on the performance of CP-tree on several synthetic and real datasets. However, in the remaining part of this section, due to the space constraint we only report the results on two real dense (*chess* and *mushroom*) and one synthetic sparse (*T10I4D100K*) datasets. All programs are written in Microsoft Visual C++ 6.0 and run on a time sharing environment with Windows

XP operating system on a 2.66 GHz machine with 1 GB of main memory. Runtime includes tree construction, tree restructure (for CP-tree only) and mining time.

Table 1 shows required time for both BSM and PAM on increase of *DB* size and that of sorting frequency for *T10I4D100K* and *chess* datasets. Results indicate that the overall restructuring efficiency notably increases on increase of *DB* size in BSM and when applied phase-by-phase (i.e. slotted) on *DB* in PAM. However, the combined approach outperforms each approach in phase-by-phase progress. Therefore, we adopt the combined approach where switching depends on the value of *DD*.

Since it has been shown in [3] that CanTree outperforms other similar algorithms say, AFPIM, CATS tree, we only state the performance comparison of CP-tree with CanTree. To generalize the performance comparison we compare CP-tree with three versions of CanTree; lexicographic order (CT^l), reverse lexicographic order (CT^r), and appearance order (CT^a). As shown in Table 2 for both datasets *T10I4D100K* and *mushroom*, restructuring time for CP-tree appears to be an overhead. However, in spite of this cost, CP-tree significantly outperforms all versions of CanTree on overall runtime due to dramatic reduction in mining time. Fig. 2 reports that CP-tree significantly outperforms CanTree on overall runtime for various *min_sup* values.

The last row of Table 2, that shows memory consumption of the algorithms, indicates that size of CanTree varies on data distribution in transactions and order of items in tree. However, size of CP-tree is independent on such parameters and it is much smaller than all versions of CanTree designed in our experiments.

Table 1. Tree restructuring approach comparison (required time in second)

Restructure approaches	chess						T10I4D100K					
	DB size (K)			No. of slots (slot size = 1K)			DB size (K)			No. of slots (slot size = 20K)		
	1	2	3	1	2	3	20	60	100	1	3	5
BSM	1.02	2.50	4.20	1.02	3.0	6.5	5.86	28.59	65.19	5.86	24.98	60.66
PAM	1.34	3.50	6.98	1.34	1.53	1.89	11.83	69.78	157.41	11.83	15.41	21.05
Combined	--	--	--	1.02	1.22	1.56	--	--	--	5.86	9.47	15.14

Table 2. CP-tree Vs CanTree time and memory comparison

	T10I4D100K ($\partial = 0.04$)				mushroom ($\partial = 0.15$)			
	CT ^a	CT ^l	CT ^r	CP-tree	CT ^a	CT ^l	CT ^r	CP-tree
Construction time (s)	58.88	61.67	57.09	61.86	5.66	4.83	4.58	5.72
Restructure time (s)	--	--	--	19.11	--	--	--	1.89
Mining time (s)	218.25	679.56	824.22	0.44	40.53	62.77	53.19	20.67
Total time (s)	277.11	741.23	881.31	81.41	46.19	67.59	57.77	28.28
Memory (MB)	14.51	14.97	14.99	14.29	0.95	0.70	0.56	0.50

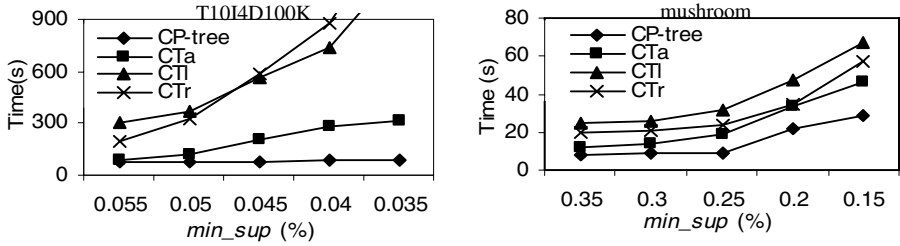


Fig. 2. Runtime comparison

4 Conclusions

We have proposed CP-tree that dynamically achieves frequency-descending prefix-tree structure with a single-pass by applying tree restructuring technique and considerably reduces the mining time. We also proposed Branch sorting method, a new tree restructuring technique, and presented guideline in choosing the values for tree restructuring parameters. We have shown that despite additional insignificant tree restructuring cost, CP-tree achieves a remarkable performance gain on overall runtime. Moreover, the easy-to-maintain feature and property of constantly capturing full database information in a highly compact fashion facilitate its efficient applicability in interactive, incremental and stream data.

References

1. Agrawal, R., Imielinski, T., Swami, A.: Mining Association Rules Between Sets of Items in Large Databases. In: SIGMOD, pp. 207–216 (1993)
2. Han, J., Cheng, H., Xin, D., Yan, X.: Frequent Pattern Mining: Current Status and Future Directions. *Data Min. Knowl. Disc. 10th Anniversary Issue* (2007)
3. Leung, C.K., Khan, Q.I., Li, Z., Hoque, T.: CanTree: A Canonical-Order Tree for Incremental Frequent-Pattern Mining. *Knowledge and Information Systems* 11(3), 287–311 (2007)
4. Koh, J.-L., Shieh, S.-F.: An Efficient Approach for Maintaining Association Rules Based on Adjusting FP-tree Structures. In: Lee, Y., Li, J., Whang, K.-Y., Lee, D. (eds.) DASFAA 2004. LNCS, vol. 2973, pp. 417–424. Springer, Heidelberg (2004)
5. Han, J., Pei, J., Yin, Y.: Mining Frequent Patterns without Candidate Generation. In: *International Conference on Management of Data* (2000)
6. Cheung, W., Zaiiane, O.R.: Incremental Mining of Frequent Patterns without Candidate Generation or Support Constraint. In: *Seventh International Database Engineering and Applications Symposium (IDEAS)* (2003)
7. Agrawal, R., Srikant, R.: Fast Algorithms for Mining Association Rules. In: *International Conference on Very Large Databases* (1994)

Combining Context and Existing Knowledge When Recognizing Biological Entities – Early Results

Mika Timonen and Antti Pesonen

VTT, Box 1000
FI-02044 VTT, Finland
{Mika.Timonen, Antti.Pesonen}@vtt.fi

Abstract. Entity recognition has been studied for several years with good results. However, as the focus of information extraction (IE) and entity recognition (ER) has been set on biology and bioinformatics, the existing methods do not produce as good results as before. This is mainly due to the complex naming conventions of biological entities. In our information extraction system for biomedical documents called OAT (Ontology Aided Text mining system) we developed our own method for recognizing the biological entities. The difference to the existing methods, which use lexicons, rules and statistics, is that we combine the context of the entity with the existing knowledge about the relationships of the entities. This has produced encouraging preliminary results. This paper describes the approach we are using in our information extraction system for entity recognition.

Keywords: entity recognition, entity classification, information extraction, bioinformatics.

1 Introduction

One important problem in recent years in the field of information extraction has been how to classify, i.e., recognize, the found entities. A lot of research has been done in this area with very good results in different domains. However, in difficult domains such as biology there is still room for improvement.

There are three main reasons why entity recognition (ER) is a real problem in biology. First, it is not possible to use simple text matching algorithms since there is no dictionary which contains a comprehensive set of biological entities. Second, in biology the same word can mean different things depending upon the context. For example, ferritin can refer to a biological substance and a laboratory test. Third, many biological entities have synonyms (e.g., PTEN and MMAC1 refer to the same gene). [2]

The methods that tackle these problems are generally based on three different approaches: lexicons, rules and statistics. Lexicon-based methods use large dictionary which contains as comprehensive set of relevant names as possible. Hanisch et al. [3] describe a lexicon-based method which uses a large dictionary of gene and protein names and semantically classified words that tend to appear in context with these names. Rule-based methods usually use part-of-speech (POS) tagger, which is considered as the most basic form of linguistic corpus annotation. The POS information

can be used for rule conditions along with morphological clues and indicator words. Tanabe and Wilbur [4] describe their rule-based method called AbGene which is considered as one of the most successful rule-based system [2]. Chang et al. [1] present a statistical approach they used when creating the GAPSCORE system. They used syntax, appearance, morphology and context in their method by quantifying them for each gene and non-gene as a numerical vector. They used the vectors to train a classifier which was used to identify new words by scoring them based on the similarities to the previously observed training set. Cohen and Hersh have written a survey [2] of current work in biomedical text mining in which they give more extensive overview of these ER methods.

The method presented in this paper differs considerably from other ER methods as it does not use natural language processing techniques or other commonly used ER techniques. Instead, our method uses both existing knowledge and the context of the entity.

The method has some characteristics of the existing methods: it reminds of the lexicon-based method described by Hanisch, et al. [3] as we use the knowledge base partly as a lexicon. Also, the use of the context has been studied in ER in some extent. The major difference is the way we use the lexicon and the context: instead of only using the dictionary for finding matching names we use also the knowledge how the entities are related to each other in the knowledge base. In this process the context of the entity plays an integral part.

This approach has benefits compared to other ER solutions. First, it does not use complex natural language processing techniques making it simpler. Second, this method is easy to implement. And third, in the case of several entities sharing the same name, our approach can identify the entities more precisely.

We have done preliminary tests in the domain of type 1 diabetes. As the knowledge base is still fairly small, more complete tests cannot be done yet. However, the early results are encouraging as our method can recognize the entities most of the time if they can be found from the knowledge base.

In order to fully understand our approach the Ontology Aided Text mining system OAT is described in next Section. Section 3 gives a full description of our method for entity recognition, which is then evaluated in Section 4. Finally, in Section 5 we give a conclusion.

2 Background

We developed Ontology Aided Text mining system (OAT) to assist biologists to automatically collect knowledge about biological entities relevant to their studies. The goal was to create a knowledge base which describes these biological entities and their relationships to one another. By presenting the relationships between the entities the knowledge base can be used for finding interesting paths from one entity to another making the knowledge usable for example for drug discovery. Figure 1 shows an example of the knowledge that can be found from the OAT knowledge base.

The knowledge is extracted from scientific articles using OAT's information extraction module. The module extracts subject-predicate-object-triplets that hold relevant information about the domain.

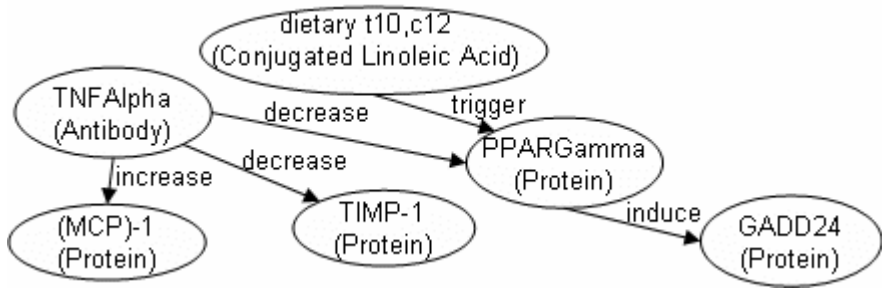


Fig. 1. Example of the knowledge stored to the OAT knowledge base

After information extraction, the triplets are manually checked and verified as being correct or false. This phase is needed because some of the knowledge IE process produces may be noisy and incorrect. Some of the knowledge could be automatically reasoned to be correct or incorrect by using existing knowledge but it is impossible to do this to all the new entities OAT collects. That is why we need to get input from an expert who in this case is a biologist.

If the biologist accepts a triplet he must classify the entities in the triplet to concepts in the ontology, i.e., recognize the entities. This classification is important because it affects the later usability of the knowledge base.

Even though experts can recognize the entities quite well they misclassify them surprisingly often. This problem is present especially with genes that have several names and when a name or a symbol can refer to several genes [3]. In order to make the job easier for biologists we developed a method that narrows down the list of possibilities and suggests the most probable concept from the ontology. This method is described in detail in the following section. More complete description of OAT is given in [5].

3 Entity Recognition

The main focus of OAT is to extract biological knowledge from different data sources and store them to a knowledge base called OAT knowledge base (OATkb). OATkb consists of instances of concepts that are defined in an ontology called OAT ontology (OATo). OATo consists of four components: biological concepts, taxonomy which organizes the concepts into a parent-child hierarchy, relationships which define how the instances of the concepts can relate to one another, and axioms which are used for reasoning. When new entities are added to the knowledge base they must be identified, i.e., mapped to a concept described in the ontology.

The method we have developed for identification of entities gives scores to possible classifications (concepts) for the entity. As a result, the method produces a set of concepts described in the ontology, which all have scores to portray the belief that the entity is of that concept.

The intuition behind our method is that the unclassified entity is likely to be similar with the previously classified entities that are similarly related to the *context entity*. Here, the context entity refers to the entity which appears in the same triplet, i.e., in the context, with the unclassified entity. If we know the concepts of the entities directly related to the context entity, i.e., its neighbors¹, we can use these concepts to deduce the concept of the unclassified entity. For example, if we have a triplet TNFAlpha–decrease–PPARGamma and from the knowledge base we can find that all the neighbors of TNFAlpha with *decrease* relationship are *Proteins*, it is highly likely that the unclassified entity PPARGamma is also a *Protein*.

The process of entity recognition can be divided into three tasks which all produce results that are used later in the ER process. First step is to check the lexicon for the given entity. This lexicon is the knowledge base we have populated with the triplets from previous processes. The query produces a set of concepts that have been previously assigned to the entity of the same name. We call this set a *lexicon set*. As the problem of homonyms is relevant in biology, the set may hold several different concepts. Now, we can calculate the distribution of the concepts. For example, if the entity e has been assigned to a concept X 65 times, concept Y 20 times and concept Z 15 times, the distribution is $(e, X) = 0.65$, $(e, Y) = 0.2$, $(e, Z) = 0.15$.

As the type of the relationship and its direction are relevant, we must take them into consideration when calculating the score for possible concepts for unclassified entities. For instance, if we have several facts that state entity A inhibits entities of *Protein*, and activates entities of *Vitamin*, and the triplet is A activates B , the score of B for *Vitamin* should be higher than score for *Protein*. Also, the direction of the relationship is relevant; if the unclassified entity is the object in the triplet we should use only in-neighbors² and if it is the subject, only out-neighbors should be taken into consideration. In other words, the triplets are not symmetric, i.e., it does not apply that if entities have a relationship $a \rightarrow b$ there would automatically be a relationship $a \leftarrow b$.

The second step is to check the neighborhood of the context entity. For instance, if we have the example triplet TNFAlpha–decrease–PPARGamma and we want to classify PPARGamma, we will check the neighborhood of the context entity TNFAlpha. When checking the neighborhood the method takes into consideration only the entities that are TNFAlpha's out-neighbors and are related to it with *decrease* link. This produces a set of concepts which is called a *neighborhood set*.

The final step is to combine the results of the lexicon set and the neighborhood set. For this, we use the following equation:

$$\text{score}(\text{Concept}) = \frac{\alpha \times nc + lc + w}{te}, \quad (1)$$

where $w = (nc \times lc) / (tc + tl)$, nc is entity count in the neighborhood set for the given concept, lc is the entity count from the lexicon set for the given concept, and te is the total entity count for all the concepts in both of the sets. The total entity count

¹ Neighborhood of an entity consists of entities that are directly related to the entity in the knowledge base. For instance, if $A \rightarrow B$ then A is in B 's neighborhood and vice versa.

² Formally, an in-neighbor c of a has a directed link $c \rightarrow a$ and an out-neighbor c of a has directed link $c \leftarrow a$. For example, in Figure 1 the neighbors of TNFAlpha are out-neighbors.

$te = \alpha \times mc + tlc$, where mc is total count of entities in the neighborhood set and tlc is total count of entities in the lexicon set. The equation uses the constant α to multiply nc and mc because we want to give more emphasis to the context. This variable can be changed depending on how much emphasis is given. In our tests we are using $\alpha = 2$. w is used to give more emphases for the cases where the entity is found both from the lexicon and the neighborhood sets.

For example, if we are calculating the result for the entity concept pair (PPARGamma, *Protein*), and from the knowledge base we get $nc = 15$ (context entity's neighborhood has 15 entities of *Protein*), $lc = 30$ (PPARGamma has been classified as *Protein* 30 times), $mc = 50$ (there are 50 entities in the neighborhood set), and $tlc = 50$ (the entity PPARGamma has been classified 50 times). We get the following result:

$$score(Protein) = \frac{2 \times 15 + 30 + 7.5}{2 \times 50 + 50} = 0.45 .$$

It should be noted that the equation takes the volume of knowledge into consideration. In other words, if there are a lot of entities in the lexicon set and few in the neighborhood set, the score is based mainly on the distribution of concepts in the lexicon.

4 Evaluation

We have done a set of preliminary tests which are for the most part theoretical. Table 1 shows a few test cases which we have used to assess the scores for entity concept pair (e, C) in different theoretical situations.

Table 1. Results for different theoretical cases for entity e and concept C pair (e, C). In these calculations $mc = 100$ and $tlc = 100$.

	Case 1.	2.	3.	4.	5.	6.	7.	8.	9.
nc	20	80	80	0	50	25	75	65	0
lc	80	20	0	80	50	100	0	50	50
Score	0,4267	0,6267	0,53	0,267	0,542	0,542	0,5	0,654	0,167

In cases 1 and 2 we demonstrate how our method emphasizes the context: in both cases there are the same total amount of entities in the sets but the score differs quite a lot. In case 1, we can see that in most cases the entity has been classified as C (80% of the times) but it has appeared with the context entity in the given context only 20% of the times. In case 2, we can see that when the entity appears in the given context, it has been usually classified as C (80% of times). Case 2 gets a higher score as it

conforms to our assumption: the unclassified entity is likely to be similar with the previously classified entities located in the context entity's neighborhood.

The same can be seen from cases 3 and 4. They show how the score changes when there are no entities in either of the sets. This represents again how we consider the context more important factor than the lexicon. However, this approach brings also problems. Some may argue that the case 4 should get higher score than the case 3. In other words, the lexicon should be valued higher than the context since there are not any instances of this concept in the neighborhood but in the lexicon the entity has been assigned to that concept most of the time.

In the cases 5 and 6 there is quite a lot of evidence from both neighborhood and the lexicon which makes the concept a good candidate. However, in the case 7, there is no evidence from the lexicon, i.e., the entity has never been classified to that concept but the entities in the context's neighborhood are mainly of that concept. In this case, intuitively the case 6 should receive higher score than case 7. This presents why we introduced the variable w .

Finally, the cases 8 and 9 show an example situation of two entities sharing the same name. As there are 100 instance of the entity in the lexicon and it has been assigned to two different concepts both 50 times, it is impossible to know without the context which the correct classification is. But, in the case 8 there are 65 entities of the concept in the neighborhood set which makes that concept much more probable over the concept in case 9.

5 Conclusions and Future Work

We have described a novel method for entity recognition which utilized existing knowledge about the relationships and the context of the entities. This method does not use complex natural language processing techniques which are difficult to implement and not always that reliable. In contrast, this method is quite simple and easy to implement. Also, the method tackles the problem of complex naming conventions in biology, especially the problem of several entities sharing the same name. As we take the context and the existing knowledge about relationships into consideration, we can classify the entities more precisely

In the future, we are planning to introduce improvements to our entity recognition. In the early stages of operational knowledge base there might be situations when the entities cannot be found from the knowledge base and therefore cannot be recognized. But as the knowledge base grows this situation becomes less and less common. Also, as we use the context, it is more likely that at least one of the entities in the triplet can be found from the knowledge base making the identification easier.

The method itself could also be improved. We could introduce a taxonomy for predicates which can be utilized when the neighborhood set is created. For instance, if a triplet has an *increase* relationship, the similar and closely related relationships in the neighborhood (e.g., *increment*, *multiplicate*, etc.) should be taken into consideration also. In this case, we could lower the score as the similarity of predicates decreases. But the first thing is to do complete tests to verify our theory in practice.

References

1. Chang, J., Schutze, H., Altman, R.: GAPSCORE: Finding Gene And Protein Names One Word at a Time. *Bioinformatics* 20(2), 216–225 (2004)
2. Cohen, A., Hersh, W.: A survey of current work in biomedical text mining. *Briefings in Bioinformatics* 6, 57–71 (2005)
3. Hanisch, D., Fluck, J., Mevissen, H.: Playing Biology's Name Game: Identifying Protein Names in Scientific Text. In: *Pacific Symposium on Biocomputing*, vol. 8, pp. 403–414 (2003)
4. Tanabe, L., Wilbur, W.: Tagging Gene And Protein Names in Biomedical Text. *Bioinformatics* 18(8), 1124–1132 (2002)
5. Timonen, M.: *Implementation of Ontology-Based Biological Knowledge Base*, Master's Thesis, Department of Computer Science, University of Helsinki, Helsinki (2007)

Semantic Video Annotation by Mining Association Patterns from Visual and Speech Features

Vincent S. Tseng, Ja-Hwung Su, Jhih-Hong Huang, and Chih-Jen Chen

Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.
tsengsm@mail.ncku.edu.tw

Abstract. In this paper, we propose a novel approach for semantic video annotation through integrating visual features and speech features. By employing statistics and association patterns, the relations between video shots and human concept can be discovered effectively to conceptualize videos. In other words, the utilization of high-level rules can effectively complement the insufficiency of statistics-based methods in dealing with broad and complex keyword identification in video annotation. Empirical evaluations on NIST TRECVID video datasets reveal that our proposed approach can enhance the annotation accuracy substantially.

1 Introduction

Recent advanced multimedia capturing technologies enable the recordings of our colorful living. To support multimedia retrieval applications, video annotation is an important issue for searching the huge amount of multimedia data in the repositories. Typically, a video can be divided into several scenes/stories and each scene contains a set of shots composed of time-split/similarity-split image frames. From these sequential frames, a representative image frame is defined as a key-frame. Due to the relations and rich contents of these sequential images, the annotation method for a video is very different from that for a single image [6].

In past studies, association rules were used to annotate a video but the effects are not satisfactory since the generated association rules may be too specialized to fit for a wide range of videos. That is to say, if the rule set is too small, we may not get sufficient matching rules to support video annotation. Hence, annotations by using only the specialized association rules will possibly lead to high errors. With more considerations than association rules, the work in [8][9][10] took account of temporal continuity and used event detection to index and explore sequential association rules in the sequential key-frames. However, the results of sequential association rules are also limited with the range of video types. In addition to rule-based solutions mentioned above, CRM (Continuous Relevance Model) [1][4] is a classic statistics-based method for annotating videos. It segments each sequential key-frame into several rectangle regions and then extracts the referred visual features from these segmented regions. The annotations of each image are yielded soon after calculating the related probabilities with Gaussian Mixture Function. By exploiting the temporal continuity of video

sequences and assuming Markovian property between image frames, DBNs (Dynamic Bayesian Networks) proposed by Luo *et al.* [5] projected low-level features onto high-level concept space. In [6][7], Tseng *et al.* proposed hybrid methods for video annotation by integrating statistics-based and rule-based methods.

In this paper, we present a hybridized solution for semantic video annotation by exploiting multi-contents of videos, namely visual features and speech features. The major contribution of the proposed method is that visual features and speech features are considered simultaneously to enhance the accuracy of video annotation. The empirical evaluations reveal that the proposed approach can effectively assign the relevant keywords to the video shots. The rest of this paper is organized as follows. In section 2, we demonstrate our proposed method for annotating videos in great detail. Experimental evaluations of the proposed methods are illustrated in section 3. Finally, conclusions and future work are stated in section 4.

