Xue Li
Shuliang Wang
Zhao Yang Dong (Eds.)

LNAI 3584

Advanced Data Mining and Applications

First International Conference, ADMA 2005
Wuhan, China, July 2005
Proceedings

Springer

Lecture Notes in Artificial Intelligence 3584

Edited by J. G. Carbonell and J. Siekmann

Subseries of Lecture Notes in Computer Science

Springer
Berlin
Heidelberg
New York
Hong Kong
London
Milan
Paris
Tokyo

Xue Li Shuliang Wang
Zhao Yang Dong (Eds.)

Advanced Data Mining and Applications

First International Conference, ADMA 2005
Wuhan, China, July 22-24, 2005
Proceedings

Springer

Volume Editors

Xue Li
University of Queensland
School of Information Technology and Electrical Engineering
Brisbane 4072, Queensland, Australia
E-mail: xueli@itee.uq.edu.au

Shuliang Wang
Wuhan University
International School of Software
Wuhan 430072, China
E-mail: slwang2005@whu.edu.cn

Zhao Yang Dong
University of Queensland
School of Information Technology and Electrical Engineering
St. Lucia, Queensland 4072, Australia
E-mail: zdong@itee.uq.edu.au

Library of Congress Control Number: 2005929058

CR Subject Classification (1998): I.2, H.2.8, H.3-4, K.4.4, J.3, I.4, J.1

ISSN	0302-9743
ISBN-10	3-540-27894-X Springer Berlin Heidelberg New York
ISBN-13	978-3-540-27894-8 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media

springeronline.com

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India
Printed on acid-free paper SPIN: 11527503 06/3142 5 4 3 2 1 0

Preface

With the ever-growing power to generate, transmit and collect huge amounts of data, information overload is now an imminent problem to mankind. The overwhelming demand for information processing is not just about a better understanding of data, but also a better usage of data in a timely fashion. Data mining, or knowledge discovery from databases, is proposed to gain insight into aspects of data and to help people make informed, sensible, and better decisions. At present, growing attention has been paid to the study, development and application of data mining. As a result there is an urgent need for sophisticated techniques and tools that can handle new fields of data mining, e.g., spatial data mining, biomedical data mining, and mining on high-speed and time-variant data streams. The knowledge of data mining should also be expanded to new applications.

The 1st International Conference on Advanced Data Mining and Applications (ADMA 2005) aimed to bring together the experts on data mining throughout the world. It provided a leading international forum for the dissemination of original research results in advanced data mining techniques, applications, algorithms, software and systems, and different applied disciplines. The conference attracted 539 online submissions and 63 mailing submissions from 25 different countries and areas. All full papers were peer reviewed by at least three members of the Program Committee composed of international experts in data mining fields. A total number of 100 papers were accepted for the conference. Amongst them 25 papers were selected as regular papers and 75 papers were selected as short papers, yielding a combined acceptance rate of 17%.

The ADMA 2005 program highlights were four keynote speeches from outstanding researchers in advanced data mining and application areas: David Olson, Deyi Li, Chenqi Zhang, and Osmar Zaïane. The conference also invited researchers from two Australian universities to report on their latest research findings.

May 2005

Xue Li,
Shuliang Wang,
Zhaoyang Dong

Conference Committee

ADMA 2005 was organized by the International School of Software, Wuhan University, China and the School of Information Technology and Electrical Engineering, the University of Queensland, Australia; sponsored by the State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing, Wuhan University, China and the WISE (Web Information Systems Engineering, [illegible]) Society; and technically co-sponsored by the IEEE Queensland Section.