2 Proposed Method

The proposed method is basically extended from the work in [6][7]. As illustrated in Figure 1, the whole procedure contains two types of prediction models: Rule-based model ($Model_{Vseq}$, $Model_{Vasso}$ and $Model_{Sasso}$) and Statistics-based model ($Model_{CRM}$). The details are described as the following subsections.

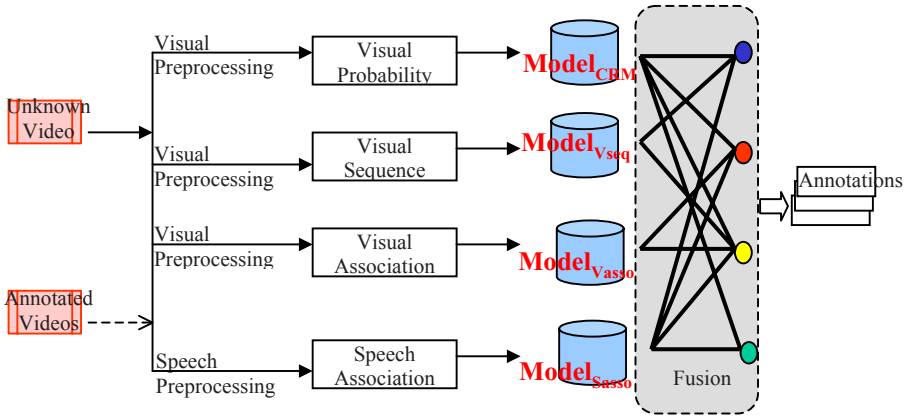


Fig. 1. Workflow of the proposed approach

2.1 Preprocessing Operation

Functionally, the preprocessing operation can be viewed as a foundational stage that generates the necessary information used in the training phase and prediction phase.

- Visual Preprocessing.** This process is primarily for visual-based models. First, we perform shot detection to divide a video and combine several sequential shots to form a scene. Then, the representative key-frame of each shot is determined. Second, each

shot of a video has to be further divided into $m*n$ rectangle regions. These regions will be the basic elements for $\text{Model}_{\text{CRM}}$. Third, scalable color and homogeneous texture are extracted from both the un-segmented shots and the segmented regions.

- **Speech Preprocessing.** This is a process for constructing speech-based model. First, after the scene division, automatic speech recognition (ASR) [2] is triggered to transform audio features into text descriptions shot by shot for each divided scene. Second, IR techniques including Removing stop-words and Stemming words are employed to filter the usable speech words. Third, we utilize JwordNet [3] to project these filtered keywords onto the specific keyword space regulated by NIST.

2.2 Training Phase

This phase is primarily concerned with the generation of four major models, namely $\text{Model}_{\text{CRM}}$, $\text{Model}_{\text{Vasso}}$, $\text{Model}_{\text{Vseq}}$ and $\text{Model}_{\text{Sasso}}$. In this phase, three rule-matching matrices for building $\text{Model}_{\text{Vasso}}$, $\text{Model}_{\text{Vseq}}$ and $\text{Model}_{\text{Sasso}}$ and a keyframe-matching matrix for building $\text{Model}_{\text{CRM}}$ are yielded by visual association rules, speech association rules and visual features, respectively.

- **Construction of $\text{Model}_{\text{Sasso}}$.** The first task in this model [7] is to establish a transaction table for $\text{Model}_{\text{Sasso}}$ based on a presetting “*shot window*”. A shot window contains a sequence of shots, and the window slides along the scene. The target keywords (annotations) of each central shot of each sliding shot window can be used to form a transaction with the speech keywords of each shot of each sliding shot window. Assume that a shot window consists of $2z+1$ shots where $z \geq 0$ and win_0 is the central shot. Then the target keywords annotated in win_0 and the speech keywords to left z shots and right z shots form $2z+1$ transactions.

- **Construction of $\text{Model}_{\text{Vseq}}$.** In this model [6], we first discover the frequent itemsets from the scene-transaction table. These generated frequent itemsets can be viewed as association rules directly since temporal continuities are inherent in them. For example, the sequential association rule ($A \rightarrow B$) can be derived from frequent sequential itemset $\{A, B\}$. Next, each generated frequent itemset is used to seek for its associated keywords and the frequencies of keywords referred each frequent itemset are used to form the rule-matching matrix $PL_{X \rightarrow W}$.

- **Construction of $\text{Model}_{\text{Vasso}}$.** As mentioned above, the major difference between $\text{Model}_{\text{Vasso}}$ and $\text{Model}_{\text{Vseq}}$ is that $\text{Model}_{\text{Vasso}}$ ignores the temporal continuities of the frequent patterns [6]. In other words, the duplicate items have to be pruned in each tuple of scene-transaction table.

2.3 Prediction Phase

As stated in the training subsection, three visual matching matrices are derived from three visual-based models that can represent the relations between key-frames and keywords and those between rules and keywords. Besides, speech association rules can reflect the relatedness between speeches and keywords. These derived matrices and rules, actually, can offer video annotation a great support.

- **Prediction by $\text{Model}_{\text{CRM}}$.** This model is mainly based on the CRM method [1][4].

- **Prediction by Model_{Vseq}**. As soon as the scenes containing a sequence of unknown shots are sequentially received in our method, each shot within a scene has to be encoded first by computing the similarity (Euclid Distance) between the shot and the clusters generated in the training phase. The prediction algorithm is discussed in [6].
- **Prediction by Model_{Vasso}**. In some cases, the temporal continuity of shots is not an essential factor for video annotation since we can get the better results without considering the temporal continuity. Moreover, due to the temporal continuity is skipped, the related rule-matching matrix derived from Model_{Vasso} differs from that derived from Model_{Vseq}. The results are accordingly changing [6].
- **Prediction by Model_{Sasso}**. In our method, Model_{Sasso} can really convey more important information than those of the other models since embedded speeches are always stably related to the referred shots. In this prediction [7], each shot is first pre-processed to generate its own speech keyword set. Next, these shots are sequentially predicted by looking for the relevant rules which left-hand itemsets are matched with the speech keywords within a specified sliding window. Finally, the average confidence of each annotation for each shot is generated.
- **Prediction by Fusion Models**. To integrate different viewpoints on four special prediction models, we design multiple fusion approaches to examine the annotation accuracy. Basically, the design of each fusion model is to take Model_{CRM} as the foundational model and the others as the auxiliary models. Due to the high variations of videos, it is hard to represent all kinds of video just by the finite rules. Hence, the primary aim of this design is to avoid the missing-rule problem in rule-based models. In other words, by employing Model_{CRM}, we can annotate any shot with at least one keyword whether joining with the rule-based models or not. Finally, the derived result of each prediction model is on the basis of its normalized Z-probability. The fusion models are defined as follows:

$$\text{Fusion 1} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vseq}}$$

$$\text{Fusion 2} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vasso}}$$

$$\text{Fusion 3} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vseq}} + \text{Model}_{\text{Vasso}}$$

$$\text{Fusion 4} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vseq}} + \text{Model}_{\text{Sasso}}$$

$$\text{Fusion 5} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vasso}} + \text{Model}_{\text{Sasso}}$$

$$\text{Fusion 6} = \text{Model}_{\text{CRM}} + \text{Model}_{\text{Vseq}} + \text{Model}_{\text{Vasso}} + \text{Model}_{\text{Sasso}}$$

3 Empirical Evaluation

The experimental data came from the collection of TREC Video Retrieval Evaluation (TRECVID) provided by the National Institute of Standards and Technology (NIST). From the TREC videos, we chose four CNN and four ABC news videos as our experimental data. The total duration of the experimental data is around 233 minutes and the data size is about 3158MB. Moreover, there are 161 scenes and 1414 shots split in this experimental data set. The evaluation was investigated in terms of *precision*. In our experiments, we adopted the 8-fold approach to carry out the evaluations. That is, seven videos took turns as a testing video and the others were taken as training videos. Figure 2 shows that all of rule-based models, Model_{Vseq}, Model_{Vasso} and Model_{Sasso}, outperform CRM in terms of the precision, and Model_{Sasso} performs better than any other model on average. This indicates that the speech rule-based models can

effectively capture the intra-relations or inter-relations among the shots as we expect. In contrast, annotations by using only visual features encounter higher difficulty in dealing with high variations of visual features in the videos. Figure 3 reveals that the precisions for fusing visual features and speech features are significantly better than

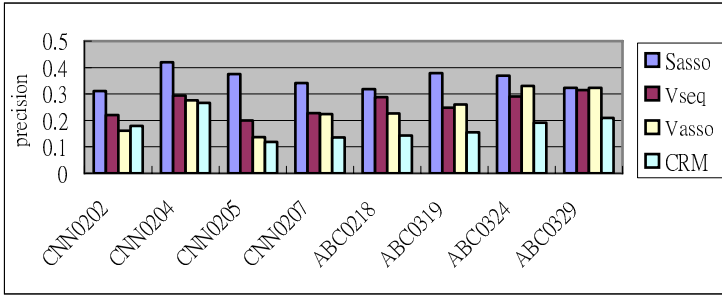


Fig. 2. The precisions of CRM, Model_{Vseq}, Model_{Vasso} and Model_{Sasso}

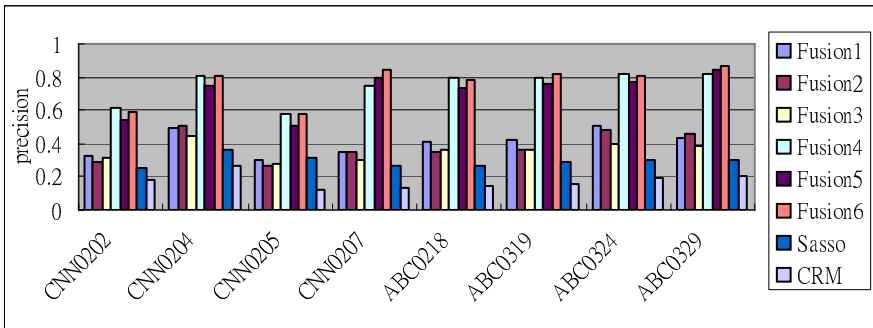


Fig. 3. The precisions of CRM and all fusion models

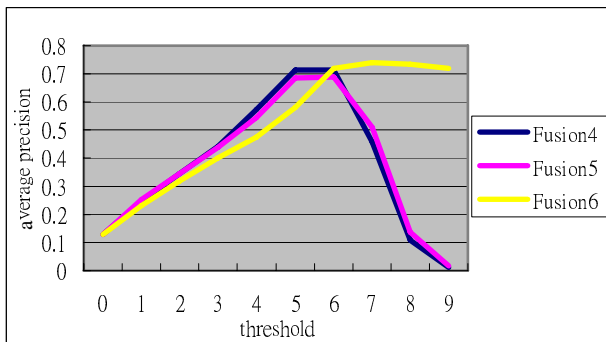


Fig. 4. The average precisions of Fusion4, Fusion5 and Fusion6 under different thresholds

those of the models that consider only visual features or considering speech features individually. In other words, higher precision relies on the integration of all involved individual models. On average, our proposed fusion method, Fusion 6, exhibits the improvements over CRM for about 335% on precision. Figure 4 reveals that Fusion 6 performs stably and outperforms the other hybrid fusion models under different Z thresholds. This delivers that correct answers are adequately strengthened by the integration of all individual models.

4 Conclusions and Future Work

In this paper, we propose a novel method to exploit visual features and speech features for video annotation by integrating statistics and association patterns. The utilization of high-level patterns can effectively complement the insufficiency of visual-based methods in dealing with complex and compound videos. As a result of the experiments, the proposed approach is shown to be very promising for video annotation through the integration of visual features and speech features. In the future, we will further investigate an adaptive fusion method by tuning the weight of each model.

Acknowledgement

This research was supported by Ministry of Economic Affairs, R.O.C. under grant no. 95-EC-17-A-02-51-024, and by National Science Council, R.O.C. under grant no. NSC96-2422-H-006-001.

References

- [1] Feng, S.L., Manmatha, R., Lavrenko, V.: Multiple Bernoulli Relevance Models for Image and Video Annotation. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2*, 1002–1009 (2004)
- [2] Hacioglu, K., Pellom, B.: A Distributed Architecture for Robust Automatic Speech Recognition. In: *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2003)*, April 2003, vol. 1, pp. 328–331 (2003)
- [3] Johar, K., Simha, R.: The George Washington University JWord 3.0, <http://www.seas.gwu.edu/~simhaweb/software/jword/>
- [4] Lavrenko, V., Feng, S.L., Manmatha, R.: Statistical Models for Automatic Video Annotation and Retrieval. In: *Proc. of the International Conference on Acoustics, Speech and Signal Processing* (May 2004)
- [5] Luo, Y., Hwang, J.-N.: Video Sequence Modeling by Dynamic Bayesian Networks: A Systematic Application from Coarse-to-Fine Grains. In: *Proc. of IEEE International Conference on Image Processing* (September 2003)
- [6] Tseng, V.S., Su, J.-H., Huang, J.-H.: A Novel Video Annotation Method by Integrating Visual Features and Frequent Patterns. In: *Proc. of 7th International Workshop on Multimedia Data Mining (held with KDD 2006)*, Philadelphia, Pennsylvania, USA (August 2006)

- [7] Tseng, V.S., Su, J.-H., Chen, C.-J.: Effective Video Annotation by Mining Visual Features and Speech Features. In: Proc. of the third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Taiwan, November 26-28 (2007)
- [8] Zhu, X., Wu, X.: Sequential Association Mining for Video Summarization. In: Proc. of the 4th IEEE International Conference on Multimedia and Expo, Baltimore, USA (July 2003)
- [9] Zhu, X., Wu, X.: Mining Video Associations for Efficient Database Management. In: Proc. of 18th the Internal Joint Conference on Artificial Intelligence, August 2003, pp. 1422–1424 (2003)
- [10] Zhu, X., Wu, X., Elmagarmid, A.K., Feng, Z., Wu, L.: Video Data Mining: Semantic Indexing and Event Detection from the Association Perspective. *IEEE Transactions on Knowledge and Data* 17(5) (May 2005)

Cell-Based Outlier Detection Algorithm: A Fast Outlier Detection Algorithm for Large Datasets

You Wan and Fuling Bian

Research Center of Spatial Information and Digital Engineering,
International School of Software, Wuhan University, 129 Luoyu Road, Wuhan 430079, China
wanyou9@gmail.com

Abstract. Finding outliers is an important task for many KDD applications. We developed a cell-based outlier detection algorithm (short for CEBOD) to detect outliers in large dataset. The algorithm is based on LOF; major difference is CEBOD can avoid large computations on the majority part of dataset by filter the initial dataset. Our experiment shows that CEBOD is more efficient than LOF, and can find outliers in large datasets fast and accurately. A large dataset is loaded into memory by blocks, and the data are placed into appropriate cells based on their values. Each cell holds a certain number of data, which represents the cell's density. Data locate in high density cells and have no nearness relationship with local outlier factor calculation are filtered. And we record these cells' density for the next block of data fill in. The final calculation will be done on those data in low density cells. In this way, we can handle a large dataset which can't be loaded into memory once, improving the algorithm's efficiency by reducing many useless computations. The time complexity of CEBOD is $O(N)$.

Keywords: Outlier Detection, Cell density filtering, Large Datasets.

1 Introduction

An outlier is an observation that deviates so much from other observations as to arouse suspicions that it was generated by a different mechanism^[1]. Outlier detection is an outstanding data mining task, methods to find outliers can be classified into four categories: statistics based^[2], distance based^{[3][4][5]}, density based^{[6][7][8][9]} and cluster based^[10].

To detect different type of outliers efficiently, many algorithms are presented. Knorr and Ng^[3] proposed FindAllOutsD algorithm to handle distance based outlier which has a linear complexity w.r.t. N . Ramaswamy^[4] proposed a partition-based algorithm for mining outliers based on the distance of a point from its k -th nearest neighbor. Bay^[11] found that the nested loops algorithm in conjunction with randomization and a pruning rule can achieve a nearly linear time performance. Jin^[7] proposed a Micro-cluster-based algorithm, which use a clustering algorithm to compress the data.

2 Problem Formulations

Outlier detection is more concern about small patterns on minority part of a dataset. To find outliers efficiently, we can do compression on the initial dataset; focus on the minority of dataset. This can avoid large computations.

Following this idea, we developed our CEBOD algorithm to detect outlier from large datasets. The outliers we found are density based, because calculations to detect these outliers are most concern about their neighbor objects; abandon other objects which have no nearness relationship will not affect the result much. Moreover, cell based approaches are used to identify an object's density, also to find its k nearest neighbors very fast.

Data space is divided into cells, and the number of data locating in a cell is used to identify a cell's density, and those data's density factor. We use a positive number k to set the threshold of each cell's density. If a cell contains more than k objects, it's a High-Density Cell; otherwise Low-Density Cell. Then we filter the high density data which have no relationship to the outlier detection calculation. The computation focuses on the rest data which will be very smaller than the original one.

Some definitions should be given, which can assure the accuracy of result.

Definition 1. temporal-Outlier (t-Outlier)

Let p be an object in the dataset D , and C_1, C_2, \dots, C_n be a partition of Dataset. For any positive integer k , p is a t-Outlier only if: $\forall p \in C_i, C_i.\text{density} < k$. Here, $C_i.\text{density}$ is the number of objects to contained in C_i

Definition 2. non-Outlier (n-Outlier)

An object p is in the dataset D is an n-Outlier only if: $\forall p \in C_i, C_i.\text{density} < k$. Compared with t-Outliers, n-Outliers are distributes in high density areas of dataset. All t-Outliers together with all n-Outliers are the whole dataset.

Definition 3. real-Outlier (r-Outlier)

Let an object p in D and the set T be the unit of all t-outliers, p is an r-Outlier only if: $\forall p \in T, p.\text{lof} > \text{LOF}$. Here, $p.\text{lof}$ is the local outlier factor of object p , and LOF is a positive input parameter. Although r-Outliers belong to t-Outliers, their local density factors are bigger than LOF .

Definition 4. relative-non-Outlier (r-n-Outlier)

Let any object p in dataset D , O be the unit of all n-outliers and p be from the set T , p is an r-n-Outlier only if: $\forall p \in O, p \notin q.\text{knn}$. Noting that $q.\text{knn}$ is the unit of all objects nearest to the object q . An r-n-Outlier is still a n-Outlier, but it has a nearness relationship with t-Outliers.

Definition 5. absolute-non-Outlier (a-n-Outlier)

Let any object p in dataset D , O be the unit of all n-outliers and p be from the set T , the object P is an a-n-Outlier only if: $\forall p \in O, p \notin q.\text{knn}$. Compared with r-n-Outliers, a-n-Outliers are objects that have no nearness relationship with those t-Outliers. They always surrounded by the High-Density Cells, and can be filtered in the first data scan. All r-n-Outliers combined with all a-n-Outliers are the whole n-Outliers.

Definition 6. remain-Dataset(r-Dataset)

Let the set A be the unit of all the a-n-outliers, the r-Dataset $R = \{p | \forall p \in D, p \notin A\}$. The r-Dataset is the set of remaining data in D where all absolute non-outliers are removed from, after the first data scan.

We use Fig 1 to illustrate the definitions above. 188 points in Fig 1 are partitioned by cells. The density threshold k equal to 5, cells have more then 5 points are identified as high density ones, otherwise low density cells. Then, we can divide the dataset into three parts: a-n-Outliers, r-n-Outliers and t-Outliers. 44 data in the center four cells are a-n-Outliers. They are surrounded by high density cells, and have no relationship with the outlier calculation. 83 data in the middle cells around the former four cells are r-n-Outliers. Cells around them contain low density cells. 61 data in the low density cells and labeled as solid square blocks are t-Outliers. On the further computation, some of them may become r-Outliers. So after filtering all the a-n-Outliers, nearly a quarter of data are deleted. And the real computations focus on just one third of the whole dataset, later experiments show the compression of data is always around one third.

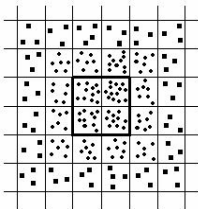


Fig. 1. Example data set

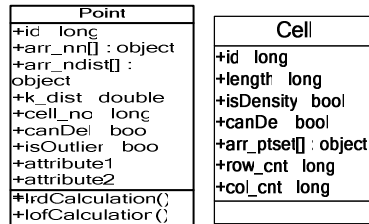


Fig. 2. Two data structures used in CEBOD algorithm

3 Algorithm Description and Complexity Analysis

3.1 Algorithm Description

We first introduce the data structures used in this algorithm and the construction of cells. The algorithm constructed two arraylists of arr_pt and arr_cell, respectively for Point and Cell. Arr_pt is used to store data objects, and arr_cell is used to store cell objects. The structures of Point and Cell are shown in Fig 2. Take two dimensional attributes for example, a Cell's length, row_cnt and col_cnt can be calculated as blow:

$$\text{The domain's } D_length = \max_x - \min_x, D_width = \max_y - \min_y;$$

$$\Rightarrow D_Area = D_length * D_width.$$

$$\text{The cell's total number } m = \lceil N/k \rceil + 1;$$

$$\Rightarrow \text{Cell's length} = \text{square}(D_Area/m),$$

$$\Rightarrow \text{Cell's row_cnt} = (\text{height}/\text{Cell.length}) + 2,$$

$$\Rightarrow \text{Cell's col_cnt} = (\text{width}/\text{Cell.length}) + 2.$$

The algorithm receives as input a dataset DB of N points (with two dimensional attributes for easy explanation), the domains of each attributes in (min_x, min_y, max_x,

max_y), the number b_size represents the size of data loaded into memory one time, the number k of neighbors to consider. And the algorithm contains three major parts shown in Fig 3:

- (1) Loading each data into cells by blocks,
- (2) Using cells to filter the dataset and mark each t-Outlier,
- (3) Calculate each t-Outlier's local outlier factor.

```

Procedure 1 DataInitialize(DB,min_x,min_y,max_x,max_y,b_size){
count=0;
Load DB's data into arr_pnt,
For each data Pi,
  count++;
  If(count%b_size==0)
    Do CellInitialize(startId);
    Do DataFilter();
  End If
Do the last CellInitialize(startId);
Do the last DataFilter();
End For
}
Procedure 2 CellInitialize(startId){
For each data object Pi in arr_pnt, is(startId,arr_pnt.size)
  If Pi is out of the boundary of Domain, P.isOutlier=true;
  Else
    Map Pi into an appropriate Cell Cj,
    set P.cell_no=j, and Cj.arr_ptset.add(i);
  End If
End For
For each cell Cj, j∈[0,arr_cell.size),
  If Cj.isDensity = false
    If (Cj.arr_ptset.size>=k)
      set Cj.isDensity=true,
      set each points in Cj isOutlier attribute equals 0;
    Else If(Cj.arr_ptset.size>0 && Cj.arr_ptset.size<k)
      set each points in Cj isOutlier attribute equals 1;
    End If
  End If
End For
}
Procedure 3 DataFilter{
For each cell Cj, j∈[0,arr_cell.size)
  If Cj.isDensity = false, continue;
  If Cj.canDel = false,
    If Cj's neighbor cells' isDensity attributes are all true,
      Cj.canDel=true;
    If Cj.canDel = true,
      For each data object Pi in Cj,
        set Pi.canDel=true; Cj.arr_ptset.clear;
      End For
    End If
  End If
End For
For each data object Pi in arr_pnt,
  If Pi.canDel = true, remove Pi
End For
}
Procedure 4 knnBuild(){
For each remain data object Pi in arr_pnt,
  If Pi.isOutlier=0, continue;
  If Pi.isOutlier=1, continue;
  //remains all t-Outliers for calculation
  scan the surrounding cells of Pi, including which Pi in,
  sort Pi's k nearest neighbors according to the distance,
End For
For each remain data Pi in arr_pnt,
  If Pi.isOutlier=1, calculate Pi's local reachability density
End For
For each remain data Pi in arr_pnt,
  If Pi.isOutlier=1, calculate Pi's local outlier factor
End For
}

```

Fig. 3. Pseudocode of CEBOD algorithm

For every data remains and marked outlier, procedure 5 lrdCalculation and 6 lofCalculation calculate their local reachability density and local outlier factor. In CEBOD, different from LOF, objects identified as a-n-Outliers are deleted before calculation, we can't get the local reachability density of the r-n-Outliers nearby them, also the local outlier factor of the t-Outliers nearby can't be calculated. To make compensation, we initialize each r-n-Outlier's local reachability density to 0.

3.2 Complexity Analysis

The time complexity of CEBOD is $O(N)$, which is the same as LOF when it uses cell structure. But as the calculation in CEBOD only concerned to about one third of the dataset, so its efficiency is better than LOF several times.

Because the expensive system I/O costs, we must minimize the number of pass over the dataset and page exchange operations, because the expensive system I/O costs. In CEBOD algorithm, if cell space is divided into m parts, the number of dataset passes can be calculated as: $1 + (m-1)/m + (m-2)*m + \dots + 1/m = (m+1)/2$. For example, $m=3$, then the pass over number is two.

The page exchange operations are not exist in CEBOD algorithm, because each time when doing computations, relative data are all exist in the memory.

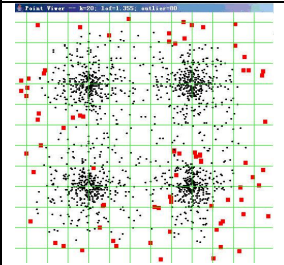
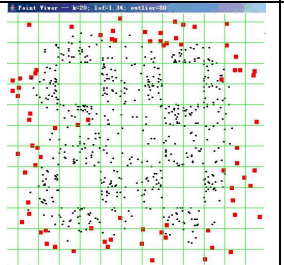
4 Experiment

We implemented the algorithm using Java. The experimental platform is a Pentium IV 2.66GHz-based machine with 512MB main memory.

4.1 Comparison of Accuracy with LOF Algorithm

We use a synthetic dataset which has 1600 two dimensional tuples to test CEBOD’s accuracy. The distribution of this dataset is shown in Table 1 (in the second column), and can be roughly divided into four clusters with some outliers around them. Our purpose is to find the top n% outliers in the dataset.

Table 1. Accuracy compare between LOF and CEBOD

Outlier	LOF	CEBOD	similarity
5%			70%

Experiments prove that two algorithms’ results have a high similarity, especially finding 1% and 2% outliers in a dataset. The 5% outliers result picture is shown in Tab 1. As we adjust the value of local outlier factor, there is an decreasing similarity. LOF found some outliers locate in the middle area of each cluster, which seems a little disorder, but it gives a good demonstration of its ability to find local outliers in different density areas. While, CEBOD is more apt to find outliers locate in the edge area of each cluster, because the central objects of a cluster is filtered before calculating, and its results show a more clear deviation of four clusters.

4.2 Influence of k Value and Data Size on Algorithm Performance

Figure 4 gives an exciting result that different k values have little effect on CEBOD algorithm. Further analysis shows that two factor lead to this. First, the filter operation before k-nn query can filter about 2/3 of the whole data. The k-nn queries in high density areas needs more calculation, more comparison and more sorting operations, especially when k is larger, as LOF shows in Figure 2. Second reason is the cells we divide the datasets. As the increasing of k, cell’s number is decreasing. This decreasing does not affect the filter operation. On the contrary, it can reduce the scan time when calculating k-nn queries.

Then we ran our algorithm on a dataset from kdd99^[12], which contains 250 thousand records. Figure 5 shows the CEBOD has a nearly linear complexity. We didn’t give the LOF’s time cost for large dataset, because in these experiments, the cost time of LOF algorithm is up to thousand second.

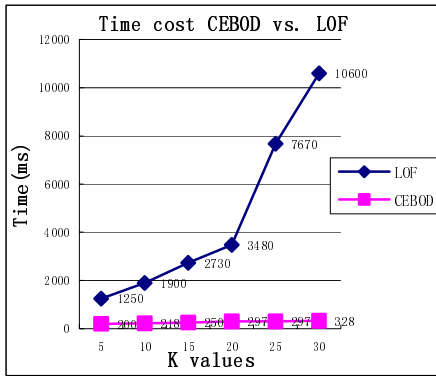


Fig. 4. Runtime of LOF and CEBOD

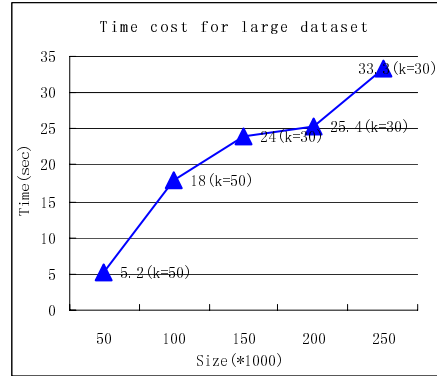


Fig. 5. Time cost for large dataset

5 Conclusions and Future Works

In this paper, we proposed a novel cell based algorithm CEBOD for finding density based outliers. The strength is it avoids computation for most objects. Experimental results prove CEBOD has high accuracy in finding outliers, and can be used for large dataset's outlier detection efficiently.

In ongoing work, we will extend CEBOD to distributed dataset. As far as we know, only distance based methods exist for distributed outlier detection. We will also make comparison between distance based and density based methods.

References

1. Hawkins, D.: Identification of Outliers. Chapman and Hall, London (1980)
2. Barnett, V., Lewis, T.: Outliers in statistical data. John Wiley, Chichester (1994)
3. Knorr, E.M., Ng, R.T.: Algorithms for Mining Distance-Based Outliers in Large Datasets. In: Proceedings of the 24rd International Conference on VLDB, pp. 392–403 (1998)
4. Ramaswamy, S., Rastogi, R., Kyuseok, S.: Efficient algorithms for mining outliers from large data sets. In: Proc. 2000 ACM SIGMOD Int. Conf. on Management of data, pp. 427–438 (2000)
5. Angiulli, F., Pizzuti, C.: Fast Outlier Detection in High Dimensional Spaces. In: Proc. 6th European Conf. on Principles of Data Mining and Knowledge Discovery, pp. 15–26 (2002)
6. Breunig, M.M., Kriegel, H.P., Ng, R.T., et al.: LOF: identifying density-based local outliers. In: Proc. 2000 ACM SIGMOD Int. Conf. on Management of data, pp. 93–104 (2000)
7. Jin, W., Tung, A.K.H., Han, J.: Mining Top-n Local Outliers in Large Databases. In: Proc. ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining, pp. 293–298 (2001)
8. Jiang, S.-Y., Li, Q.-H., Li, K.-L., Wang, H., Meng, Z.-L.: GLOF: a new approach for mining local outlier. In: Int. Conf. Mach. Learn. Cybern., vol. 11, pp. 157–162 (2003)

9. Tang, J., Chen, Z., Fu, A.W.-C., Cheung, D.W.-L.: Enhancing Effectiveness of Outlier Detections for Low Density Patterns. In: Chen, M.-S., Yu, P.S., Liu, B. (eds.) PAKDD 2002. LNCS (LNAI), vol. 2336, Springer, Heidelberg (2002)
10. He, Z., Xu, X., Deng, S.: Discovering cluster-based local outliers. *Pattern Recognition Letters* 24(9-10), 1641–1650 (2003)
11. Bay, D.S., Schwabacher, M.: Mining distance-based outliers in near linear time with randomization and a simple pruning rule. In: Proc. 2003 ACM SIGKDD Int. Conf. on Knowledge discovery and data mining (2003)
12. KDD 1999 (1999),
<http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html>

Fighting WebSpam: Detecting Spam on the Graph Via Content and Link Features*

Yu-Jiu Yang^{1,2}, Shuang-Hong Yang^{1,2}, and Bao-Gang Hu^{1,2}

¹ National Laboratory of Pattern Recognition

Institute of Automation, Chinese Academy of Sciences

² Beijing Graduate School, Chinese Academy of Sciences

P.O. Box 2728, Beijing, 100080 China

{yjyang, shyang, hubg}@nlpr.ia.ac.cn

Abstract. We address a novel semi-supervised learning strategy for Web Spam issue. The proposed approach explores graph construction which is the key of representing data semantical relationship, and emphasizes on label propagation from multi views under consistency criterion. Furthermore, we infer labels for the rest of the unlabeled nodes in fusing spectral space. Experiments on the Webspam Challenging dataset validate the efficiency and effectiveness of the proposed method.

1 Introduction

Detecting spam is one of the most important problems for improving the quality of search engine [3] [4]. Starting with the topology of the web, one can view the *Web Pages* as a connected directed large-scale graph on which Web Spam can be detected via the properties of the link-spam structure. On the other hand, the Web Pages can be represented as Vector Space Model(VSM) to capture their semantical information, such as Email spam filter in classical text mining community. In general, these techniques are independent each other. Moreover, how to fuse these dual prior information into a unified framework to reinforce detecting system is an interesting issue.

In this paper, we present a novel fusing strategy to boost the performance of the spam detection system via exploring the two aspects of Web Pages: that is, link features from hyperlink and semantical features from the k -way graph Laplacian based on content information. Our main contributions include: 1) Construct a **similarity graph** (G_{sim}) to measure the similarity relationship via combining **constraint graph** (G_c) and **nearest neighborhood graph** (G_{NN}). 2) Label the most confident nodes from multi-view graph under the consistency criterion for improving the robustness of detection system. 3) Construct a new spectral space to integrate content features and link information. The rest of this paper is organized as follows: Section [2] describes the related work. Section [3] depicts our proposed strategy for learning on the Web graph with partial labeled nodes.

* This work is partially supported by Natural Science Foundation of China under grant No. 60275025 and No. 60121302.

The detailed experimental results on ECML challenge dataset are presented in Section 4. We draw a conclusion for this paper in Section 5.

2 Related Work

In this section, we give a brief review on Web Spam studies. Methods for the detection of link-based spam explore link structures and spam link form on the graph, then re-rank score on the revised graph, such as propagating trust or distrust through links [4], or deleting links that look suspicious from the statistics characters. On the other hand, some statistics on link structures on web graph can be considered as the characters of the related pages in bag-of-word fashion. Hence, It is easy for us to transform a link spam detection problem into a typical machine learning issue.

For the purpose of going further to boost the performance of the learned classifier, Castillo et al. proposed a spam detection system that combines link-based and content-based features [3]. They apply stacked graphical model obtained by base classifier to implement some topology analysis.

Recently, Zhou et al. considered discrete analysis on directed graph for detecting web spam, and constructed a discrete analogue of classical regularization theory via discrete analysis with different transductive methods [6]. In the literature [5], Kamvar et al. presented a spectral learning algorithm with some constraints for clustering or classification. Their work demonstrated that the compacted spectral space offers a new powerful representation for data. In this paper, we will focus on spam detection on graph in semi-supervised fashion.

3 Our Algorithm

To describe conveniently, Web Spam detection problem is defined as follows:

Problem Statement 1. *Given a set of hosts(pages) $\{\mathbf{v}_i\}_1^n \in \mathcal{V}$, the corresponding link graph G_{link} from the hyperlinks and label set $\mathcal{V}_L = \{\mathbf{v}_i\}_1^l; Y_L \in \mathcal{Y}$. for each host(pages), we know its low-level content information in bag-of-words manner. The detection goal is to predict labels $Y_U \in \mathcal{Y} = \{1, -1\}$ for the rest hosts(pages) $\mathcal{V}_U \in \mathcal{V}$.*

Following Zhou et al. work [6], we model the data relation as two typical graphs, which capture content-based features and link-based features respectively. Differing from Zhou's work in which a markov mixture was constructed from multi views [6], we focus on the graph construction itself and then implement a co-training phase from multi views.

As aforementioned in the introduction, our method can be divided into three stages: 1) Construct two informative graphs, 2) Label some unlabeled data with a very high level of confidence and under the consistency criterion, 3) Analyze the rest of unlabeled data in the fusing spectral space.

3.1 Construct an Informative Similarity Graph

In our approach, the nodes in *Link-Graph* G_{link} and *Similarity-Graph* G_{sim} are the page objects. In general, A *Link-Graph* is available directly by crawling system and the edges in the *Link-Graph* are obtained from hyperlinks. A natural question raising here is how to create an appropriate graph to incorporate the content information in our detecting system. Let a tuple $G_{sim} = (V, E^S, W^S)$ denote a semantical *Similarity-Graph*, where $W^S = [w_{ij}]_{N \times N}$ is the weight matrix with the $(i; j)$ -th element w_{ij} indicating the strength of immediate connectivity between vertices v_i and v_j . For the purpose of data classification, the vertex set v_i coincides with the set of data points (labeled or unlabeled), and w_{ij} is a quantitative measure of the closeness of data points v_i and v_j .

Similar to Graph-based algorithm in semi-supervised learning, the edges in the Similarity-Graph measure the similarities between nodes. To capture the local and global semantical structure, the k -nearest neighbor graph is a common selection. Generally, there are two techniques to determinate the nearest neighbor: k -NN and ϵ -NN. For the simplification, we only consider k -NN graph in this paper. It should be mentioned that they are not symmetric on measure space, sometimes we force it to be a symmetric graph for computation convenience. Let $\mathcal{N}(v_i)$ be the set of the v_i nearest neighbors, then the edge $e_{ij}^{(N)}$ is defined by

$$e_{ij}^{(N)} = \begin{cases} 1 & \text{if } v_j \in \mathcal{N}(v_i) \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

On the other hand, noticing the importance of the label propagation in within-classes, we construct a with-class constraint graph $G_{(C)}$ to depict the label propagation in the same class. the edge $e_{ij}^{(C)}$ is defined by

$$e_{ij}^{(C)} = \begin{cases} 1 & \text{if } v_i \text{ and } v_j \text{ belong to the same class.} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

In contrast to G_{NN} , G_C is bidirectional and symmetric.

Now, A weight Similarity-Graph can be written as: $G_{sim} = \lambda G_{NN} \oplus (1-\lambda)G_C$, where the symbol \oplus depicts the Entry-wise sum, that is,

$$w_{ij} = \lambda w_{ij}^{(N)} + (1-\lambda)w_{ij}^{(C)} \quad \forall e_{ij}^{(N)} = e_{ij}^{(C)} = 1 \tag{3}$$

and λ is a trade-off factor. In practice, we choose the parameter λ using cross validation trick(it is set as 0.5 in this paper). A strength of this model lies in the fact that it incorporates labeled data to alleviate the noise effect, whereas the majority of graph deal strictly with the spatial relationship in unsupervised learning.

3.2 Label Propagation under the Consistency Criterion

The motivation for this phase is driven by the fact that the more labeled training data we have, the better performance is achieved. So, why not label some

unlabeled data with high confidence to improve the performance of the detection system? On the other hand, the idea of our algorithm is consistent with the common semi-supervised learning assumptions: 1) nearby points are likely to have the same label; 2) points on the same structure (such as a cluster or a sub-manifold) are likely to have the same label.

Inspired by the success of co-training method, we conduct belief propagation on the multi view graphs. If one node is labeled the same class label from the different graphs, we call it as *confidence node* and label its class label assuredly. To implement label propagation on the similarity graph, we apply spectral clustering on the G_{sim} in this phase. The benefit from spectral clustering is that it can capture multi-topical distribution if we choose an appropriate cluster number. However, it is harder to determinate the optimal cluster number in principle. For webspam detection task, we select a number bigger than 2 in our experiments. Then, we category unlabeled data according to clustering results and labeled data in the corresponding cluster.

For Link-Graph, we can conduct TrustRank algorithm [4] to score each node, and category all unlabeled nodes into spam and good hosts(pages). From the results of above procedure, we continue to find all confidence nodes in term of the consistency criterion. Let L^{con} denote the confident nodes set. We can rearrange a new labeled set $\hat{L} = L \cup L^{con}$ and a new unlabeled one $\hat{U} = U \setminus L^{con}$ by label propagation, then we get a new Similarity Graph \hat{G}_{sim} combining G_{NN} and new constrain graph \hat{G}_C . For the rest of unlabeled data, we infer their class label on the fusing spectral space which is conducted by the new Similarity-Graph \hat{G}_{sim} and Link Graph.

3.3 Spectral Space Analysis

So far, we have a Similarity Graph \hat{G}_{sim} , Link Graph G_{link} and the new labeled set \hat{L} to represent the data information. Now, we conduct a fusing framework to capture these useful prior information. Remember that the spectral vectors encode the fidelity of a cluster, we can utilize these graph matrices to build a compressing spectral space.

For Link Graph, we calculate the TrustRank scores and SpamRank Scores [2] using the new labeled set \hat{L} . Let A be an $n \times n$ adjacency matrix for a given web graph such that $A_{ji} = 1$ if page i links to page j and $A_{ji} = 0$ otherwise, a TrustRank [4] is defined as

$$\mathbf{P}_{trust} = \alpha \mathbf{T} \mathbf{P}_{trust} + (1 - \alpha) \mathbf{d} \quad (4)$$

where \mathbf{T} is a stochastic matrix which is related to the adjacency matrix \mathbf{A} , $\alpha \in [0, 1]$ is a given scalar and \mathbf{d} is a non-negative, L_1 normalized, personalized vector. The vector \mathbf{P}_{trust} can be computed by the power iteration and is the stationary distribution vector of Link Graph with a biased random walk.

Similar to the TrustRank algorithm, we can get spam score vector \mathbf{P}_{spam} using a different personalized vector and propagation direction according to in-link directionality. Actually, both \mathbf{P}_{trust} and \mathbf{P}_{spam} are the different attributions

in spectral space for the special nodes. Recall that new Similarity-Graph \hat{G} describes the semantical characters, we can resort to Laplacian operator to measure the semantical attribution in the spectral space.

Finally, the discussion above allow us to integrate these attributes of each node into a unitary vector. New low-dimensionality representation contains sufficient discriminative information for detecting spam. Consequently, we can detect spam in the new space using the classical classification algorithm or something else. In summary, we give the corresponding algorithm below:

Algorithm 1. Spectral Space Analysis

Semantical representation:

- 1: Given \hat{G} , form the Laplacian matrix $L \in \mathbb{R}^{n \times n} = D - W$ where D be the diagonal matrix with $D_{ii} = \sum_j W_{ij}$
- 2: Find x_1, \dots, x_k , the k smallest eigenvector of L and form the matrix $X = [x_1, \dots, x_k] \in \mathbb{R}^{n \times k}$
- 3: Normalize the rows of X to be unit length.

Link Analysis:

- 4: Given G_{link} and labeled set \hat{L} , form the corresponding Compute personalized vector v and compute the TrustRank vector r_{trust} .
- 5: Similar to step 4, reckon r_{spam} based on SpamRank algorithm.

Spectral space combination and classification:

- 6: Combine the k -way vectors, the TrustRank vector r_{trust} and the SpamRank vector r_{spam} as a new representation. So, construct the matrix $\hat{X} = [x_1, \dots, x_k, r_{trust}, r_{spam}]$.
 - 7: Classify unlabeled points in \mathbb{R}^k using any reasonable classifier.
 - 8: Assign the data point i the class C that X_i was assigned.
-

4 Experiments

To illustrate the effectiveness of our algorithm, we conduct the proposed algorithm on the corpus 1 from the ECML/PKDD Web spam challenge(see [1] for more details). In our experiments, firstly, we apply feature selection processing for the purpose of reducing problem size. By applying information gain scoring on the labeled data, 9862 features are selected from the original 4,924,007 features as a concise representation.

All parameters related Link-Graph are configured by the default value, such as the parameter α in TrustRank/SpamRank is chosen as 0.85. For the simplicity, we apply the standard KNN classification to infer the remained unlabeled data in the last phase of our method on the low-dimensionality spectral space.

Since there are about four times as many non-spam hosts as spam hosts in Webspam challenge data., spam detection is a highly unbalanced classification issue. In addition, as a cost-sensitive problem, classifying a normal host into spam is much worse than classifying a spam host into normal. Hence, we need to measure algorithmic performances via precision, recall and classification accuracy, rather than a single evaluation index.

Table 1. Summary of Classifier Performance for the Corpus 1. The table shows the effectiveness of the proposed algorithm.

Methods	Features	Evaluated Data Sets	Precision	Recall	Accuracy
SVM (baseline)	Content	Validation	1.00	.031	.791
		Test	.936	.042	.797
Spam/TrustRank	Link	Validation	.848	.200	.820
		Test	.827	.212	.824
Transductive learning	Con/Link	Validation	.793	.409	.845
		Test	.774	.421	.837
Our approach	Con/Link	Validation	.801	.412	.915
		Test	.766	.462	.868

For a comparison purpose, we firstly use a standard SVM classifier, based on content features, as one baseline classifier in supervised learning manner; we also report the result of the fusing SpamRank and TrustRank strategy for Webspam detection, this method is based on Link-Graph; as a comparison of the performance, a Transductive learning algorithm on the multi-view graph is applied the same dataset. Experimental results is reported in Table 1.

From Table 1, we observe that the baseline classifier obtains the highest precision, but it is difficult to accept that the recall is less than 10 percent. It also means that content features are useful for detecting Webspam. Comparably, the semi-supervised learning outperforms the supervised learning according to accuracy index and the classifier based combining features perform well. Our proposed algorithm utilizes both link and content features and improves the overall performance of the detection system.

5 Conclusion

The proposed method integrates unlabeled data and labeled data into a unified learning framework. Empirical studies show that it is competitive with the start-of-the-art detecting system in terms of some standard evaluation indexes. For future work, we will extend the proposed algorithm to the out-of-sample case and explore an efficiency approximation algorithm to alleviate computation complexity.