Organizing Committee

Honorary Chair	Lotfi A. Zadeh (Berkeley University of California, USA) Jingnan Liu (Wuhan University, China)
Conference Co-chairs	Deren Li (Wuhan University, China) Xiaofang Zhou (University of Queensland, Australia)
Program Co-chairs	Xue Li (University of Queensland, Australia) Shuliang Wang (Secretary-General, Wuhan University, China) Zhaoyang Dong (University of Queensland, Australia)
Organizing Chairs	Yixin Zeng and Min Chen (Wuhan University, China)
Publicity Chair	Yanchun Zhang (Victoria University, Australia)

Local Advisory Committee

Liu Jingnan, Chen Shaofang	Wuhan University, China
Li Wenxing, Huang Jin	Wuhan University, China
Zhou Chuangbing	Wuhan University, China
Liu Haixin	Technology Bureau of Hubei Province, China

Local Organizing Committee

Liu Fang, Yang Jing, Zhou Xiaoming, Lin Bin, Zhu Guobing, Zheng Jing, Sun Ling, Li Li	Wuhan University, China

Program Committee Members

Jesus Aguilar, Spain
Viorel Ariton, Romania
Michael Bain, Australia
Jose Luis Balcazar, Spain
Elena Baralis, Italy
Petr Berka, Czech Republic
Michael R. Berthold, Germany
Fernando Berzal, Spain
Fuling Bian, China
Francesco Bonchi, Italy
Jean-Francois Boulicaut, France
Rui Camacho, Portugal
Guoqing Chen, China
Min Chen, China
Krzysztof Cios, USA
Bruno Cremilleux, France
Luc Dehaspe, Belgium
Kaichang Di, USA
Floriana Esposito, Italy
Marcus Gallagher, Australia
Joao Gama, Portugal
Dragan Gamberger, Croatia
Jean-Gabriel Ganascia, France
Junbin Gao, Australia
Christophe Giraud-Carrier, USA
Bart Goethals, Belgium
Michael Frank Goodchild, USA
Vladimir Gorodetsky, Russia
Jiawei Han, USA
Keqing He, China
Yi Hong, Australia
Andreas Hotho, Germany
Zhanyi Hu, China
Alípio Jorge, Portugal
Mehmed Kantardzic, USA
Eamonn Keogh, USA
Adam Krzyzak, Canada
Andrew Kusiak, USA
Longin Jan Latecki, USA
Andre Ponce Leao, Brazil
Deyi Li, China
Qiaoyun Li, USA
Qing Li, Hong Kong, China
Hui Lin, Hong Kong, China
Xuemin Lin, Australia
Wanquan Liu, Australia
Yungang Liu, China
Giuseppe Manco, Italy
Rosa Meo, Italy
Dunja Mladenic, Slovenia
Iveta Mrazova, Czech Republic
Olfa Nasraoui, USA
Daniel Neagu, UK
Claire Nedellec, France
Mircea Neogita, New Zealand
Arlindo Oliveira, Portugal
David L. Olson, USA
Yonghong Peng, UK
Johann Petrak, Austria
Pearl Pu, Switzerland
Raghu Ramakrishnan, USA
Jan Rauch, Czech Republic
Zbigniew W. Ras, USA
Cesar Rego, USA
Christophe Rigotti, France
Joseph Roure, Spain
Juho Rousu, UK
Celine Rouveirol, France
Daniel Sanchez, Spain
Yucel Saygin, Turkey
Marc Sebban, France
Giovanni Semeraro, Italy
Seyed A. Shahrestani, Australia
Wenzhong Shi, Hong Kong, China
Andrzej Skowron, Poland
Robert H. Sloan, USA
Carlos Soares, Portugal
Olga Stepankova, Czech Republic
Ah-Hwee Tan, Singapore
Kay Chen Tan, Singapore
Kok Kiong Tan, Singapore
Arthur Tay, Singapore
Luis Torgo, Portugal
Shusaku Tsumoto, Japan
Brijesh Verma, Australia
Ricardo Vilalta, USA