References

1. Baeza-Yates, R., Castitlo, C., Davison, B.D., Denoyer, L., Gallinari, P.: Web spam challenge (2007), <http://webspam.lip6.fr/wiki/pmwiki.php>
2. Benczúr, A.A., Csalogány, K., Sarlós, T., Uher, M.: Spamrank – fully automatic link spam detection. In: AIRWeb, pp. 25–38 (2005)

3. Castillo, C., Donato, D., Gionis, A., Murdock, V., Silvestri, F.: Know your neighbors: Web spam detection using the web topology. In: Proceedings of SIGIR, Amsterdam, Netherlands, pp. 423–430. ACM Press, New York (2007)
4. Gyöngyi, Z., Garcia-Molina, H., Pedersen, J.: Combating web spam with trustrank. In: VLDB, pp. 576–587 (2004)
5. Kamvar, S.D., Klein, D., Manning, C.D.: Spectral learning. In: IJCAI, pp. 561–566 (2003)
6. Zhou, D.Y., Burges, C.J.C.: Spectral clustering and transductive learning with multiple views. In: ICML 2007, pp. 1159–1166 (2007)

A Framework for Discovering Spatio-temporal Cohesive Networks

Jin Soung Yoo¹ and Joengmin Hwang²

¹ Department of Computer Science, Indiana University-Purdue University,
Fort Wayne, Indiana, USA
yooj@ipfw.edu

² Computer Science & Engineering Department, University of Minnesota,
Minneapolis, Minnesota, USA
jhwang@cs.umn.edu

Abstract. A *spatio-temporal cohesive network* represents a social network in which people often interact closely in both space and time. Spatially and temporally close people tend to share information and show homogeneous behavior. We discuss modeling social networks from spatio-temporal human activity data, and alternative interest measures for estimating the strength of subgroup cohesion in spatial and temporal space. We present an algorithm for mining spatio-temporal cohesive networks.

1 Introduction

The recent revolution in mobile aware technology, e.g., GPS, mobile phones, and in-car navigation systems, has allowed rich data to be collected about the activities of individuals. The daily activity data can form complex and dynamic networks of spatial and spatio-temporal interactions of people. Spatial social science [1] recognizes the key role that spatial concepts, such as location, distance, proximity, neighborhood, and region, play in human society. Spatially and temporally close social groups tend to share information and have homogeneous behavior in space and time. The identification of the interesting patterns can provide important insights into many application domains such as homeland defense, public health, ecology, business and education.

Social networks, in the most general sense, refer to relationships that shape a society's social interaction [1]. Sociologists have long studied spatially complete social networks based on local interaction [5,2,13]. However, there are very few theoretical works that study space and time simultaneously. In the data mining literature, there is increasing interest in mining social networks. Most works have concentrated on identifying social networks based on non spatial context. In other hand, [4] views the movements of people among specific locations as a spatial interaction problem. [7] worked on mining social networks using spatio-temporal events. However, the event is defined to a semantic event which is any social collectivity of actors, e.g., conferences and games, not having actual geographic location.

One of the key problems in mining spatio-temporal social networks is how to appropriately model human activity data to find social networks. We discuss

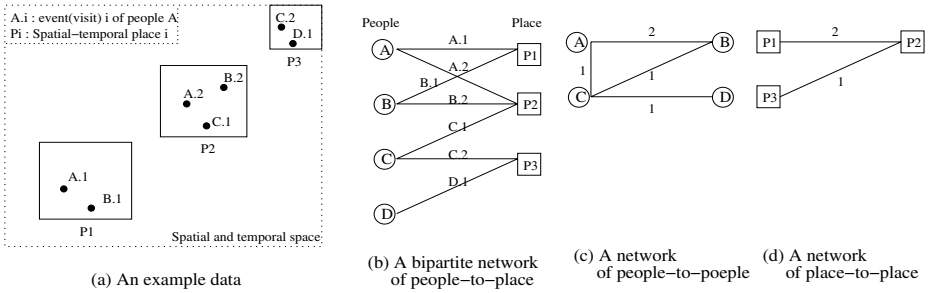


Fig. 1. Spatio-Temporal Semantic Model

the two approaches, *spatio-temporal semantic model* and *spatio-temporal location model*. In this paper, we focus on the problem to discover *spatio-temporal cohesive networks* based on the spatio-temporal location model. One important cohesive network is a *clique* [8]. A clique refers to a network in which there is a relationship between any two people. Our spatio-temporal cohesive network could be interpreted as one in which has a relationship with everyone in his/her spatial neighborhood within a time interval. Another important problem for discovering the patterns is to use proper interest measures for estimating the strength of subgroup cohesion in spatial and temporal space. We present alternative interest measures and compare them. Spatio-temporal cohesive network mining also presents computational challenges since the activity objects are embedded in continuous space and time. We extend our algorithm proposed for mining co-located itemsets in space [12] to discover spatio-temporal cohesive networks from spatio-temporal datasets.

The remainder of the paper is organized as follows. Section 2 discusses social network models of spatio-temporal human activity data. Section 3 defines the problem of mining spatio-temporal cohesive networks and presents the algorithm. Section 4 is concluded with future work.

2 Social Network Modeling

We first discuss two different approaches for modeling social networks from spatio-temporal human activity data. Then we define spatio-temporal cohesive networks and present the interest measures.

2.1 Spatio-temporal Semantic Model

The spatio-temporal semantic model considers the visit activity of people to specific places and time (intervals). Figure 1 (a) shows an example of visited places {P1, P2, P3} of four persons {A, B, C, D}. A place means a semantic location which is visited by people. For instance, a conference is an example of a semantic place in which researchers gather and exchange their work and thought. A name of a locality and a ZIP code are another example of semantic

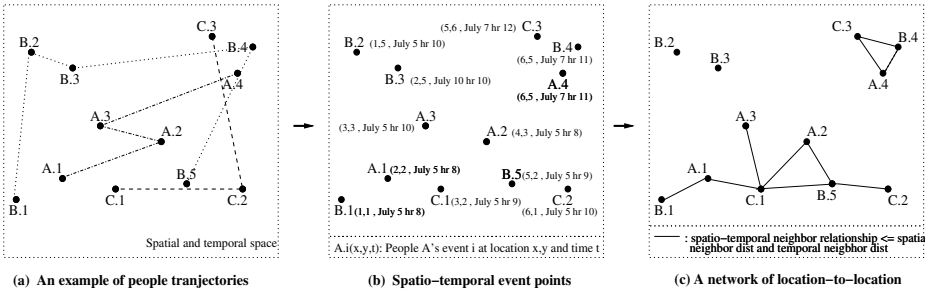


Fig. 2. Spatio-Temporal Location Model

location. The places can be different geographic locations or can be the same location with different time intervals of visit. In Figure 2 (a), each visit activity is represented by the visitor and an unique activity id per person, e.g., A.1.

People-to-Place. An individual and a place have a relationship if the person visits the place in a time interval. Bipartite graphs are often used to model people and place relations [3]. In a bipartite graph, vertices are divided into two disjoint sets, e.g., a set of people and a set of places, as Figure 2 (b). An edge may be labeled by an activity event to link its people and place. A bipartite network can be transformed into a one-mode social network which depends on the emphasis of a specific type of interaction, i.e. people to people or place to place.

People-to-People. In a people-to-people model, the people are linked in a social network based on their common visits to places. The number of common visits can be used as the weight of each link. For example, in Figure 2 (c), people A and B are linked with a weight of 2 due to their visits to places P1 and P2.

Place-to-Place. In a place-to-place network, two places are connected if they share at least one visitor. Figure 2 (d) shows the weight of connection between two places which is defined by the number of visitors.

2.2 Spatio-temporal Location Model

The spatio-temporal semantic models are limited to interactions based on specific semantic places rather than the contact of people in arbitrary geographic location. In contrast, the spatio-temporal location model captures the interaction of people in the geographic context with time. For example, suppose Figure 2 (a) shows a simplified example of people moving in space and time. An activity event can be defined with a geographic location where people stay for a while. For example, an activity event can be modeled as a tuple \langle person, event, location (x, y) , time (t) , other attributes \rangle , where person is an individual who engages in an event and event is a distinguished event per person, x and y represent a geographic location where the event happens, and time is the start time (optionally including duration) of the event. The activity event can also include non spatial and temporal attributes, e.g., activity type. Figure 2 (b) shows the spatio-temporal event location points from Figure 2 (a).

Location-to-Location. The activity events in spatial and temporal space can be connected by a spatial relationship, e.g., metric relationship (e.g., Euclidean distance), topology relationship (e.g., within, nearest), and a temporal relationship, e.g., before, overlap, contain. Figure 2 (c) shows a network of location-to-location in spatial and temporal space. Two events (e.g., A.1 and B.1) are connected because they occur close to one another within a spatial distance and a temporal distance.

2.3 Spatio-temporal Cohesive Network

We model our spatio-temporal cohesive social network based on the spatio-temporal location model. Despite the variability in semantics, social networks share a common structure in which social entities are generally termed *actors* and the relationships between a pair of social entities are known as *ties* [9]. Let us suppose that A is a set of actor and E is a set of spatio-temporal events of the actors. We define a spatio-temporal cohesive network as follows.

Definition 1. A spatio-temporal cohesive network N is a subset of actors, $N \subseteq A$, whose events $I \subseteq E$ often form cliques using a spatial neighbor relationship and a temporal neighbor relationship.

We introduce two different interest measures for measuring cohesion of networks. First, cohesive index represents a probability that makes cliques from all possible events of the actors.

Definition 2. The cohesive index $CI(N)$ of a network $N = \{a_1, \dots, a_k\}$ is defined as $\frac{|cohesive\ instances\ of\ N|}{\prod_{i=1}^k |events\ of\ a_i|}$, where a_i is an actor of N and Π is the multiplication function.

In the definition, the cohesive instance of a network N is a subset of events, $I \subseteq E$, that includes the events of all actors in the network N and forms a clique using a spatial relationship and a temporal relationship.

Definition 3. The participation index $PI(N)$ of a network $N = \{a_1, \dots, a_k\}$ is defined as $\min_{a_i \in N} \left\{ \frac{|\pi_{a_i}(cohesive\ instances\ of\ N)|}{|events\ of\ a_i|} \right\}$, where π_{a_i} is selection of distinct events of actor a_i .

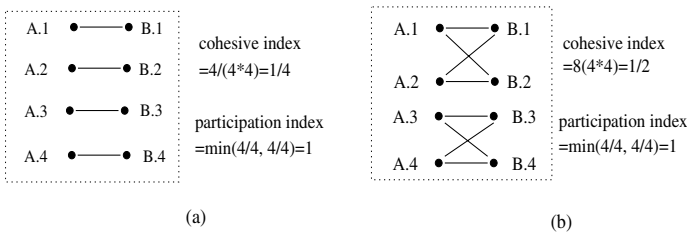


Fig. 3. Comparison of Interest Measures

The participation index was first introduced for spatial co-location mining in [10]. We adopt the measure for measuring the cohesion of objects in space and time. The participation index considers the ratio of actor events who participate in a cohesive network. A high cohesive index or participation index indicates that the actors in the social network likely have ties together through their activities. However, the two interest measures have different characteristics. Figure 3 shows a comparison of them. In the case of Figure 3 (a), each event of actor A (or B) has a neighbor relationship with the corresponding event of actor B(or A). The participation index seems to perfectly capture the co-occurrence relationship with the value 1. In contrast, the cohesive index shows a low strength value since only four pairs among all possible 16 pairs have neighbor relationships. Next, let us consider the case of Figure 3 (b). The cohesive index reflects the strength of neighbor relationships with the increase of neighbor pairs. In contrast, the participation index does not reflect the strength well. In both datasets, the participation index shows the same prevalence values. The choice of interest measure leaves to the application.

3 Spatio-temporal Cohesive Network Mining

We define the problem of mining spatio-temporal cohesive networks as follows. Given a set of actor $A = \{a_1, \dots, a_n\}$, a set of spatio-temporal events $E = E_1 \cup \dots \cup E_n$, where $E_i (1 \leq i \leq n)$ is a set of events of an actor a_i , a spatial distance neighbor relationship, a temporal distance neighbor relationship, and a minimum prevalent threshold, we want to find spatio-temporal cohesive networks whose prevalence values (CI or PI) are greater than the prevalence threshold.

Spatio-temporal cohesive network mining presents computational challenges since the event objects are embedded in continuous space and time. It is hard to transactionize a spatial-temporal dataset to apply traditional data mining techniques. Another way is to find all spatio-temporal neighboring object pairs from the input dataset, represent them to a neighborhood graph, and then find all cohesive instances from the graph for calculating the prevalence values. However, finding all cliques from the graph is NP-complete in the graph-theory. It is also non-trivial to reuse subgraph mining [6]. The subgraph mining was used to find frequent subgraphs in a large graph database (i.e., a set of graphs). We extend our algorithm proposed for mining co-located feature sets in space [12] to discover spatio-temporal cohesive networks from spatio-temporal datasets. The following shows the pseudo code of the algorithm.

4 Conclusion

This paper presents a framework for mining cohesive networks to study the interaction of people in space and time. We need detail optimization of the proposed algorithm according to spatial and temporal characteristics of data, and the evaluation of the scalability with large real datasets. In the future, we also plan to explore more relaxed structures as well as clique as spatio-temporal cohesive subgroups, and develop the mining algorithm.

Inputs

A : a set of actors, E : a spatio-temporal event dataset
 SR : a spatial neighbor relationship, TR : a temporal neighbor relationship, min_prev : minimum prevalence threshold

Output

Spatio-temporal cohesive networks whose prevalence $\geq min_prev$

Variables

NR : a neighborhood relation table, C_k : a set of size k candidate networks, CI_k : candidate cohesive instances of size k networks, TI_k : true cohesive instances of size k networks, P_k : a set of size k prevalent spatio-temporal cohesive networks

Method

```

1)  $NR = \text{gen\_star\_neighborhood\_relation\_table}(E, SR, TR)$ ;
2)  $P_1 = A$ ;  $k = 2$ ;
3) while (not empty  $P_{k-1}$ ) do
4)    $C_k = \text{gen\_candidate\_networks}(P_{k-1})$ ;
5)   for  $r \in NR$  do
6)      $CI_k = \text{filter\_star\_instances}(C_k, r)$ ;
7)   end do
8)   if  $k = 2$  then  $TI_k = CI_k$ 
9)   else do  $C_k = \text{filter\_coarse\_prev\_networks}(C_k, CI_k, min\_prev)$ ;
10)       $TI_k = \text{filter\_clique\_instances}(C_k, CI_k)$ ;
11)   end do
12)   $P_k = \text{find\_prev\_cohesive\_networks}(C_k, TI_k, min\_prev)$ ;
13)   $k = k + 1$ ;
14) end do
15) return  $\bigcup(P_2, \dots, P_k)$ ;

```

References

- Center for spatially integrated social science. Center for Spatially Integrated Social Science, <http://www.csiss.org/>
- Coleman, J.S.: Foundations of Social Theory. Harvard University Press (1990)
- Eubank, S., Guclu, H., Anil Kumar, A., Marathe, M., Srinivasan, A., Toroczkai, Z., Wang, N.: Modeling Disease Outbreaks in Realistic Urban Social Networks. Nature 429, 180–184 (2004)
- Guo, D.: Mining and Visualizing Spatial Interaction patterns for Pandemic Response. In: Workshop on Spatial Data Mining, SIAM Conf. on Data Mining (2006)
- Johnson, C., Gilles, R.P.: Spatial Social Networks. Review of Economic Design 5, 273–300 (2000)
- Kuramochi, M., Karypis, G.: Frequent Subgraph Discovery. In: Proc. of IEEE Intl. Conf. on Data Mining (2001)
- Lauw, H.W., Lim, E.P., Tan, T.T., Pang, H.H.: Mining Social Network from Spatio-Temporal Events. In: Workshop on Link Analysis, Counterterrorism and Security (2005)
- Luce, R., Perry, A.: A Method of Matrix Analysis of Group Structure. Psychometrika 14, 95–116 (1949)
- Scott, J.: Social Network Analysis: A Handbook, 2nd edn. Sage, Thousand Oaks (2000)
- Shekhar, S., Huang, Y.: Co-location Rules Mining: A Summary of Results. In: Proc. of Intl. Sym. on Spatio and Temporal Database (2001)
- Wasserman, S., Faust, K.: Social Network Analysis. Cambridge University Press, Cambridge (1994)
- Yoo, J.S., Shekhar, S., Celik, M.: A Join-less Approach for Spatial Co-location Mining: A Summary of Results. In: Proc. of Fifth IEEE Intl. Conf. on Data Mining (2005)
- Goodchid, M.F., Janelle, D.G.: Spatially Integrated Social Science. Oxford University Press, Oxford (2004)

Efficient Mining of Minimal Distinguishing Subgraph Patterns from Graph Databases^{*}

Zhiping Zeng, Jianyong Wang, and Lizhu Zhou

Tsinghua University, Beijing, 100084, P.R. China

clipse.zeng@gmail.com, {jianyong, dcszlj}@tsinghua.edu.cn

Abstract. Distinguishing patterns represent strong distinguishing knowledge and are very useful for constructing powerful, accurate and robust classifiers. The *distinguishing graph patterns (DGPs)* are able to capture structure differences between any two categories of graph datasets. Whereas, few previous studies worked on the discovery of DGPs. In this paper, as the first, we study the problem of mining the complete set of *minimal DGPs* with any number of positive graphs, arbitrary positive support and negative support. We proposed a novel algorithm, MDGP-Mine, to discover the complete set of minimal DGPs. The empirical results show that MDGP-Mine is efficient and scalable.

Keywords: Graph Mining, Distinguishing Pattern, Distinguishing Subgraph.

1 Introduction and Formulation

1.1 Motivation

Distinguishing patterns are those whose frequencies change significantly from one dataset to another. They are able to capture regions of high differences between two classes of data and emerging trends in business or demographic data, and can be used to construct accurate and robust classifiers. Like other patterns or rules composed of conjunctive combinations of elements, distinguishing patterns can be easily understood and used directly by people. Emerging pattern is such a kind of distinguishing pattern. Emerging patterns show strong distinguishing knowledge and have been shown to be very successful for constructing accurate and robust classifiers[3], as well as providing intuitive description of sharp differences between classes of data. They have also been used in bioinformatics applications, like predicting the likelihood of diseases such as acute lymphoblastic leukemia and discovering knowledge in gene expression data[2]. The authors of [1] introduced the concept of a distinguishing subsequence, which is a subsequence that appears frequently in one class of sequences, yet infrequently in another. Distinguishing subsequences can be applied to the comparison of proteins, design of microarrays, characterization of text and the building of classification models.

^{*} This work was partly supported by National Natural Science Foundation of China under Grant No. 60573061, National Basic Research Program of China under Grant No. 2006CB303103, Basic Research Foundation of Tsinghua National Laboratory for Information Science and Technology (TNList), and Program for New Century Excellent Talents in University, State Education Ministry of China under Grant No. NCET-07-0491.

Furthermore, since graphs can represent more complicated relationships among different objects, distinguishing graphs have raised great interest and played a significant role in the mining of distinguishing patterns. There are many situations where DGPs can be applied, such as comparing structural differences between chemical compounds.

1.2 Preliminary Concepts

To simplify our discussion, in the following we introduce some preliminary concepts and notations. The problem of mining minimal DGPs is also formulated.

In this paper, we consider only simple graphs, i.e., undirected graphs without multi-edges and self-loops. An **undirected labeled graph** G can be represented by a 6-tuple, $G=(V, E, L_v, L_e, F_v, F_e)$, where $V=\{v_1, v_2, \dots, v_k\}$ is the set of vertices, $E\subseteq V\times V$ is the set of edges in G , L_v and L_e are the sets of vertex labels and edge labels respectively, $F_v:V\rightarrow L_v$ and $F_e:E\rightarrow L_e$ are mapping functions assigning the labels to the vertices and edges respectively. A graph G_1 is **graph isomorphic** to another graph G_2 iff there exists a bijection $f:V_1\rightarrow V_2$ such that for any vertex $v\in V_1$, $f(v)\in V_2\wedge F_v(v)=F_v(f(v))$, and for any edge $(u, v)\in E_1$, $(f(u), f(v))\in E_2\wedge F_e(u, v)=F_e(f(u), f(v))$. G_1 is a **sub-graph** of another graph G_2 iff $V_1\subseteq V_2$ and $E_1\subseteq E_2\cap(V_1\times V_1)$. Equivalently, G_2 is the **supergraph** of G_1 . If G_1 is graph isomorphic to a subgraph g of G_2 , we say G_1 is **subgraph isomorphic** to G_2 and g is an **instance** of G_1 in G_2 .

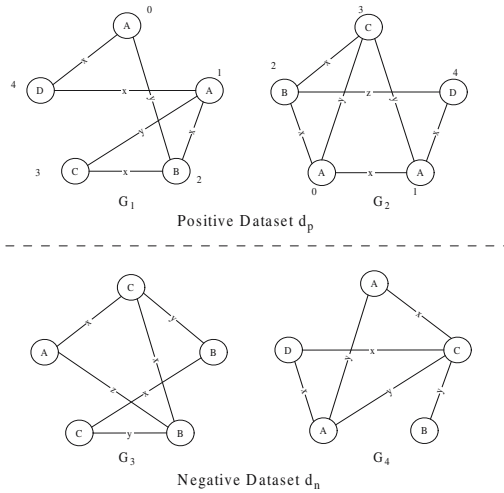


Fig. 1. A running example of graph database d_0 containing d_p and d_n

An input graph database D is a set of input graphs, the number of graphs in D is denoted by $|D|$. Given a database D , the number of graphs that contain at least one instance of g is called the **support** of g w.r.t. D , denoted by $sup(g, D)$. Assume we have a database D defined upon a set of input graphs and a partition of D into two sets, the positive class of input graphs(denoted by D_p) and the negative class of input graphs(denoted by D_n). The supports of g w.r.t. D_p and D_n (i.e., $sup(g, D_p)$ and

$sup(g, D_n)$) are named as **positive support** and **negative support**, respectively. While the context is clear, we will omit the dataset names and use $psup(g)$ and $nsup(g)$ instead of $sup(g, D_p)$ and $sup(g, D_n)$, respectively.

Definition 1. (Distinguishing Subgraph Pattern) *Given a database which consists of a positive dataset D_p and a negative dataset D_n , the positive support threshold α and the negative support threshold β ($\alpha, \beta \in [0, 1], \alpha \gg \beta$), a **distinguishing subgraph pattern** (abbreviated as **DGP**) is a subgraph g satisfying the following two constraints: (1) $psup(g)/|D_p| \geq \alpha$, (2) $nsup(g)/|D_n| \leq \beta$. Furthermore, g is a **minimal DGP** if no proper-subgraph of g is a DGP.*

In the following, by default we assume the input graph database D consists of two classes of graphs, one is positive and the other is negative. The set of positive graphs is denoted by D_p , while the set of negative graphs is denoted by D_n .

Problem Statement: *Given an input graph database D which is composed of a positive dataset D_p and a negative dataset D_n , a positive support threshold α and a negative support threshold β ($\alpha, \beta \in [0, 1], \alpha \gg \beta$), we study the problem of mining the complete set of minimal DGPs from the database D .*

2 Related Work

Distinguishing pattern mining has been extensively studied in recent years, such as emerging pattern and emerging rule mining, distinguishing subsequence discovery [1], contrast graph mining [4] and so on. However, no existing work can be directly used to enumerate interesting differences between classes of graphs. Contrast graphs introduced in [4] is defined as the graph structure appears in one positive input graph but never appears in the negative input graphs, which is a special case of DGP on the condition that $|D_p|=1$, $\alpha=1$ and $\beta=0$. Thus, the algorithm they proposed in [4] cannot be applied widely and popularly. DGPs proposed in this paper, however, does not have these restrictions. Therefore, algorithms for mining minimal DGPs can be applied more widely and popularly. Whereas, no previous work were done on this issue and it is urgent to devise new algorithms for mining minimal DGPs.