Paul Vitanyi, The Netherlands
Dianhui Wang, Australia
Ke Wang, Canada
Wei Wang, USA
Xinzhou Wang, China
Xizhao Wang, China
Marco Wiering, The Netherlands
Janet Wiles, Australia
Raymond Hau-San Wong, Hong Kong, China
Dash Wu, Canada
Dongming Xu, Australia
Zijiang Yang, Canada
Jeffrey Xu Yu, Hong Kong, China
Philip S. Yu, USA
Osmar R. Zaïane, Canada
Gerson Zaverucha, Brazil
Sarah Zelikovitz, USA
Benjamin Zhan, USA
Shichao Zhang, Australia
Chenghu Zhou, China
Djamel A. Zighed, France
Blaz Zupan, Slovenia

External Reviewers

Mohsin Ali, Australia
Dingyi Chen, Australia
Xia Chen, China
Marian Craciun, Romania
Tomaz Curk, Slovenia
Yi Ding, Australia
Gongde Guo, UK
Zi Huang, Australia
Zheng Liu, Australia
Gregor Leban, Slovenia
Juggapong Natwichai, Australia
Daniel Neagu, UK
Son Nghu, Australia
Anisah Nizar, Australia
Christoph Schmitz, Germany
Dawei Song, Australia
Gerd Stumme, Germany
Xingzhi Sun, Australia
Yidong Yuan, Australia
Shuai Zhang, UK
Junhua Zhao, Australia

Table of Contents

Keynote Papers

Invited Papers

Association Rules

Classification

Clustering

Novel Algorithms

Text Mining

Multimedia Mining

Sequential Data Mining and Time Series Mining

Web Mining

Biomedical Mining

Advanced Applications

Security and Privacy Issues

Spatial Data Mining

Streaming Data Mining

Decision Making with Uncertainty and Data Mining

David L. Olson[1] and Desheng Wu[1,2]

[1] Department of Management, University of Nebraska,
Lincoln, NE 68588-0491
dolson3@unl.edu
[2] Depart of Mechanical and Industrial Engineering,
University of Toronto, 5 King's College Road,
Toronto, Ontario M5S 3G8
dash@mie.utoronto.ca

Abstract. Data mining is a newly developed and emerging area of computational intelligence that offers new theories, techniques, and tools for analysis of large data sets. It is expected to offer more and more support to modern organizations which face a serious challenge of how to make decision from massively increased information so that they can better understand their markets, customers, suppliers, operations and internal business processes. This paper discusses fuzzy decision-making using the Grey Related Analysis method. Fuzzy models are expected to better reflect decision maker uncertainty, at some cost in accuracy relative to crisp models. Monte Carlo Simulation, a data mining technique, is used to measure the impact of fuzzy models relative to crisp models. Fuzzy models were found to provide fits as good as crisp models in the data analyzed.

1 Introduction

Data mining is a newly developed and emerging area of computational intelligence that offers new theories, techniques, and tools for analysis of large data sets. This emerging development corresponds to the current needs of information intensive organizations transform themselves from passive collectors to active explorers and exploiters of data. Modern organizations face a serious challenge that is how they should make decisions from massively increased information so that they can better understand their markets, customers, suppliers, operations and internal business processes. The field of data mining aims to improve decision making by focusing on discovering valid, comprehensible, and potentially useful knowledge from large data sets.

This paper presents a brief demonstration of the use of Monte Carlo simulation in grey related analysis. Simulation provides a means to more completely describe expected results, to include identification of the probability of a particular option being best in a multiattribute setting. The next section describes a Monte Carlo simulation of results of decision tree analysis of real credit card data. Monte Carlo simulation provides a means to more completely assess relative performance of alternative decision tree models. Relative performance of crisp and fuzzy decision tree models is assessed in the conclusions.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 1–9, 2005.