To our best knowledge, in this paper we propose the first algorithm for mining minimal DGPs from two classes of graph datasets. According to the definition of DGPs, we can get a rudimentary solution for mining DGPs. Firstly, we can discover the complete set of frequent subgraphs in D_p , denoted by S_p . Secondly, we can mine the complete set of frequent subgraphs in D_n , denoted by S_n . Finally, the set $S_p - S_n$ is the result of DGPs and the minimal DGPs can be generated from it. Apparently, this approach is rather brute-force and will take an unacceptable time. To improve the efficiency of the above method, we use an enumeration strategy to inspect the frequent subgraphs w.r.t. the support threshold α in D_p . Once inspect a frequent subgraph g in D_p , we check the support of this subgraph in D_n . If $sup(g, D_n)/|D_n| \leq \beta$, we can say that g is a DGP. Together with other techniques, we can determine whether g is a minimal DGP or not.

3 The MDGP-Mine Algorithm

3.1 Enumeration of Positive Frequent Subgraphs

Many previous work were done on mining frequent graph patterns, typical examples include gSpan[6], CLAN[5], Cocain*[7] and so on. As shown in previous research, to discover and enumerate the frequent graph patterns in the graph dataset, we need to handle two basic problems discussed in this section.

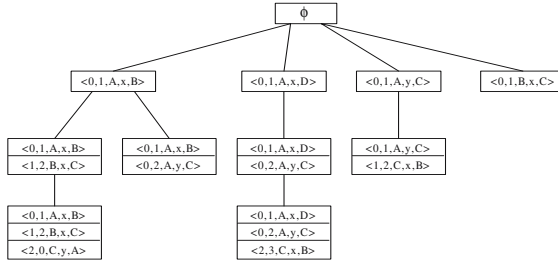


Fig. 2. The DFS Code Tree for the frequent subgraphs in d_p in Figure 1 with support 100%

The first problem is to find a canonical representation of graphs s.t. if two graphs have identical canonical representations, they are isomorphic. The **DFS Code** introduced in [6] is a popular representation of graphs which is widely used in recent years. Minimum DFS code has a nice property: two graphs g and g' are graph isomorphic iff $min(g)=min(g')$ ($min(g)$ represent the minimum DFS code of the graph g). Moreover, with the help of minimum DFS codes, the problem of mining frequent subgraphs is reduced to mining frequent minimum DFS codes which are sequences. After finding a canonical representation for graphs, we need an enumeration strategy to guarantee that we can discover the complete set of frequent subgraphs. Based on the minimum DFS code, **DFS Code Tree** enumeration strategy was proposed in [6], whose advantage and efficiency are verified by the experimental results on both real and synthetic datasets. Therefore, this efficient enumeration strategy is also adopted in this work. Figure 2 shows the DFS Code Tree for the frequent subgraph patterns with support 100% in d_p in our running example shown in Figure 1.

3.2 Discovery of Minimal DGPs

Lemma 1. (Early Pruning) *If g is a DGP w.r.t. the positive and negative datasets, then all descendants of this pattern in the DFS Code Tree will not be minimal DGPs.*

Proof. Since g is a DGP, $\frac{sup(g,D_p)}{|D_p|} \geq \alpha$ and $\frac{sup(g,D_n)}{|D_n|} \leq \beta$. Assume g' is a descendant of g in the DFS Code Tree, g' must be a proper-supergraph of g , according to the property of Frequency Antimonotone [6], $\frac{sup(g',D_n)}{|D_n|} \leq \beta$. No matter g' is frequent or infrequent in d_p , g' must be not a minimal DGP. This is because even g' is frequent and is a DGP pattern, it will not be a minimal DGP as one of its proper-subgraph, g , is a DGP.

According to Lemma 1, we can prune unpromising parts of search space. However, we still cannot determine whether a DGP is minimal or not. Even if the current subgraph is a DGP, but we cannot state it is minimal. Therefore, while a DGP is discovered, two operations should be performed. The first one(**CHK1**) checks whether the current pattern is a proper-supergraph of other already discovered DGPs. If so, just discard it. The second(**CHK2**) checks whether the current pattern is a proper-subgraph of already discovered DGPs. If so, remove these patterns and insert this DGP to the result set.

Fortunately, if a DGP p is a proper-supergraph of another DGP p' which is discovered before, p will not be enumerated. If p is a descendant of p' in the DFS Code Tree, p must be pruned according to Lemma 1 otherwise, if p is generated from other paths, the DFS Code of p must be not minimal and will be pruned by the $s \neq \min(s)$. Consequently, we do not need to do the operation CHK1.

3.3 Algorithms

In this section, we will describe the algorithms used in our solution by integrating various techniques discussed earlier.

ALGORITHM 1: MDGP-Mine(D_p, D_f, α, β)

INPUT: (1) D_p – the input positive graph dataset; (2) D_n – the input negative graph dataset;
 (3) α – the positive support threshold; (4) β – the negative support threshold.
 OUTPUT: rs – the set of DGPs.
 BEGIN
 1. Scan the positive graph dataset D_p to get the frequent edge set E_f ;
 2. Remove the edges from D_p and D_n which is not in E_f ;
 3. Remove the edges from E_f, D_p and D_n which satisfy $sup(e, D_p) \leq \beta$ and insert these edges to rs ;
 4. Sort the edges in E_f ;
 5. For each edge e in E_f
 6. **DGP-Enum**(e);
 END

MDGP-Mine Algorithm. At first, we introduce ALGORITHM 1 **MDGP-Mine** which can discover the complete set of minimal DGPs. We first scan D_p to get frequent edge set E_f of D_p (line 1). After then, we remove the edges which are not in E_f from both D_p and D_n (line 2). Thirdly, we find the edges in D_n which satisfy the negative support constraint. Since these single edges are DGPs and all supergraphs which contain one of them will be not minimal. Therefore, we insert them to the result set directly and remove them from the datasets and E_f (line 3). After then, we sort the edges remaining in E_f , and call **DGP-Enum** for each edge to discover minimal DGPs(lines 4-5).

SUBALGORITHM 2: DGP-Enum(d)

INPUT: d – a DFS code representing a subgraph pattern.
 OUTPUT: rs – the set of valid extensible candidates w.r.t. g .
 BEGIN
 07. If $d \neq \min(d)$
 08. return;
 09. if $sup(d, D_p) \geq \alpha$ and $sup(d, D_n) \leq \beta$
 10. Insert d to rs according to the minimal checking scheme;
 11. return;
 12. Get the extensible DFS edge set E for g ;
 13. Sort the edges in E ;
 14. For each edge $e \in E$
 15. **DGP-Enum**($d \diamond e$);
 END

DGP-Enum Algorithm. Whereafter, SUBALGORITHM 2 **DGP-Enum** for inspecting the frequent subgraphs in D_p and mining minimal DGPs will be introduced. First, we check whether current DFS code d is minimal or not, if not we can stop growing d (lines 07-08). Second, we determine whether d is a DGP, if so we insert d to the result according to the minimal checking scheme and stop growing d in term of Lemma 1 (lines 09-11). Meanwhile, if d is not a DGP, we get the extensible DFS edge set E for d (line 12) and sort them(line 13). For each DFS edge e in E , we add e into d to get a new DFS code and enumerate it(lines 14-15).

4 Experiments and Conclusion

We conducted comprehensive experiments to evaluate MDGP-Mine. All experiments were performed on a PC running FC 4 Linux and with 1.8GHz AMD CPU and 1GB memory. Datasets are generated by a synthetic generator. The parameters accepted by the generator are the same as described in [7]. In Figure 3(a) we fixed the size of positive input graphs $|D_p|$ at $40k$ and varied the size of negative input graphs $|D_n|$ from $50k$ to $100k$. While in Figure 3(b) we fixed $|D_n| = 50k$ and varied $|D_p|$ from $50k$ to $100k$. The results in Figure 3 show that the runtime of MDGP-Mine has a linear relationship with both $|D_p|$ and $|D_n|$. Therefore, the algorithm MDGP-Mine is scalable.

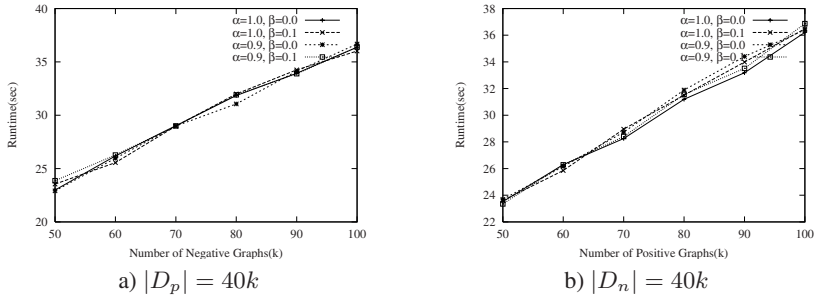


Fig. 3. Efficiency and Scalability

In this paper, we firstly studied the problem of mining the complete set of minimal DGPs and proposed a novel algorithm MDGP-Mine. Comprehensive experiments show that MDGP-Mine is efficient and scalable.

References

1. Ji, X., Bailey, J., Dong, G.: Mining minimal distinguishing subsequence patterns with gap constraints. In: ICDM 2005, Houston, Texas, USA, pp. 194–201 (2005)
2. Li, J., Liu, H., Ng, S.-K., Wong, L.: Discovery of significant rules for classifying cancer diagnosis data. In: ECCB 2003, pp. 93–102 (2003)
3. Ramamohanarao, K., Bailey, J., Fan, H.: Efficient mining of contrast patterns and their application to classification. In: ICISIP 2006, Bangalore, India (2006)

4. Ting, R.M.H., Bailey, J.: Mining minimal contrast subgraph patterns. In: Jonker, W., Petković, M. (eds.) *SDM 2006*. LNCS, vol. 4165, Springer, Heidelberg (2006)
5. Wang, J., Zeng, Z., Zhou, L.: Clan:an algorithm for mining closed cliques from large dense graph databases. In: *ICDE 2006*, April 2006, pp. 73–82 (2006)
6. Yan, X., Han, J.: gspan: Graph-based substructure pattern mining. In: *ICDM 2002*, Washington, DC, USA, pp. 721–724 (2002)
7. Zeng, Z., Wang, J., Zhou, L., Karypis, G.: Out-of-core coherent closed quasi-clique mining from large dense graph databases. *ACM Transactions on Database Systems* 32(2) (2007)

Combined Association Rule Mining

Huafeng Zhang, Yanchang Zhao, Longbing Cao, and Chengqi Zhang

Faculty of IT, University of Technology, Sydney, Australia
P.O. Box 123, Broadway, 2007, NSW, Australia
{hfzhang, yczhao, lbcao, chengqi}@it.uts.edu.au

Abstract. This paper proposes an algorithm to discover novel association rules, combined association rules. Compared with conventional association rule, this combined association rule allows users to perform actions directly. Combined association rules are always organized as rule sets, each of which is composed of a number of single combined association rules. These single rules consist of non-actionable attributes, actionable attributes, and class attribute, with the rules in one set sharing the same non-actionable attributes. Thus, for a group of objects having the same non-actionable attributes, the actions corresponding to a preferred class can be performed directly. However, standard association rule mining algorithms encounter many difficulties when applied to combined association rule mining, and hence new algorithms have to be developed for combined association rule mining. In this paper, we will focus on rule generation and interestingness measures in combined association rule mining. In rule generation, the frequent itemsets are discovered among itemset groups to improve efficiency. New interestingness measures are defined to discover more actionable knowledge. In the case study, the proposed algorithm is applied into the field of social security. The combined association rule provides much greater actionable knowledge to business owners and users.

1 Introduction

Association rule mining aims to discover relationships among data in huge database. These relationships may provide some clues for business users to perform actions. In recent years, researchers [6,9] have focused on discovering more actionable knowledge. However, conventional association rules can only provide limited knowledge for potential actions. For example, in the Customer Relationship Management (CRM) field, association rule mining can be used to prevent churning. One possible rule is “ $Demo : D \Rightarrow Churning$ ”. With this rule, business users may take some actions on the customers with “ $Demo : D$ ” to prevent churning. However, from the mined rule, business users cannot get knowledge on what action should be taken to retain these customers, though there might be many candidate actions.

We have previously defined combined association rule [11] to mine actionable knowledge. However, in [11], all of the attributes are treated equally when finding the frequent itemsets. The algorithm is time-consuming when a large number of

attributes are in database. In this paper, we differentiate the attributes and find the frequent itemsets on groups of itemsets. Furthermore, since data imbalance is often encountered in data mining tasks, we will also tackle data imbalance problem in combined association rule mining.

The paper is organized as follows. Section 2 gives the definition of combined association rule and its characteristics. Section 3 proposes the interestingness measures and algorithm outline. Section 4 introduces a case study. Section 5 presents some related work. Section 6 is the summary of this paper.

2 Definition of Combined Association Rule

Let T be a dataset. In this dataset, each tuple is described by a schema $S = (S_{D1}, \dots, S_{Dm}, S_{A1}, \dots, S_{An}, S_C)$, in which $S_D = (S_{D1}, S_{D2}, \dots, S_{Dm})$ are m non-actionable attributes, $S_A = (S_{A1}, S_{A2}, \dots, S_{An})$ are n actionable attributes, and S_C is a class attribute. Note that the data for combined association rule is not limited to one dataset. In fact, different kinds of attributes are often from multiple datasets [11].

Combined association rule mining is to discover the association among the ‘attribute-value’ pairs. For the convenience of description, we call an ‘attribute-value’ pair an ‘item’. Suppose itemset $D \subseteq I_D$, I_D is the itemset of any items with attributes $(S_{D1}, S_{D2}, \dots, S_{Dm})$, itemset $A \subseteq I_A$, I_A is the itemset of any items with attributes $(S_{A1}, S_{A2}, \dots, S_{An})$, C is 1-itemset of class attribute, a combined association rule set is represented as

$$\begin{cases} D + A_1 \Rightarrow C_{k1} \\ \vdots \\ D + A_i \Rightarrow C_{ki} \end{cases} \quad (1)$$

Here, “+” means itemsets appearing simutaniouly. Since one action may result in different classes while one class may correspond to different actions, $C_{k1} \dots C_{ki}$ rather than $C_1 \dots C_i$ are used in Eq. 1.

3 Combined Association Rule Mining

In order to make the combined association rules in a rule set containing the same non-actionable itemset, it is important to firstly discover frequent non-actionable itemsets. Once these itemsets are discovered, the relationships of frequent non-actionable itemsets with target classes and actionable attributes are mined. In the rule generation step, the conditional support [10] is employed to tackle data imbalance problem.

3.1 Interestingness Measures

For a single combined association rule $D + A_i \Rightarrow C_{ki}$, the conventional interestingness measures are its confidence and lift. However, these two interestingness

measures are not sufficient to mine actionable knowledge from combined association rule. We illustrate this problem using an example. For a discovered frequent pattern $D + A_i \Rightarrow C_{ki}$, suppose $Conf(D + A_i \Rightarrow C_{ki})$ is 60% and the expected confidence of C_{ki} is 30%. So the lift of this frequent pattern is 2, which is high enough in most association rule mining algorithms. However the confidence of $D \Rightarrow C_{ki}$ is 70%, which means objects with non-actionable attribute D have 70% probability to be class C_{ki} . On the other hand, if action A_i happens, objects with non-actionable attribute D only have 60% probability to be class C_{ki} . Obviously action A_i is negatively correlated to class C_{ki} with respect to non-actionable itemset D .

Hence, a new lift named conditional lift is defined as follows to measure the interestingness of a combined association rule.

$$ConLift = \frac{Conf(D + A_i \Rightarrow C_{ki})}{Conf(D \Rightarrow C_{ki})} = \frac{Count(D \cap A_i \cap C_{ki}) \cdot Count(D)}{Count(D \cap A_i) \cdot Count(D \cap C_{ki})} \quad (2)$$

where *ConLift* stands for the conditional lift of combined association rule $D + A_i \Rightarrow C_{ki}$. $Count(\times)$ is the count of the tuples containing itemset “ \times ”. Note that D , A_i , and C_{ki} are all itemsets so that $D \cap A_i \cap C_{ki}$ means D, A_i , and C_{ki} occur simultaneously.

Briefly, Eq. 2 is the lift of $D + A_i \Rightarrow C_{ki}$ with D as a pre-condition, which shows how much is the contribution of A_i in the rule.

3.2 Algorithm Outline

The combined association rule mining procedure in this paper consists of two steps. The first step is to find single rule composed of frequent itemsets. The second step is to extract interesting combined association rule sets. Since itemsets are treated as different groups, the time complexity of the algorithm is much lower than searching in the whole space of itemsets. In order to calculate the interestingness measures, the support count of each frequent itemset is recorded in the frequent itemset generation step. The outline of combined association rule mining is shown as follows:

1. Discovering frequent non-actionable itemsets I_D and the corresponding support counts C_D ;
2. For each frequent non-actionable itemsets I_D
3. Finding frequent itemsets including target class I_{DC} ;
4. Recording the support count C_{DC} for each I_{DC} ;
5. Calculating conditional support $ConSup(DC)$;
6. If $(ConSup(DC) > MinSup)$, for each I_{DC}
7. Finding candidate pattern of three kinds of itemsets I_{DCA} ;
8. Recording the support count C_{DCA} for each I_{DCA} ;
9. Calculating conditional support: $ConSup(DA)$;
10. Calculating $Conf$, $Lift$ and $ConLift$;
11. If $(Conf \geq min_c \ \& \ Lift \geq min_l \ \& \ ConLift \geq min_{cl})$
12. Adding the mined frequent itemsets to the rule set.

4 Case Study

Our proposed technique has been tested with real-world data in Centrelink, which is an Australian Government Service Deliver Agency delivering a range of Commonwealth services to the Australian public.

4.1 Business Background and Problem Statement

When customers receive Commonwealth payments to which they were not entitled, these payments become customer debt that must be recovered. The purpose of data mining in debt recovery is trying to make the customers to repay their debts in a shortened timeframe according to historical debt recovery data and customer demographics. From a technical point of view, the objective is to mine the combined association rule with respect to the demographic attributes and debt information of customers, the arrangement, and the target class. Suppose some customers with similar demographic attributes and debt information belong to different target classes under different arrangements, Centrelink will recommend an arrangement to assist them to pay off a debt in the shortest possible time. Note that an arrangement is an agreement between a customer and Centrelink officer on the method, amount and frequency of repayment.

4.2 Data Involved

The dataset used for the combined association rule mining is composed of customer demographic data, debt data and repayment data. The customer demographic data includes customer ID, gender, age, marital status, salary, and so on. The debt data includes the debt related information. The repayment data includes the debt recovery arrangement, debt repayment amount and debt repayment date. The class ID of each customer is defined by business experts based on the information in debt data and arrangement data.

4.3 Experimental Results

In our experiment, the frequent patterns of the demographic itemsets were first mined using standard Apriori algorithm [1] on demographic data. The *Conf*, *Lift* and *ConLift* can be calculated on each frequent itemset. In the experiments, we set $minconf = 0.45$, $minlift = 1.2$, and $minconlift = 1.2$. Using these parameters and the calculated interestingness measures, the interesting combined association rule sets are selected. In this case study, 28 rule sets are discovered, which include 111 single rules altogether. Selected results are shown in Table 1. For privacy reason, the benefit type, arrangement pattern and class ID are recoded in the experiments.

With the mined combined association rules, much actionable knowledge can be obtained. For example, suppose the priority of target class in this experiment is $C_2 > C_1 > C_3$, if a customer is with demographic attributes *MARITAL* : *SIN* & *Age* : 26y – 50y & *Earnings* : [\$200, \$400), the arrangement A_2 will be recommended to him/her with the greatest priority. If A_2 is impossible, A_1 will be recommended. The arrangement A_{10} is recommended with the least priority.

Table 1. Selected results of combined association rules

Demographics	Arg	Class	Conf	Lift	ConLift	IsRule
BENType:AAA & MARITAL:MAR & Age:65y+	A ₁	C ₁	0.46	1.31	1.70	Yes
BENType:AAA & MARITAL:MAR & Age:65y+	A ₂	C ₂	0.92	2.01	1.61	Yes
BENType:AAA & MARITAL:MAR & Age:65y+	A ₃	C ₂	0.91	1.97	1.58	Yes
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₁	C ₁	0.78	2.20	1.83	Yes
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₄	C ₁	0.29	0.83	0.69	No
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₁₁	C ₁	0.50	1.42	1.18	No
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₂	C ₂	0.79	1.72	2.15	Yes
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₇	C ₃	0.42	2.27	2.04	No
MARITAL:SIN & Age:26y-50y & Earnings:(\$200, \$400)	A ₁₀	C ₃	0.46	2.47	2.23	Yes
BENType:BBB & Earnings:0 & Children:0	A ₅	C ₁	0.77	1.67	2.97	Yes
BENType:BBB & Earnings:0 & Children:0	A ₇	C ₃	0.64	3.44	1.47	Yes
BENType:BBB & Earnings:0 & Children:0	A ₈	C ₃	0.50	2.70	1.16	No

5 Related Work

The work in this paper is obviously different from any previous association rule mining algorithms. Hilderman et al. [4] extended simple association rule to mine characterized itemsets. Employing the concept of “share measures”, their algorithm may present more information in terms of financial analysis. Different from Hilderman et al.’s algorithm, each single rule in this paper is associated with a target class to provide ordered action list.

Ras et al. [78] proposed to mine action rules. They divided the attributes in a database into two groups: stable ones and flexible ones. The action rules are extracted from a decision table given preference to flexible attributes. However, in their algorithm, only flexible attributes appear in the mined rules.

In combined association rule mining, each single combined association rule is similar to class association rule (CAR), which was proposed by Liu et al. [5] in 1998. However, in [5], the class association rules are mined to build associative classifier while the combined association rule sets are mined for direct actions rather than prediction.

Data imbalance problem has attracted much research attention in recent years. Arunasalam and Chawla [2] studied the data imbalance in association rule mining. Their algorithm is focused on the imbalanced distribution of one attribute, the target class. In our algorithm, the data imbalance problem occurs not only on target class but also actionable attributes.

6 Summary

This paper proposes an efficient algorithm to mine combined association rules on imbalanced datasets. Unlike conventional association rules, our combined association rules are organized as a number of rule sets. In each rule set, single combined association rules consist of different kinds of attributes. A novel frequent pattern generation algorithm is proposed to discover the complex inter-rule and intra-rule relationships. Data imbalance problem is also tackled in this paper.

The proposed algorithm is tested in a real world application. The experimental results show the effectiveness of algorithm.

Acknowledgments

We would like to thank Mr. Hans Bohlscheid, Business Manager and Project Manager of Centrelink Business Integrity and Information Division, for his on-going support, and Mr. Fernando Figueiredo and Mr. Peter Newbigin for their assistance in extracting Centrelink data.

This work was supported by Discovery Projects DP0449535, DP0667060, DP0773412, Linkage Project LP0775041 from Australian Research Council (ARC) and Early Career Research Grants from University of Technology, Sydney (UTS).

References

1. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: VLDB 1994, Santiago de Chile, Chile, pp. 487–499 (1994)
2. Arunasalam, B., Chawla, S.: Cccls: a top-down associative classifier for imbalanced class distribution. In: KDD 2006, Philadelphia, PA, USA, pp. 517–522 (2006)
3. Gu, L., Li, J., He, H., Williams, G., Hawkins, S., Kelman, C.: Association rule discovery with unbalanced class distributions. In: Gedeon, T.D., Fung, L.C.C. (eds.) AI 2003. LNCS (LNAI), vol. 2903, pp. 221–232. Springer, Heidelberg (2003)
4. Hilderman, R.J., Carter, C.L., Hamilton, H.J., Cercone, N.: Mining market basket data using share measures and characterized itemsets. In: Wu, X., Kotagiri, R., Korb, K.B. (eds.) PAKDD 1998. LNCS, vol. 1394, pp. 159–170. Springer, Heidelberg (1998)
5. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: KDD 1998, New York, NY, USA, pp. 80–86 (1998)
6. Liu, B., Hsu, W., Ma, Y.: Identifying non-actionable association rules. In: KDD 2001, San Francisco, CA, USA, pp. 329–334 (2001)
7. Ras, Z.W., Alicija, W.: Action-rules: How to increase profit of a company. In: Zighed, A.D.A., Komorowski, J., Żytkow, J.M. (eds.) PKDD 2000. LNCS (LNAI), vol. 1910, pp. 587–592. Springer, Heidelberg (2000)
8. Ras, Z.W., Ras, Z.W., Tzacheva, A.A., Tsay, L.-S., Giirdal, O.: Mining for interesting action rules. In: Tzacheva, A.A. (ed.) IAT 2005, pp. 187–193 (2005)
9. Yang, Q., Yin, J., Ling, C., Pan, R.: Extracting actionable knowledge from decision trees. *IEEE TKDE* 19(1), 43–56 (2007)
10. Zhang, H., Zhao, Y., Cao, L., Zhang, C.: Class association rule mining with multiple imbalanced attributes. In: Orgun, M.A., Thornton, J. (eds.) AI 2007. LNCS (LNAI), vol. 4830, pp. 582–587. Springer, Heidelberg (2007)
11. Zhao, Y., Zhang, H., Figueiredo, F., Cao, L., Zhang, C.: Mining for combined association rules on multiple datasets. In: Proceedings of the KDD 2007 Workshop on Domain Driven Data Mining, San Jose, CA, USA, pp. 18–23 (2007)

Enriching WordNet with Folksonomies

Hao Zheng, Xian Wu, and Yong Yu

Shanghai Jiao Tong University
Shanghai 200240, China
{zhenghao,wuxian,yyu}@apex.sjtu.edu.cn

Abstract. Manually constructed thesauri are not updated regularly, so they are hard to catch the fast emergence of new words. Moreover, the vocabularies of the professionals who construct the thesauri may not completely match the vocabularies of normal users. Recently, Folksonomy services are very popular and highly sensitive to information drift and the change of users' vocabularies. In this paper, we explore a method for enriching formal thesauri with informal Folksonomies. We demonstrate our method by semi-automatically enriching WordNet with new words emerging from a social bookmark service. Tags are related to each other by the subsumption relationships extracted from Folksonomies. New words are recommended to be placed in appropriate synsets of the WordNet hierarchy. An initial evaluation on our experimental result shows the effectiveness of our method.

Keywords: Folksonomy, Social Annotation, WordNet.

1 Introduction

WordNet [1] is a huge hand-built thesaurus and has been widely employed in many tasks. But with the fast emergence of new words in our times, WordNet may not be able to catch the fast pace of the changes of vocabularies, and its lack of new words apparently has become a more and more serious problem. Recently, Folksonomies are very popular web services and they allow annotators to use freely chosen strings as tags without any apriori thesaurus. A cursory analysis of the tags reveals that the tags can perfectly reflect the emergence of new words, most probably owing to the popularity of such services. But the shortcoming of Folksonomies is their unrestricted use of words and their flat structure, which obstructs their direct application.

This paper describes a method for semi-automatic enrichment of WordNet with Folksonomies. Our method aims to derive the emergent semantics of Folksonomies and places the new words found in Folksonomies to appropriate positions in WordNet according to the subsumption relationships [2] extracted from Folksonomies. Although it is relatively simpler than more sophisticated methods, the technique of subsumption itself has long been employed in constructing concept hierarchies and has achieved some success [2]. In our preliminary experiment, the method appears to be effective.

2 Related Work

2.1 Folksonomies

Social annotations are very popular in the past few years. What's special is their use of keywords called "tags". These user-created annotations were coined the name "Folksonomy", a combination of "folk" and "taxonomy" [3]. In our work, the most important strength of Folksonomies is that they directly reflect users' vocabularies. Traditionally, dictionaries or thesauri can only represent the vocabularies of lexicographers, and there is a great gap between these two vocabularies. [4] recommends using Folksonomies as the start of professionally designed controlled vocabularies and also makes a good analogy between Folksonomies and "desire lines". Some early reviews on Folksonomies have been published [5,3,6]. [7] introduced the social dimension into a unified model of social networks and semantics. [8] gave a detailed analysis of annotation data in Delicious. [9,10] proposed to integrate Folksonomies with the Semantic Web.

2.2 Hyponym/Hypernym Relation Extraction

The current automated approaches for Hyponym/Hypernym relation extraction are classified into two classes: 1). approaches based on **lexical or syntactic analysis**, which rely on the lexical or syntactic patterns to discover relationships between words in text. [11] described a method by use of lexico-syntactic patterns manually identified. [12] applied the pattern learning method to Part-Whole relations. [13] employed Formal Concept Analysis for a specific domain. 2). approaches based on **co-occurrence distributions of words**, which just treat text as bag-of-words without depending on any syntactic features. [2] introduced a document-based definition of *subsumption*. We adopt the work to Folksonomies. [14,15] derived subsumption relations on text associated with image captions and descriptions. [16,17] employed subsumption to enhance effective browsing of social annotations.

3 Exploiting Folksonomies

Folksonomy is usually formalized into a tripartite model [7], consisting of 3 disjoint sets $U = \{u_1, u_2, \dots, u_K\}$, $T = \{t_1, t_2, \dots, t_N\}$, and $R = \{r_1, r_2, \dots, r_M\}$ corresponding to K users, N tags, and M web resources annotated. What interests us is the co-occurrence of users, resources and tags, which is defined as a set of triples (*user, resource, tag*). However, the tags in Folksonomies are unrestricted and in a single flat namespace. In this section, we describe the techniques for solving these problems in detail.

3.1 Tag Selection

The tags in Folksonomies vary greatly in their quality, due to lack of control. After analyzing a large amount of tags, we conclude some causes of low quality:

- No support for Hierarchies, e.g. “coding.languages.php”, “software.os.linux”.
- No support for Spaces, e.g. “programminglanguage”, “dosomethingwiththis”.
- Numbers, Letters and other Index-like strings, e.g. “2004-10”, “A”, “1.11.05”.
- Personal Choices, e.g. “_to_read”, “~4_wednesday”, “{darren}”, “*temp”.

In order to filter out such tags of low quality, we take into consideration the consensus of users on a specific tag. The number of users who make annotation with the tag shows the degree to which the community reaches a consensus. This is analogous to common life: the more people the word is used by, the more likely it will appear in dictionaries. We find that the consensus of tag choices by users follow a power law distribution. In our dataset, 2283 tags are only used by one user, while only 2116 tags are shared by more than 20 users. We choose the number of 20 as tag selection threshold. After tag selection, we regard the remaining tags to be of “high” quality. The next subsection discusses how to relate these tags in a flat namespace to each other.

3.2 Subsumption Relationships

The Hyponym/Hypernym relation is the most important one for the organization of WordNet. We aim to extract such relationship from co-occurrence of tags. The triples (*user, resource, tag*) are converted to a tag-by-resource matrix $C_{n \times m}$. Each row of C is a tag. Each column of C is a resource. C_{ij} denotes the times of tag t_i used to annotate resource r_j . t_x is said to *subsume* t_y , if t_y is used to annotate a small portion of the resources that are annotated by t_x .

$$P(t_x|t_y) \geq \tau, P(t_y|t_x) < 1, \tag{1}$$

$$P(t_x|t_y) = \frac{f(t_x, t_y)}{f(t_y)}, \tag{2}$$

$$f(t_y) = \sum_{j=1}^M C_{yj}, \quad f(t_x, t_y) = \sum_{j=1}^M \min(C_{xj}, C_{yj}). \tag{3}$$

where τ ($0 < \tau \leq 1$) denotes the relaxed subsumption threshold, $f(t_y)$ denotes the number of resources annotated with t_y , and $f(t_x, t_y)$ denotes the number of resources annotated with both t_x and t_y . Different τ generates different number of Hyponym/Hypernym pairs. We set τ to 0.8 conservatively, comparable to the value determined empirically by [2]. The last problem to be tackled is the issue of placing new words into the appropriate position in WordNet.

3.3 Enriching WordNet

Words in WordNet are organized into synsets, which are the basic elements of WordNet. One word may be included in several synsets. The relations between synsets include among others, Synonym, Hypernym, Hyponym, Meronym, Holonym, etc. To place new words in appropriate synsets, we define the similarity between a tag and a synset as $sim(synset, tag) = \max_{w \in synset \cap T} f(w, tag)$, where $f(w, tag)$ is defined in [3]. It calculates the co-occurrence of the tag and

the words appearing both in the tag set T and the synset. It could serve a measure of how close the tag is to the synset.

When comparing the Hyponym/Hypernym t_0/t_1 pairs discovered by subsumption to the noun hierarchy of WordNet, 4 kinds of outcome are possible:

1. Both t_0 and t_1 are in WordNet. If the relationship is already in WordNet, it justifies the subsumption method, so these pairs are used as evaluation set of our approach (Sect. 4.1). If the relationship does not exist, it is treated as a false pair generated by subsumption, assuming all Hyponym/Hypernym relations between existing words are already in WordNet, although indeed, it could be a potential relationship missing in WordNet.
2. t_0 is in, but t_1 is not present. For each Hypernym synset s of each sense of t_0 , $sim(s, t_1)$ is calculated. The co-occurred synsets ordered by $sim(s, t_1)$ are recommendations of where t_1 be placed. Our method recommends that t_1 be placed either in these existing synsets or in a newly created sibling synset, i.e. at the same level of them. If all Hypernym synsets s have $sim(s, t_1) = 0$, a new synset without recommendation for its position is created for t_1 .
3. t_1 is in, but t_0 is not present. This case is contrary to case 2.
4. Both t_0 and t_1 are not present. Initially, there may be many pairs in this case. However, some t_0 or t_1 also belong to some pairs in case 2 or 3, so after these tags are placed in WordNet by case 2 or 3, these pairs could be converted to case 2 or 3 iteratively. Finally, the pairs still left in this case are discarded without processing.

4 Experiment and Evaluation

In our experiment, WordNet 2.0 and the social bookmark service Delicious are used. We collected a sample of Delicious data, which consists of 479,035 annotation triples made by 29,221 users on 16,963 web resources with 8,445 distinct tags.

After tag selection, 2116 tags of high quality, i.e. shared by more than 20 users, are involved in subsumption checking. About 50% of them (1143 of 2116) are already in WordNet, and the others (973 of 2116) can be considered as good candidates for new words. By setting τ to 0.8 in (1), we extracted 987 pairs of Hyponym/Hypernym relations. These pairs are divided into 4 cases discussed in Sect. 3.3 (Table 1). It shows, after iteratively moved to other cases, only 1 pair remains in case 4 and is discarded without processing.

Evaluation presents a challenge, because no objective standard exists. In previous work, evaluation is usually conducted manually by human judges. In our work, we find that the tags already in WordNet could provide a good ground truth. So we conduct both objective and subjective evaluations.

4.1 Evaluation by WordNet

For 274 subsumption pairs whose Hyponym and Hypernym are both in WordNet already, 192 pairs (*precision* 0.70) are found in WordNet. As for recall, Hyponym/Hypernym relations are tested pairwise between these 1143 tags, and 3231 relations are found. The *recall* 0.06 is rather low, since most relations found

Table 1. The number of subsumption pairs in 4 cases: both tags are in, only Hyponym is in, only Hypernym is in, and neither tag is in WordNet

	Both	Hyponym	Hypernym	Neither
initially	274	92	425	196
finally	274	128 (+36)	584 (+159)	1 (-195)

in WordNet are too general to be extracted by co-occurrence. For example, in relation “t-shirt/object”, “object” is too general to be used with “t-shirt” in Delicious, thus the relation could not be extracted by subsumption. By removing all relations whose Hypernym is too general, the recall increases to 0.65.

4.2 Evaluation by Human

Manual evaluation is only conducted on the subsumption pairs in which only Hyponym is registered (128 pairs), or in which only Hypernym is registered (584 pairs). Human judges mark each pair t_0/t_1 by the type of relation between t_0 and t_1 . 4 options are provided: “Hyponym/Hypernym”, “Meronym/Holonym”, “Same”, and “Unknown”, in accordance with [2]. [2] applied subsumption to free text, and it provided us with a valuable baseline.

Table 2. Human evaluation of the subsumption pairs derived from Delicious. Each pair is marked by one of 4 options.

	Hyponym/Hypernym	Meronym/Holonym	Same	Unknown
Baseline	23%	49%	8%	19%
only Hyponym in	38%	25%	32%	5%
only Hypernym in	45%	28%	17%	10%
Average	42%	26%	25%	7%

In general, the results are encouraging (Table 2). We notice that 19% of the relations are classified as “Unknown” in baseline. By contrast, only 7% in average are judged as “Unknown” in our method. It shows the subsumption pairs produced from Folksonomies are relatively high quality, compared to those produced from free text. We expect this to occur because when subsumption was used upon free text directly, a crucial step would be term selection. When subsumption is employed upon Folksonomies, however, the problem no longer exists. Users of Folksonomies undertake the task for us, i.e. choose the tags best summarizing the web resources, presumably better than any automated process. It justifies the effectiveness of adapting subsumption to Folksonomies.

5 Conclusion

In this paper, we presented a method for enriching formal thesauri with informal Folksonomies. Traditional human-built thesaurus could not catch the pace of fast emergence of new words. On the other hand, Folksonomies are very popular web

services recent years and are highly sensitive to information drift and the change of users' vocabularies. Our method selects tags of high quality, extracts the subsumption relation among them, and then places them into the appropriate synsets in WordNet. Through preliminary experiment, we show that our method is effective.

Acknowledgement. This work is supported by National Natural Science Foundation of China under the grant number of 60473122. We also thank the anonymous reviewers for their helpful comments.

References

1. Fellbaum, C.: Wordnet: an electronic lexical database. MIT Press, Cambridge (1998)
2. Sanderson, M., Croft, B.: Deriving concept hierarchies from text. In: SIGIR, Berkeley, California, United States, pp. 206–213 (1999)
3. Smith, G.: Atomiq: Folksonomy: social classification (August 2004)
4. Merholz, P.: Metadata for the masses (October 2004), <http://www.adaptivepath.com/publications/essays/archives/000361.php>
5. Hammond, T., Hannay, T., Lund, B., Scott, J.: Social bookmarking tools (i) - a general review. D-Lib Magazine 11(4) (2005)
6. Mathes, A.: Folksonomies - cooperative classification and communication through shared metadata (December 2004), <http://www.adammathes.com/academic/computer-mediated-communication/folksonomies.html>
7. 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. 122–136. Springer, Heidelberg (2005)
8. Golder, S., Huberman, B.A.: The structure of collaborative tagging systems (2005)
9. Specia, L., Motta, E.: Integrating Folksonomies with the Semantic Web. In: Francioni, E., Kifer, M., May, W. (eds.) ESWC 2007. LNCS, vol. 4519, pp. 624–639. Springer, Heidelberg (2007)
10. Damme, C.V., Hepp, M., Siorpaes, K.: Folksonology: An integrated approach for turning folksonomies into ontologies. In: SemNet, pp. 57–70 (2007)
11. Hearst, M.A.: Automatic acquisition of hyponyms from large text corpora. In: Association for Computational Linguistics, pp. 539–545 (1992)
12. van Hage, W.R., Kolb, H., Schreiber, G.: A method for learning part-whole relations. In: Cruz, I., Decker, S., Allemang, D., Preist, C., Schwabe, D., Mika, P., Uschold, M., Aroyo, L.M. (eds.) ISWC 2006. LNCS, vol. 4273, pp. 723–735. Springer, Heidelberg (2006)
13. Cimiano, P., Hotho, A., Staab, S.: Learning concept hierarchies from text corpora using formal concept analysis. JAIR 24, 305–339 (2005)
14. Dakka, W., Ipeirotis, P.G., Wood, K.R.: Automatic construction of multifaceted browsing interfaces. In: CIKM, Bremen, Germany, pp. 768–775 (2005)
15. Clough, P., Joho, H., Sanderson, M.: Automatically Organising Images using Concept Hierarchies. In: Proc. of the SIGIR Workshop on Multimedia Information Retrieval (2005)
16. Li, R., Bao, S., Yu, Y., Fei, B., Su, Z.: Towards effective browsing of large scale social annotations. In: WWW, Banff, Alberta, Canada, pp. 943–952 (2007)
17. Schmitz, P.: Inducing ontology from flickr tags. In: WWW 2006, Edinburgh, Scotland (May 2006)

A New Credit Scoring Method Based on Rough Sets and Decision Tree

XiYue Zhou, DeFu Zhang, and Yi Jiang

Department of Computer Science, Xiamen University, Xiamen 361005, China
elice.zhou@gmail.com, dfzhang@xmu.edu.cn, jiangyi@xmu.edu.cn

Abstract. Credit scoring is a very typical classification problem in Data Mining. Many classification methods have been presented in the literatures to tackle this problem. The decision tree method is a particularly effective method to build a classifier from the sample data. Decision tree classification method has higher prediction accuracy for the problems of classification, and can automatically generate classification rules. However, the original sample data sets used to generate the decision tree classification model often contain many noise or redundant data. These data will have a great impact on the prediction accuracy of the classifier. Therefore, it is necessary and very important to preprocess the original sample data. On this issue, a very effective approach is the rough sets. In rough sets theory, a basic problem that can be tackled using rough sets approach is reduction of redundant attributes. This paper presents a new credit scoring approach based on combination of rough sets theory and decision tree theory. The results of this study indicate that the process of reduction of attribute is very effective and our approach has good performance in terms of prediction accuracy.

Keywords: Data Mining, Credit Scoring, Rough Sets, Decision Tree, Attribute Reduction.

1 Introduction

The credit scoring model has been used in commercial and consumer loan for a few decades. Numerous methods have been presented in many literatures to develop the credit scoring model. Those models include traditional statistical models (e.g.: logistic regression [4]), nonparametric statistical models (e.g., k-nearest neighbor [5]), decision trees [11, 12] and neural network models [3]). All these models are widely used. But they didn't all process the original sample data when they were used to build the credit scoring model. It is necessary and very important to preprocess the original sample data to eliminate redundant data and noise data, etc. In this paper, we finish the process by using rough set theory [9]. In addition, due to higher prediction accuracy and generating automatically classification rules [7], we will build the credit scoring model by using decision tree method.

The rest of this paper is organized as follows. We will briefly explain the basic concepts of rough set in section 2, and discuss the decision tree algorithm C4.5 [11, 12]

in section 3. The design and generating of our model will be illustrated in section 4. In section 5 we will analyze the experimental results of the credit scoring model in this paper and compare with other methods in the prediction accuracy. Finally, section 6 addresses the conclusion and discusses the possible future research work.

2 Rough Sets Theory

2.1 The Information System

An information system can be represented as follows:

$$S = \langle U, A, V, f \rangle \tag{1}$$

Where, U is a non-empty finite set of objects called universe; A is a non-empty finite set of attributes; V_a is the range of the attribute a ; $V = \bigcup_{a \in A} V_a$; $f : U \times A \rightarrow V$ is the information function such that $f(x, a) \in V_a$, for any $a \in A$ and $x \in U$.

2.2 Indiscernibility Relation

The indiscernibility relation [10, 13, 14] is an equivalence relation on the set U and can be defined as follows:

There is an indiscernibility relation $INP(P)$, for arbitrary attribute subset $P \subseteq A$:

$$IND(P) = \{ \langle x, y \rangle \in U \times U \mid \forall a \in P, f(x, a) = f(y, a) \} \tag{2}$$

If $\langle x, y \rangle \in IND(P)$, that means objects x and y are indiscernible with attribute set P .

2.3 Reduction of Concept and the Core

In real-world application, we are often required to eliminate irrelevant or redundant attributes; meanwhile, we must maintain the primary areas of the information system. This problem refers to two basic concepts: *reduction* and *core* [9, 13, 14].

2.4 Discernibility Matrix

Discernibility matrix [13, 14] is a very important concept in rough set theory. Discernibility matrix can be used to complete attributes reduction.

Definition 1. [13, 14]: Given an information system S , $U = \{x_1, x_2, \dots, x_n\}$ is the set of objects, $C = \{c_1, c_2, \dots, c_m\}$ is the predictive attributes set, D is the class attribute. Discernibility matrix is denoted by $M(S)$, whose elements are as follows:

$$m_{ij} = \begin{cases} \{a \mid a \in C : f(x_i, a) \neq f(x_j, a) \wedge f(x_i, D) \neq f(x_j, D)\} & \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $i, j = 1, 2, \dots, n$, here $n=|U|$.

3 C4.5

The ID3 [11] is a famous algorithm to construct a decision tree. And the C4.5 [12] is the extended version of the ID3. The C4.5 mainly contains two phases: generating an initial decision tree and pruning the initial decision tree.

3.1 Generating Decision Tree

The original ID3 algorithm used a criterion called *gain* to select the test attribute. The criterion *gain* refers to the concept *entropy* in information theory [6, 10].

3.2 Pruning Decision Tree

Since there is a less objects to work with after each decision node split, it is necessary to prune the decision tree to get a better accuracy of prediction. On this issue, C4.5 uses a specific technique to estimate the prediction error rate. This technique is called *pessimistic error pruning* [11].

4 New Credit Scoring Model

4.1 Design

In this paper, the credit scoring model is built based on the combination of rough set theory and decision tree theory. Firstly, our model preprocesses the sample data by using rough set; then generates the credit scoring model by using C4.5. This approach can bring many benefits. Rough set can not only remove redundant data but also simplify the dimension of input information space by discovering the relation among all data.

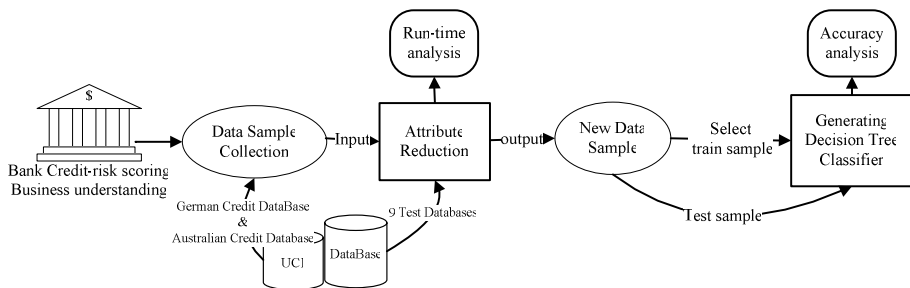


Fig. 1. Experiment procedures

Fig. 1 illustrates the self-explanatory experimental procedure for generating the credit scoring classifier. The details will be presented in next section.

4.2 Reduction of Attributes

Before the classifier generated, we will preprocess the original sample data. The process mainly processes the redundant attribute. In this paper the process is called *reduction of attributes*. We process the redundant attributes by using the algorithm in Wang and Pei’s paper [14] (denoted by WPA in this paper). In this paper we improved the WPA, and denotes by **Algorithm 1**. The **Algorithm 1** is faster than WPA in run-time.

Before discussion the **Algorithm 1**, we give a theorem in Boolean algebra as follows:

Theorem 1. [1]: $a \wedge (a \vee b) \leftrightarrow a$.

We will not prove the **Theorem 1** because it is easy to prove it.

Algorithm 1. Compute all reductions of attributes.

Input: Information system $S = \langle U, A, V, f \rangle$;

Output: All reductions of attributes of the information system $S = \langle U, A, V, f \rangle$;

Procedure:

1. construct the discernibility matrix $M(S) = [c_{ij}]_{n \times n}$ of information system S , and construct $M = \{c_{ij} \mid 0 \leq i, j \leq n - 1\}$;
2. construct discernibility function $f_M(S)$ according to M , the form of the function $f_M(S)$ is disjunctive normal form;
- 3*. reduce the *disjunctive normal form* according to the **Theorem 1**;
4. $FunctionMinDnf(M, DNF_M)$;
5. get all reduction of attributes from DNF_M .

Algorithm 2. $FunctionMinDnf(M, DNF_M)$

Input: the discernibility matrix $M(S) = [c_{ij}]_{n \times n}$ of information system S , and the set $M = \{c_{ij} \mid 0 \leq i, j \leq n - 1\}$

Output: DNF_M

Procedure:

1. $DNF_{M1} = \emptyset$; $DNF_{M2} = \emptyset$;
2. if $M = \emptyset$ return \emptyset ;
3. if $|M| = 1$ return $DNF_{M1} = M$;
4. divide M into M_1, M_2 ; $FunctionMinDnf(M_1, DNF_{M1})$;
 $FunctionMinDnf(M_2, DNF_{M2})$;
5. Construct $R = \{d_1 \wedge d_2 \mid d_1 \in DNF_{M1}, d_2 \in DMF_{M2}\}$
6. $DNF_M = reduction\ of\ R$; return DNF_M ;

The definition of *reduction of R*, the definition of discernibility function $f_M(S)$, and the definition of the *DNF* are in [14]. The step 3 with notation ‘*’ is our optimization step in algorithm 1, and other steps is the same as the WPA.

The time complexity of WPA is $T = O(|A|(K^4 \log |U| + |U^2|))$, where $K = \text{Max} \{ \text{Card}(\text{DNF}_G) | G \subseteq N \}$ [14]. The time complexity is decreased in the Algorithm 1. The value of *K* will be greatly decreased, because the size |M| of the set M is greatly decreased. And the rank of *K* is 4, so the Algorithm 1 will evidently reduce the running time. The experimental results of the efficiency of reduction of attribute will be given in the section 5.

4.3 Generating Classifier

After reduction of attributes, we got a new sample data set. We can randomly select a majority of the instances as train sample of decision tree classifier from the new sample. We build the credit scoring model by using the Algorithm C4.5.

5 Experimental Results

5.1 Efficiency of Reduction of Attributes

To compare with WPA, we selected the same databases as Wang and Pei’s. We selected nine databases from the database of UCI machine learning. The experiment was completed on the same PC (Intel-Celeron, 2.4GHz, 256MB RAM, WinXP Professional). We obtained the same results by the two methods. And the experimental results on the running time are reported at the Table 1:

Table 1. Comparison of the running time of reduction algorithms

Name of database	Number of instances	Number of attributes	Algorithm 1 Running time(s)	WPA Running time(s)
Postoperative Patient	90	9	0.000	0.102
Hayes-Roth	132	6	0.031	0.155
Balance-scale	625	5	0.156	5.410
Teaching Assistant Evaluation	152	6	0.035	0.226
Zoo	101	17	0.046	0.150
Tic-Tac-Toe Endgame	958	10	0.453	17.885
Car Evaluation	1728	7	0.846	68.756
BUPA liver disorders	345	7	0.062	1.520
Monk's Problems(1)	432	7	0.093	2.633
Monk's Problems(2)	432	7	0.109	2.598
Monk's Problems (3)	432	7	0.093	2.375

According to the Table 1, obviously, the **algorithm 1** has a higher efficiency comparing with WPA on the running time. It shows that it is effective that we improved the WPA. The analysis about time complexity of the **Algorithm 1** is validated by the result.

5.2 Prediction Accuracy Analysis

The two databases of in our experiments are from the UCI Machine Learning Repository [8]: German Credit database and Australian Credit database. For the German Credit, there are in all 1000 instances which contain 700 good credit instances and 300 bad credit instances, and each instance consists of 20 predictive attributes and 1 class attribute. For the Australian Credit, there are in all 690 instances which contain 303 good credit instances and 387 bad credit instances, and each instance consists of 14 predictive attributes and 1 class attribute.

Table 2. Description of Databases from the UCI Machine Learning

Name	Instance	Predictive Attributes	class attribute	Good credit	Bad credit
German Credit	1000	20	1	700	300
Australian Credit	690	14	1	303	387

We respectively test the two databases by using the rough set & C4.5 method (denoted by RSC) and the single C4.5 with two different ratios. The two ratios are 7:3 and 8:2 between the size of train sample and the size of test sample. We had 20 experiments by using RSC and the single C4.5 for each database. The process of choice train sample is stochastic. Remainder instances of the database are used for test sample after choosing train sample. The experimental results are respectively reported at the Table 3 and at the Table 4.

Table 3. (UCI—German Credit) prediction accuracy

Methods	Number of predictive attribute	Max prediction accuracy (%)	Min prediction accuracy (%)	Average of prediction accuracy (%)
C4.5 (7:3)	20	74.0	68.0	72.0
RSC (7:3)	12	80.67	72.0	78.67
C4.5 (8:2)	20	74.5	68.0	73.5
RSC (8:2)	12	82.0	72.0	79.5

Table 4. (UCI—Australian Credit) prediction accuracy

Methods	Number of predictive attribute	Max prediction accuracy (%)	Min prediction accuracy (%)	Average of prediction accuracy (%)
C4.5(7:3)	14	88.12	81.26	85.31
RSC(7:3)	11	90.68	83.47	87.78
C4.5(8:2)	14	88.41	81.26	85.45
RSC(8:2)	11	90.95	84.78	88.21

According to the Table 3, only 12 of all 20 predictive attributes in RSC were used to build the credit scoring model after the reduction of attributes. Comparing with the single C4.5 method, the prediction accuracy of the RSC method are both evidently improved with the radio 7:3 and with 8:2. The average prediction accuracy is heightened

about 6%~7%. The RSC method had a good performance for the German Credit database. From the Table 4, we can find that the RSC has also a good performance for the Australian Credit database.

By comparing with the single C4.5, we conclude that the RSC method have a good performance for the two databases. The reduction of attribute not only reduced the dimension of the decision table (the original sample), but also enhanced the prediction accuracy of the credit scoring model. *Reduction of attribute* before building model is very effective. *Reduction of attribute* plays a very important role in the process of building the credit scoring model.

However, we must notice the index *min prediction accuracy* in the Table 3 and Table 4. The *min prediction accuracy* is low in the RSC method, which indicates that the stability of RSC is not good. We think that it is caused by the algorithm C4.5 by theoretical analysis. We can get the worst instance, and the prediction accuracy is just 0%. Suppose the 700 train instances are all from the 700 good instances and the 300 test instances are all from the 300 bad instances, which is possible though the probability is very tiny. If it happened, the generated decision tree would just have one node, and the *class* of the node is *good*. So we would get the 100% *error rate* when we test all bad instances in the test experiment. Hence, we can conclude that the prediction accuracy of model is associated with the ratio of good instances and bad instances in the train sample.

5.3 Comparing with Other Methods

Because those methods on credit scoring problem in many papers often use the k-fold cross validation to complete experiment, we tested again the prediction accuracy of our model by using the 10-fold cross validation by using the RSC method to compare with other methods. The result is the average of the accuracy determined for each of the 10 independent stochastic data set partitions. For the two databases, we all had 10 experiments with the 10-fold cross validation (10x10-CV). The results are reported at the Table 5.

Table 5. The accuracy rates (%) with the 10-fold cross validation for German credit database and Australian credit database by using the RSC method

	No.1	No.2	No.3	No.4	No.5	No.6	No.7	No.8	No.9	No.10	Avg.
German	79.9	79.4	79.9	80.2	79.0	79.8	79.2	79.6	79.8	79.5	79.63
Australian	88.55	87.97	88.26	88.55	88.70	88.26	88.59	88.99	88.84	88.70	88.54

At the present, many data mining techniques such as neural networks and genetic programming and SVM-based are successfully applied to build the credit scoring model, and they usually have good prediction accuracy. Therefore, the RSC method will be compared with these methods for the German Credit database and Australian Credit database.

We will compare RSC with the single C4.5, BPN (Back-propagation Neural Network), GP (Genetic Programming) [6], SVM+GA (Support Vector Machine +Genetic

Algorithm) [2]. The results of the two databases are summarized in the Table 6 by using RSC, single C4.5, BPN, GP, and SVM+GA. Where, the results of BP, GP, and SVM+GA are from the paper [2].

In the paper [2], the two credit scoring databases are partitioned into training and independent test sets by the same 10-fold cross validation procedure. The GP specific parameters for the set two credit datasets areas follows: population size is 250, reproduction rate is 0.2, crossover rate is 0.7, mutation rate is 0.08, and maximum number of generations is 2000-3000; For the BP model, several options of the neural network configurations are tested, in which 14-32-1 and 24-43-1 respectively for the Australian data and German data are selected to obtain better results. Additionally, the learning rate and momentum are set to 0.8 and 0.2, respectively; For C4.5 and SVM+GA, it chooses their default settings.

Table 6. Result summary with the 10-fold cross validation for German credit database and Australian credit database

Method	German Avg. (%)	Australia Avg. (%)
RSC	79.63	88.54
C4.5	73.50	85.31
BPN	77.83	86.83
GP	78.10	87.00
SVM+GA	77.92	86.90

On the basis of the results of Table 6, we can conclude that the RSC method in our study outperforms to other methods for Australian Credit database and German Credit database. It indicates that the RSC method is effective and successful to build the credit scoring model in this paper.

6 Conclusions and Future Works

The Data Mining technique is a very effective approach to research the financial orderliness and quickly make decision. The credit scoring model based on rough set and decision tree in this paper fully exhibits the advantages of rough set and decision tree. On the basis of the result of the section 5, we can conclude that the *reduction of attribute* in this paper is a very important and effective instrument to improve the prediction accuracy. Reducing the redundant attributes not only avoids the harmful data to impact the prediction accuracy but also reduces the cost of calculation in the process of building credit scoring model. Moreover, the RSC method has higher prediction accuracy than the single C4.5, BP, GP, and SVM+GA on the benchmarks. The RSC method is effective and successful on the credit scoring problem in this paper. However, the method has some limitations and need to improve the stability even as our analysis in the section 5. In future works, we can try to improve the RSC method combining with the Boosting Algorithm or Bagging Algorithm to get higher accuracy,.

Acknowledgements

The paper is supported by the National Nature Science Foundation of China (Grant no. 60773126) and the Province Nature Science Foundation of Fujian (Grant no. A0710023) and academician start-up fund (Grant No. X01109) and 985 information technology fund (Grant No. 0000-X07204) in Xiamen University.

References

1. Hamilton, A.G.: Logic for Mathematicians. Cambridge University Press, Cambridge (1988)
2. Huang, C.-L., Chen, M.-C., Wang, C.-J.: Credit scoring with a data mining approach based on support vector machines. *Expert Systems with Applications* (2006), doi: 10.1016/j.eswa.2006.07.007
3. Desai, V.S., Crook, J.N., Overstreet, G.A.: A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research* 95(1), 24–37 (1996)
4. Henley, W.E.: Statistical aspects of credit scoring. Dissertation, The Open University, Milton Keynes, UK (1995)
5. Henley, W.E., Hand, D.J.: A k-nearest neighbor classifier for assessing consumer credit risk. *Statistician* 44(1), 77–95 (1996)
6. Koza, J.R.: Genetic programming: On the programming of computers by means of natural selection. The MIT Press, Cambridge, MA (1992)
7. Kantardzic, M.: Data Mining: Concept, Models, Methods, and Algorithms. IEEE Press, America (2002)
8. Murphy, P.M., Aha, D.W.: UCI Repository of Machine Learning Database (2001), <http://www.ics.uci.edu/~mlearn/MLRepository.html>
9. Pawlak, Z.: Rough sets. *International Journal of Computer and Information Science* 11(5), 341–356 (1982)
10. Hu, Q., Zhao, H., Xie, Z., Yu, D.: Consistency Based Attribute Reduction. In: Zhou, Z.-H., Li, H., Yang, Q. (eds.) PAKDD 2007. LNCS (LNAI), vol. 4426, pp. 96–107. Springer, Heidelberg (2007)
11. Quinlan, J.R.: Introduction of decision trees. *Machine Learning* 1(1) (1986)
12. Quinlan, J.R.: C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco (1993)
13. Skowron, A., Rauszer, C.: The discernibility matrices and function in information system. In: Slowinski, R. (ed.) *Intelligent Decision support Handbook of Application and Advances of the Rough sets Theory*, pp. 331–362. Kluwer Academic Publisher, Dordrecht (1991)
14. Yuan-Zhen, W., Xiao-Bing, P.: A Fast Algorithm for Reduction Based on Skowron Discernibility Matrix. *Compute Science (in China)* 32(4), 42–44 (2005)

Analyzing the Propagation of Influence and Concept Evolution in Enterprise Social Networks through Centrality and Latent Semantic Analysis

Weizhong Zhu, Chaomei Chen, and Robert B. Allen

College of Information Science and Technology, Drexel University
19014 Philadelphia, USA
{wz32, chaomei.chen, bob.allen}@ischool.drexel.edu

Abstract. Understanding the propagation of influence and the concept flow over a network in general has profound theoretical and practical implications. In this paper, we propose a novel approach to ranking individual members of a real-world communication network in terms of their roles in such propagation processes. We first improve the accuracy of the centrality measures by incorporating temporal attributes. Then, we integrate weighted PageRank and centrality scores to further improve the quality of these measures. We valid these ranking measures through a study of an email archive of a W3C working group against an independent list of experts. The results show that time-sensitive Degree, time-sensitive Betweenness and the integration of the weighted PageRank and these centrality measures yield the best ranking results. Our approach partially solves the rank sink problem of PageRank by adjusting flexible jumping probabilities with Betweenness centrality scores. Finally the text analysis based on Latent Semantic Indexing extracts key concepts distributed in different time frames and explores the evolution of the discussion topics in the social network. The overall study depicts an overview of the roles of the actors and conceptual evolution in the social network. These findings are important to understand the dynamics of the social networks.

Keywords: Betweenness Centrality, Weighted PageRank, Social Network Analysis, Time Series Analysis, Latent Semantic Indexing.

1 Introduction

Social Network analysis (SNA) investigates the interactions among people, organizations or communities. Two factors are essential for understanding the social status of an actor --- popularity and prestige. Popularity can be measured by the quantity of endorsements the actor receives from other actors, whereas the prestige is shown by the quality of the received endorsements, for example, the prestige of endorsing actors [2]. The quality of scholarly communication is often assessed in terms of the number of citations it has received. We extend this notion to the study of the influence of an individual in a network of email communication. Our study aims to address the extent one can identify the influential status of group members based on the structure of their email communications and explore the evolution of the key discussion topics over an

extended period of time. A common criticism of social network research is that the study of prestige has not directly addressed the dynamic information flow in such networks [10, 19]. In this paper, we develop a similarity measure between nodes, which incorporates time factor and simulates the speed and frequency of email conversations. This measure is particularly useful for discovering long-term active experts and contemporary experts.

The Degree and Closeness [17] centrality are generally accepted as indicators of influential status, and are based on the number of neighbors for a node in a network and the distances between nodes. However, they primarily indicate the popularity rather than prestige. A potential measure of prestige is Betweenness centrality [8] [9], which is based on the critical members in the shortest paths between any pair of nodes in a network. Another possible measure of prestige is the PageRank algorithm [6] [16], which computes the influence of a web page based on a combination of the number of hyperlinks that point to the page and the influence of the pages that the hyperlinks originate from. PageRank, restricted to random walks, is essentially a special case of eigenvector centrality. All the four measures, i.e. Degree Centrality, Closeness, Betweenness Centrality, and PageRank, assume that influence propagates via restricted paths. We evaluate all these measures of influence by a member network in W3C according to their historical email conversations. First, we compare the results of these measures and identify the inter-relationships between them. Then we integrate two of them to improve the performance because the correlation relationships among these measures are statistically significant. We show that PageRank enhanced by time-sensitive Betweenness improves influence ranking by solving the rank sink problem of PageRank. To evaluate the consequences of such a change on the assessment of influence, we use the mail collection of the W3C URI working group selected from the TREC Enterprise 2005. To show the trends of concept evolution, LSI-based concept ranking method [24] is applied to extract and rank the discussion topics over a 10-year period (1904-2004).

The rest of this paper is organized as follows: Section 2 reviews related work, Section 3 discusses how to construct the time sensitive centrality. Section 4 presents the weighted PageRank algorithm and how to propagate with Betweenness centrality. Section 5 describes LSI-based concept extraction and ranking. Section 6 presents the experimental results and discussions. Section 7 is the conclusions.

2 Related Work

Borgatti [3] [4] addresses the problem of discovering key players by explicitly measuring the contribution of a set of actors to the cohesion of a network with two analytical functions. White and Smyth [20] define the most important nodes in the network by considering the referral links like PageRank [16] and HITS [12]. In [7] a linear model is used to produce sub-graphs on the basis of electrical circuit formulae. Huberman [13] uses the same approach and exploits Kirchhoff's Laws to model a social network. Other approaches such as [14] use Betweenness to find crucial central nodes. Pujol [11] proposed a PageRank style ranking algorithm that uses the out-degree that could be thought as a slight variant of absolute out-degree centrality to weigh the random jumping probability. In this paper, we extend weighted Page Rank

algorithm to social network analysis which follows the same traversing mechanism. Our novel method propagates weighted Page Rank with Betweenness to solve the “tank sink” problem of random walks.

3 Time Series Analysis

The simulation study in [10] reported that the centrality of nodes is affected by the characteristics of dynamics of information flows. Motivated by this study, we developed the time sensitive centrality measures. We divide a long period of time into a number of consecutive time slices. Participating actors in a social network are presented as the vectors of the email conversation frequencies in a time series. The groups of actors are clustered based on the similarity between their vectors. A linkage is defined by a send-reply chain between two actors in a time slice such as a month. The linkage is weighted by the cosine similarity between the vectors. With a chosen threshold, the graph for the network is generated and analyzed with small-world network model in Pajek [1]. Centrality scores are obtained through Degree/Closeness/Betweenness analysis in Pajek. The Betweenness analysis is an implementation of Brandes’s algorithm [5].

The original centrality measures assume the impacts of the conversations are equally important over time, which may not consider an expert who contributed to the enterprise community during an early decade. If we are supposed to find an “emerging expert”, this measure may not also be accurate. So we develop another measure by assigning each conversation a delaying weight depending on its age. The modified conversation frequency is divided by $(T(\text{current}) - T(i) + 1)^\delta$. If δ is set to 1, the conversation frequency is divided by the age. The measure particularly favors a recently active member in a community.

4 Propagation of Weighted PageRank with Betweenness

The definition of original PageRank assumes that prestige is equally distributed across all the links of a web page. In a social network, however, not all edges are equal. Some actors interact more often and/or more profoundly with others. In this context, the PageRank equation should take into account weighted communication links and to what extent they should transfer PageRank values. In our weighted PageRank equation, a propagation proportion is defined as $w(a_j, a_i)$ between actors a_i and a_j by normalizing the link weights emanating from a particular actor a_j as follows:

$$w(a_j, a_i) = W(a_j, a_i) / \sum_k W(a_j, a_k) \quad (1)$$

For any particular actor a_j , $w(a_j, a_i)$ is defined as the ratio between the number of email conversations between a_i and a_j to the number of all the email conversations of a_j . Therefore it can be used to determine the fraction of an actor’s PageRank that transfers to other linked actors.

The PageRank algorithm computes the importance scores of web pages through a stochastic irreducible Markov transition matrix, yielding the “rank sink” problem. To solve this problem, [6] introduces a uniform matrix and linearly interpolates it with

the normalized adjacency matrix with a fixed random jumping probability β . A surfer would be better off to follow the out-links of a high-quality hub page rather than a low-quality one. This motivates us to think that a dynamic β value based on the communication properties of an actor can be a better choice in SNA. Interestingly, we observe that Betweenness centrality $B(i)$ can be defined as the average probabilities across all possible pairs of nodes that the shortest path between any two nodes will pass through the given node i [8]. The Betweenness score could be seen as the average probability that any other node goes through the selected node. Driven by this definition, we hypothesize that the score of Betweenness could dynamically be used as the value of the parameter $(1-\beta)$, and model a PageRank Markov matrix more accurately. This assignment assumes any pair of nodes in the network communicates through shortest paths. We use this approach to extend our weighted PageRank to rank the actors in a social network. Then the Weighted PageRank Eq. 2 for an actor a_i is defined as follows:

$$PR_w(a_i) = (1 - \lambda) / N + \lambda \sum_j PR_w(a_j) * w(a_j, a_i) \quad (2)$$

According to Eq. 1, the transfer of prestige from one actor to the other is modulated by the propagation proportion $w(a_j, a_i)$. The parameter λ , which equals $B(a_i)$, represents the attenuation of prestige values as they are transferred from one actor to the other.

5 Concept Extraction and Ranking

Our natural idea is to represent each actor in the social network with a document which contains all the emails the actor had sent in a time frame and then to extract and rank concepts from these documents according their global statistical contribution. Firstly, 4257 emails of the W3C URI working group that belong to 388 members are divided into 388 separate documents. Each document is chopped into 11 pieces due to the years. Next, STANFORD part-of-speech (POS) tagging, stop-word filtering with the Google stop word list and Port stemming are applied to the corpus. A total of 8,647 single noun terms are selected for the subsequent text analysis. All the nouns with a less than 2 term frequency are excluded. The associative relationship between a noun term and a document is weighted by traditional TFIDF. These concepts are ranked by the feature selection algorithm in [24]. The ranking algorithm has been applied to news articles and the ISI citation abstracts related to SDSS (Sloan Digital Sky Survey). For instance, the algorithm extracts the top 5 concepts from the 61 ISI records of Dr. Michael Vogeley, like “void galaxy”, “power spectrum”, “genu curve”, “largescale” and “release”. He verifies that these concepts are good summaries for his research.

6 Experiments Results and Discussion

The W3C email collection used for the experiment was crawled from w3c.org. We automatically identified threads (in-reply-to chains) from the in-reply-to fields of

the email messages. There are three types of information, unique message ID, non-trivial subject lines, and null (not a reply). By linking messages with unique message ID and nontrivial subject lines, the pairs of senders and receivers are treated as discussion threads/links. This yielded 3032 discussion threads/links among 330 members from 4257 emails of the URI working group at W3C across 10-year period (1994-2004).

Table 1. The top 10 actors ranked by the 10 ranking algorithms

	BW	CL	DE	T_CL	T_DE
1	Larry Masinter*	Larry Masinter*	Larry Masinter*	Larry Masinter*	Larry Masinter*
2	Roy T. Fielding*	Roy T. Fielding*	Roy T. Fielding*	Roy T. Fielding*	Roy T. Fielding*
3	Michael Mealling*	Martin Duerst*	Martin Duerst*	Dan Connolly*	Dan Connolly*
4	Martin Duerst*	Dan Connolly*	Michael Mealling*	Michael Mealling*	Michael Mealling*
5	Dan Connolly*	Michael Mealling*	Dan Connolly*	Martin Duerst *	Martin Duerst *
6	Paul Hoffman	Al Gilman	Al Gilman	Al Gilman	Al Gilman
7	Al Gilman	Graham Klyne	Patrick Stickler	Paul Hoffman	Paul Hoffman
8	Patrick Stickler	Paul Hoffman	Graham Klyne	Daniel LaLiberte	Harald Tveit Alvestrand
9	Daniel LaLiberte	Daniel LaLiberte	Paul Hoffman	Graham Klyne	Daniel LaLiberte
10	Aaron Swartz	Patrick Stickler	Daniel LaLiberte	Ronald E. Daniel	Leslie Daigle*
	W_PR	PR_BW	T_BW	PR_TBW	TE_BW
1	Larry Masinter*	Larry Masinter *	Larry Masinter*	Larry Masinter*	Roy T. Fielding*
2	Roy T. Fielding*	Roy T. Fielding*	Roy T. Fielding*	Roy T. Fielding*	Michael Mealling*
3	Martin Duerst *	Michael Mealling*	Dan Connolly*	Michael Mealling*	Patrick Stickler
4	Michael Mealling*	Martin Duerst *	Michael Mealling*	Martin Duerst*	Al Gilman
5	Al Gilman	Paul Hoffman	Martin Duerst*	Dan Connolly*	Martin Duerst *
6	Patrick Stickler	Al Gilman	Al Gilman	Al Gilman	Chris Lilley
7	Dan Connolly*	Dan Connolly*	Harald Tveit Alvestrand	Paul Hoffman	Graham Klyne
8	Graham Klyne	Patrick Stickler	Paul Hoffman	Daniel LaLiberte	Daniel LaLiberte
9	Daniel LaLiberte	Daniel LaLiberte	Daniel LaLiberte	Harald Tveit Alvestrand	Larry Masinter*
10	Paul Hoffman	Aaron Swartz	Leslie Daigle*	Leslie Daigle*	John Cowan

*indicates the most influential members.

According to the evaluation of expert search task in TREC Enterprise 2005 competition, we select Dan Connolly, Michael Mealling, and Leslie Daigle as URI experts in addition to well-known members, Tim Beners-Lee, Larry Minster (URI chair), Roy T. Fielding and Martin Duerst. In table 1, BW denotes Betweenness. CL denotes Closeness. DE denotes output Degree Centrality. T_BW denotes time-sensitive Betweenness. T_CL denotes time-sensitive Closeness. T_DE denotes time-sensitive Degree Centrality. TE_BW denotes Betweenness centrality with the delaying weights on time. W_PR denotes weighted PageRank with a fixed parameter $\lambda=0.85$. PR_BW denotes weighted PageRank with a dynamic parameter λ generated from BW. PR_TBW

denotes weighted PageRank with a dynamic parameter λ generated from T_BW. The top 10 ranking lists of BW, CL, DE, W_PR, PR_BW and T_CL include 5 influential members. T_BW, T_DE and PR_TBW identify 6 influential members, including an additional influential member, Leslie Daigle. TE_BW identify four more currently active experts, which excludes Leslie Daigle and Dan Connolly. It suggested these two experts might be more active in the early developing stage of this working group. Tim Beners-Lee doesn't appear in the top ranking list because his conversation frequency is only ranked as 25th in our dataset. With BW, his ranking is 25th. With W_PR, his ranking drops to 27th. With PR_BW, his ranking is enhanced to the 22nd. If W_PR is extended by T_BW, his ranking is 18th. But if measured by T_BW, his best ranking, 15th, is achieved.

Table 2. The Spearman Correlations among the 9 ranking algorithms except TE_BW

	BW	CL	DE	T_BW	T_CL	T_DE	W_PR	PR_BW
CL	.65*	----	----	----	----	----	----	----
DE	.85*	.78*	----	----	----	----	----	----
T_BW	.73*	.73*	.81*	----	----	----	----	----
T_CL	.65*	.57*	.71*	.81*	----	----	----	----
T_DE	.64*	.55*	.70*	.80*	.99*	----	----	----
W_PR	.83*	.65*	.88*	.76*	.65*	.64*	----	----
PR_BW	.22*	.17*	.16*	.22*	.57*	.55*	.18*	----
PR_TBW	.17*	.11*	.12*	.02	.71*	.70*	.16*	.48*

*indicates statistically significant.

In table 2, the results indicate that most of the Spearman correlations between the 9 algorithms tend to be statistically significant. Even the correlation scores are changed from 0.11 to 0.99. Correlation coefficients among BW, CL, DE, T_BW, T_CL, T_DE and W_PR are larger than 0.5 from 0.64 to 0.99. Most of the coefficients among PR_BW, PR_TBW and other algorithms are less than 0.5 from 0.11 to 0.48 except T_CL and T_DE. The results indicate the integration of PageRank and Betweenness dramatically changes the topology of the graph. Through a node with a higher Betweenness score, information flows more likely follow the shortest pathways that are linked to the most influential actors.

Topical Terms in table 3 are ranked by the algorithm at section 5. For each year, the top ten themes are listed. These terms cover the key themes for URI, for instance, “urn, iri, character, base uri, fragment, ipv, resource” (see <http://gbiv.com/protocols/uri/rfc/rfc3986.html>). The rankings include one for the overall 10 year period and eleven for each separate year. Because of TFIDF weights are used in text processing, these noun terms with an IDF = 0 are excluded in the ranking list. For instance, the ranking list for 1994-2004 excludes a list of terms, “uri, url, name, scheme, and example”. Obviously, these terms are very important and should be considered to understand the concept evolution. The highlighted terms reflect the concept building history in the URI working group in the ten-year period. The concepts shift from “url, uri” to “urn” and then “iuri, iri”, which matches the concept development history of URI (see, <http://www.w3.org/Addressing/>).

Table 3. Discussion Topic Evolution in the URI working group from 1994 Dec to 2004 May

Rank	1994-2004	1994	1995	1996	1997	1998
1	fragment	some	urn	urn	rtsp	academy
2	urn	body	rate	vemmi	utf	nntp
3	character	davenport	cid	irc	character	lb
4	lid	ics	Initiative	wnetc	div	encode
5	rate	harald	range	fragment	chri	script
6	utf	norwegian	digest	mud	imap	gaymen
7	base	usenet	ipv	draft	base	utf
8	turi	alvestrand	finger	deployment	susan	axiom
9	iri	cmu	docid	local	numer	cesused
10	vemmi	usage	lyco	acct	fragment	networkd
Actor amount	388	11	137	54	70	37
Rank	1999	2000	2001	2002	2003	2004
1	urn	lid	null	lm	urn	snmp
2	admin	utf	webdav	smb	mm	file
3	error	xml	ark	base	fragment	dollar
4	palceum	base	dav	query	openurl	namespace
5	busy	Reagl	protozilla	rdf	catalog	iri
6	termin	sysrcus	tftp	offer	tld	sm
7	nature	idn	index	iri	ni	associative
8	product	urn	christian	oai	thing	fragment
9	paper	entity	iri	identity	pgp	resource
10	leslie	gerald	urn	yahoo	dan	info
Actor amount	19	39	78	67	97	52

7 Conclusion

In summary, there is no substantial difference among the six centrality measures, BW, CL, DE, T_CL and W_PR. Weighted PageRank integrated with time-sensitive Betweenness (PR_TBW), time-sensitive Betweenness (T_BW), and time-sensitive (T_DE) perform 60% (6 out of the top 10 ranks) accuracy. They appear to be the best measures to identify influential members from email conversations compared to any other algorithm. Even though TE_BW identify 4 out of the top 10, it emphasizes on the discovery of the contemporarily active experts. So including the time attribute improves the centrality measures. Betweenness Centrality is validated to be a good estimator of random jumping probabilities in a social network and it partly solved the rank-sink problem of PageRank. Furthermore, our novel approach integrates content analysis to bootstrap key concepts distributed in the social network. The top-ranking concepts selected from the different years demonstrate the development history of the discussion topics in the enterprise working group. The overall study highlights the roles of the most influential actors and demonstrates the evolution of the conceptual temporal structures in the social networks.

Our current analysis emphasizes on the study of the strong ties among conversation links with restrict paths. However, there are many unrestricted pathways in the communication system such as the broadcast emails. In future studies, we will explore and study the approaches from information diffusion perspectives like graph theory [15] [18], eigenvector centrality and information centrality [21] [22] [23] on the social or biological communities with unrestricted topology properties.

Acknowledgments. The work is in part supported by the National Visualization and Analytics Center (NVAC) through the Northeast Visualization and Analytics Center (NEVAC).

References

1. Batagelj, V., Mrvar, A.: Pajek - Analysis and Visualization of Large Networks. In: Jünger, M., Mutzel, P. (eds.) *Graph Drawing Software*, Springer, Berlin (2003)
2. Bollen, J., Rodriguez, M.A., Van de Sompel, H.: Journal status. ArXiv, January 9 (2006)
3. Bonacich, P.: Factoring and weighting approaches to status scores and clique identification. *Journal of Mathematical Sociology* 2, 113–120 (1972)
4. Borgatti, S.P.: Centrality and network flow. *Social Networks* 27, 55–77 (2005)
5. Brandes, U.: A Faster Algorithm for Betweenness Centrality. *Journal of Mathematical Sociology* 25(2), 163–177 (2001)
6. Brin, S., Page, L.: The anatomy of a large-scale hypertextual web search engine. *Computer Networks* 30(1-7), 107–117 (1998)
7. Faloutsos, C., McCurley, K., Tomkins, A.: Fast discovery of connection subgraphs. In: *ACM SIGKDD*, pp. 118–127 (2004)
8. Freeman, L.C.: A set of measures of centrality based on Betweenness. *Sociometry* 40, 35–41 (1997)
9. Freeman, L.C.: Centrality in social networks: Conceptual clarification. *Social Networks* 1, 215–239 (1979)
10. Friedkin, N.E.: Theoretical foundations for centrality measures. *American Journal of Sociology*, 1478–1504 (1991)
11. Pujol, J.M., Sangüesa, R., Delgado, J.: Extracting reputation in multi agent systems by means of social network topology. In: *Proceedings of the first international joint conference on Autonomous agents and multi-agent systems*, pp. 467–474 (2002)
12. Kleinberg, J.: Authoritative sources in a hyperlinked environment. *Journal of ACM* 46, 604–632 (1999)
13. Huberman, B., Wu, F.: Finding communities in linear time: a physics approach. *The European Physical Journal B - Condensed Matter* 38(2), 331–338 (2004)
14. Newman, M.: Who is the best connected scientist? A study of scientific co-authorship networks. In: Ben-Naim, E., Frauenfelder, H., Toroczkai, Z. (eds.) *Complex Networks*, pp. 337–370. Springer, Heidelberg (2004)
15. Nobel, C., Cook, D.J.: Graph-based anomaly detection. In: *ACM SIGKDD*, pp. 631–636 (2003)
16. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the Web. Technical report, Stanford Digital Library Technologies Project (1998)
17. Sabidussi, G.: The centrality index of a graph. *Psychometrika* 31, 581–603 (1966)
18. Shetty, J., Adibi, J.: Discovering important nodes through graph entropy: The case of Enron email database. In: *ACM SIGKDD* (2005)
19. Watts, D.: *Six degrees: The science of a connected age*. Norton, New York (2003)
20. White, S., Smyth, P.: Algorithms for estimating relative importance in networks. In: *ACM SIGKDD*, pp. 266–275 (2003)
21. Freeman, L.C., Borgatti, S.P., White, D.R.: Centrality in valued graphs: A measure of Betweenness based on network flow. *Social Networks* 13, 141–154 (1991)

22. Latora, V., Marchiori, M.: cond-mat/0402050
23. Newman, M.E.J.: A measure of Betweenness centrality based on random walks, arXiv:cond-mat/0309045 v1 (2003)
24. Zhu, W., Chen, C.: Storylines: Visual exploration and analysis in latent semantic spaces. *International Journal of Computers and Graphics. Special Issue on Visual Analytics* 31(3), 338–349 (2007)

Author Index

- Achuthan, N.R. 554
Ahmed, Chowdhury Farhan 1022
Aizawa, Akiko 76
Allen, Robert B. 1090
Anderson, Grant 494
Aoki-Kinoshita, Kiyoko F. 184
Archambeau, Cédric 527
Arimura, Hiroki 2, 234, 357
Assent, Ira 40
- Bai, Shuo 877
Banerjee, Arindam 896
Bao, Shenghua 393
Beard, Mitchell 1015
Berthold, Michael R. 14
Bian, Fuling 1042
Blansch e, Alexandre 849
Bouguila, Nizar 503
Boulcaut, Jean-Fran ois 112
Brajczuk, Dale A. 653
Bringmann, Bj orn 858
- Cao, Longbing 1069
Carmichael, Christopher L. 644
Carrasco-Ochoa, J. Ariel 697
Chang, Jason 626
Chang, Lei 405
Chawla, Nitesh 519
Chen, Chaomei 1090
Chen, Chih-Jen 1035
Chen, Chun-Hao 864
Chen, Guoqing 777
Chen, Jingnian 870
Chen, Kuei-Hsien 785
Chen, Lien-Chin 759
Chen, Ming-Syan 53, 626
Chen, Ying-Ju 53
Cheng, Hong 970
Cheng, Xueqi 877
Cheung, David W. 381
Chiu, Chuang-Cheng 749
Cho, Sungzoon 608
Christen, Peter 511, 536
Chu, Yi-Hong 53
Chui, Chun-Kit 64
- Churilov, Leonid 715
Cieslak, David 519
Cordero Hernandez, Jorge 441
- Dai, Qionghai 830
Dang, Van B. 76
Delannay, Nicolas 527
Denny 536
Desmarais, Michel C. 562
Dill, Fabian 14
Ding, Wei 88
Do, Thanh-Nghi 634
Drummond, Chris 26
D zeroski, Sašo 284, 454
- Eguchi, Koji 705
Eick, Christoph F. 88
ElGuebaly, Walid 503
Elomaa, Tapio 544
Endo, Tsutomu 1006
Erwin, Alva 554
Euachongprasit, Waiyawuth 100
- Faloutsos, Christos 1
Famili, A. Fazel 196
Fan, Wei 970
Fu, Huirong 479
Fu, Shunkai 562
Fu, Yongjian 479
- Gay, Dominique 112
Gong, Caichun 877
Gong, Xueqing 160
Gopal, Rupesh K. 884
Gopalan, Brian 1015
Gopalan, Raj P. 554
Guo, Ling 124
Guo, Songtao 124
- Hacid, Hakim 985
Han, Dingyi 393
Han, Shuguo 136
Hassan, Md. Rafiul 572
He, Qing 222
Herbers, J rg 40

- Hernández-Rodríguez, Selene 697
 Hido, Shohei 148
 Hirata, Kouichi 184, 600
 Holmes, Geoffrey 296
 Holte, Robert C. 26
 Hong, Tzung-Pei 864
 Hossain, M. Maruf 572
 Hsieh, Nan-Chen 890
 Hsu, Kuo-Wei 896
 Hsu, Ming-Li 999
 Hu, Bao-Gang 417, 813, 1049
 Hu, Sanqing 830
 Hu, Tianming 160
 Huang, Houkuan 870
 Huang, Jih-Hong 1035
 Huang, Yulan 877
 Hung, Edward 381
 Hwang, Joengmin 1056

 Idé, Tsuyoshi 148, 333
 Inuzuka, Nobuhiro 582
 Irani, Pourang P. 644
 Iwata, Shuichi 849

 Jaroszewicz, Szymon 172
 Jeong, Byeong-Soo 1022
 Jiamthaphaksin, Rachsuda 88
 Jiang, Dan 88
 Jiang, He 839
 Jiang, Yi 1081
 Johansson, Ulf 592

 Kangkachit, Thanapat 767
 Kao, Ben 64
 Kao, Ching-Pin 759
 Kashima, Hisashi 148
 Katoh, Takashi 600
 Kim, Dongil 608
 Kirkby, Richard 296
 Kirley, Michael 572
 Kitagawa, Genshiro 30
 Kitsuregawa, Masaru 923
 Kiyasu, Senya 964
 Kobayashi, Mei 616
 Kobayashi, Shigenobu 320
 Kobyliński, Lukasz 904
 Koh, Yun Sing 910, 916
 Kötter, Tobias 14
 Krieger, Ralph 40

 Kubo, Harunobu 148
 Kuboyama, Tetsuji 184

 Lallich, Stéphane 634
 Lazarescu, Mihai 662
 Leckie, Christopher 369
 Lee, Kuo-Chen 626
 Lee, Young-Koo 1022
 Lehtinen, Petri 544
 Lenca, Philippe 634
 Létourneau, Sylvain 196
 Leung, Carson Kai-Sang 644, 653
 Li, Jintao 803
 Li, Jiye 673
 Li, Juanzi 466, 821
 Li, Lin 923
 Li, Ming 209
 Li, Ming 992
 Li, Ming-Chu 839
 Li, Rui 393
 Liang, Chunyan 938
 Liewlom, Peera 767
 Lin, Xiaowei 932
 Liu, Duen-Ren 999
 Liu, Huan 381
 Liu, Jicheng 938
 Liu, Liu 466
 Liu, Qiuge 222
 Lo, Chia-Hao 945
 Löfström, Tuve 592
 Long, Jun 951
 Lühr, Sebastian 662
 Lukose, Rajan 673

 Mark, Leo 247
 Martínez-Trinidad, J.Fco. 697
 Marukatat, Sanparith 958
 Masada, Tomonari 964
 Masuyama, Shigeru 977
 Mateo, Mark Anthony F. 653
 Matsuzawa, Hirofumi 148
 Matwin, Stan 196
 Meher, Saroj K. 884
 Minato, Shin-ichi 234
 Miyahara, Sueharu 964
 Motoyama, Jun-ichi 582

 Nakagawa, Hiroshi 260
 Nakajima, Shinichi 333
 Nakano, Tomofumi 582
 Nelson, Gregory 1015

- Ng, Wee Keong 136
 Nguyen, Minh Quoc 247
 Nijssen, Siegfried 858
 Niklasson, Lars 592
- Ohkawa, Takenao 272, 705
 Okumura, Manabu 731
 Omiecinski, Edward 247
 Ono, Shingo 260
 Ozaki, Tomonobu 272
- Park, Laurence A.F. 681
 Parmar, Rachana 88
 Patist, J.P. 689
 Pears, Russel 916
 Peckov, Aleksandar 284
 Peng, Wen-Chih 945
 Penmetsa, Satyanarayana Raju 673
 Pesonen, Antti 1028
 Pfahringer, Bernhard 296, 494
 Pham, Nguyen-Khang 634
- Qiu, Zhengyuan 970
- Rakthanmanon, Thanawin 767
 Ramamohanarao, Kotagiri 369, 681
 Ratanamahatana, Chotirat Ann 100
 Ren, Jiangtao 970
 Rendle, Steffen 308
- Sakai, Hiroyuki 977
 Sakaji, Hiroki 977
 Sakuma, Jun 320
 Sato, Issei 260
 Sayed, Ahmad El 985
 Schmidt-Thieme, Lars 308
 Seidl, Thomas 40
 Selmaoui, Nazha 112
 Sese, Jun 333
 Shen, Wei 830
 Shen, Yi-Dong 992
 Shen, Zhi-Yong 992
 Shi, Baile 795
 Shi, Zhongzhi 222
 Shih, Meng-Jung 999
 Shimada, Kazutaka 1006
 Shiozaki, Hitohiro 705
 Siew, Eu-Gene 715
 Singh, Lisa 1015
 Smith-Miles, Kate A. 715
- Song, Yixu 932
 Sriphaew, Kritsada 731
 Srivastava, Jaideep 896
 Stepinski, Tomasz F. 88
 Sturmberg, Joachim P. 715
 Su, Chwen-Tzeng 785
 Su, Ja-Hwung 1035
 Sugiyama, Masashi 333
 Sun, Jun 992
 Sung, Sam Yuan 160
- Tanbeer, Syed Khairuzzaman 1022
 Tang, Jie 466
 Tang, Sheng 803
 Terlecki, Pawel 723
 Theeramunkong, Thanaruk 731
 Thiel, Kilian 14
 Tian, Fengzhan 870
 Tian, Shengfeng 870
 Timonen, Mika 1028
 Todorovski, Ljupčo 284
 Tsai, Cheng-Fa 739
 Tsai, Chieh-Yuan 749
 Tseng, Vincent S. 759, 864, 1035
- Uno, Takeaki 234, 345, 357
 Urazawa, Shinpei 582
- Verleysen, Michel 527
- Waiyamai, Kitsana 767
 Walczak, Krzysztof 723, 904
 Wan, You 1042
 Wang, Hao 777
 Wang, Jianyong 1062
 Wang, KeHong 821
 Wang, Liang 369
 Wang, Peng 795
 Wang, Tengjiao 405
 Wang, Wei 795
 Wang, Xiaozhe 369
 Welter, Petra 40
 Williams, Graham J. 536
 Wong, Jui-Tsung 785
 Wong, W.K. 381
 Wu, Gang 821
 Wu, Xian 1075
 Wu, Xiaochen 795
 Wu, Xintao 124

- Xia, Tian 803
 Xiong, Hui 160
 Xu, Kaifeng 393

 Yang, Bishan 405
 Yang, Dongqing 405
 Yang, Shuang-Hong 417, 813, 1049
 Yang, Yu-Jiu 813, 1049
 Yang, Zehong 932
 Yang, Zhenglu 923
 Yen, Chia-Chen 739
 Yin, Jianping 951
 Yoo, Jin Soung 1056
 Yoshida, Minoru 260
 Yu, Philip S. 970
 Yu, Yang 429
 Yu, Yong 393, 1075
 Yung, Raylene 616

 Zeng, Yifeng 441
 Zeng, Zhiping 1062
 Ženko, Bernard 454

 Zhang, Chengqi 1069
 Zhang, DeFu 1081
 Zhang, Huaifeng 1069
 Zhang, Jing 466
 Zhang, Kuo 821
 Zhang, Xian-Chao 839
 Zhang, Xing 777
 Zhang, Yongdong 803
 Zhang, Zhongfei (Mark) 209
 Zhao, Wentao 951
 Zhao, Yanchang 1069
 Zheng, Hao 1075
 Zheng, Huicheng 830
 Zhou, Jing 673
 Zhou, Lizhu 1062
 Zhou, XiYue 1081
 Zhou, Zhi-Hua 209, 429
 Zhu, En 951
 Zhu, Weizhong 1090
 Zhu, Ye 479
 Zighed, Djamel 985
 Zong, Yu 839