2 Simulation of Grey Related Analysis

Multiattribute decision making with uncertainty has progressed in a variety of directions throughout the world, which greatly enrich the development of probability theory [8], fuzzy theory [4], rough sets[7], grey sets[3] and vague sets [9]. The basic multiattribute model can be written as follows:

$$value_j = \sum_{i=1}^{K} w_i \times u(x_{ij}) \tag{1}$$

where w_i is the weight of attribute *i*, *K* is the number of attributes, and $u(x_{ij})$ is the score of alternative x_j on attribute *i*. Real life decisions usually involve high levels of uncertainty which can be reflected in the development of multiattribute models. Decision making methods with uncertain input in the form of fuzzy sets and rough sets have been widely published in both multiattribute decision making [1] and in data mining [2, 5]. The method of grey analysis [3] is another approach reflecting uncertainty into the basic multiattribute model. This paper discusses the use of a common data analysis technique, i.e, Monte Carlo Simulation, to this model to reflect uncertainty as expressed by fuzzy inputs. While this example is on a small data set, it can be extended to large data sets in data mining contexts as well. Monte Carlo simulation thus provides an efficient way to analyze grey analysis data[10].

Grey related analysis is a decision making technique which can be used to deal with uncertainty in forms of fuzzy data. Suppose that a multiple attribute decision making problem with interval numbers has m feasible plans $X_1, X_2, ..., X_m$, n indexes, weight value w_j of index G_j is uncertain, but we know $w_j \in [c_j, d_j]$, $0 \le c_j \le d_j \le 1$, $j = 1,2,...,n$, and the index value of j-th index G_j of feasible plan X_i is an interval number $[a_{ij}^-, a_{ij}^+], i = 1,2,...,m$, $j = 1,2,...,n$. When $c_j = d_j$, $j = 1,2,...,n$, the multiple attribute decision making problem with interval numbers is called a multiple attribute decision making problem with interval-valued indexes; When $a_{ij}^- = a_{ij}^+$, $i = 1,2,...,m$, $j = 1,2,...,n$, the multiple attribute decision making problem with interval numbers is called a multiple attribute decision making problem with interval-valued weights. Now the principle and steps of the Grey Related Analysis method are demonstrated by the following case illustration.

Consider the following problem consisting of six applicants for a position, each evaluated over seven attributes. Attributes are Experience in the Business Area(C1), Experience in the Specific Job Function(C2), Educational Background(C3), Leadership Capacity(C4), Adaptability(C5), Age(C6), Aptitude for Teamwork(C7). Raw data is provided in the form of trapezoidal data, which can be converted to an interval value using α-cut technology to build a membership function [4]. In this case, using an α of 0.5, we obtain the data in Table 1:

Table 1. Interval Data

weights	[0.20 0.35]	[0.30 0.55]	[0.05 0.30]	[0.25 0.50]	[0.15 0.45]	[0.05 0.30]	[0.25 0.55]
Performance	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Antônio	[0.65 0.85]	[0.75 0.95]	[0.25 0.45]	[0.45 0.85]	[0.05 0.45]	[0.45 0.75]	[0.75 1.00]
Fábio	[0.25 0.45]	[0.05 0.25]	[0.65 0.85]	[0.30 0.65]	[0.30 0.75]	[0.05 0.25]	[0.05 0.45]
Alberto	[0.45 0.65]	[0.20 0.80]	[0.65 0.85]	[0.50 0.80]	[0.35 0.90]	[0.20 0.45]	[0.75 1.00]
Fernando	[0.85 1.00]	[0.35 0.75]	[0.65 0.85]	[0.15 0.65]	[0.30 0.70]	[0.45 0.80]	[0.35 0.70]
Isabel	[0.50 0.95]	[0.65 0.95]	[0.45 0.65]	[0.65 0.95]	[0.05 0.50]	[0.45 0.80]	[0.50 0.90]
Rafaela	[0.65 0.85]	[0.15 0.35]	[0.45 0.65]	[0.25 0.75]	[0.05 0.45]	[0.45 0.80]	[0.10 0.55]

All of these index values are positive. The next step of the grey related method is to standardize the interval decision matrix. This step is omitted since our data is already on a 0-1 range. Next we need to calculate the interval number weighted matrix C, which consists of the minimum weight times the minimum alternative performance score for each entry as the left element of the interval number, and the maximum weight times the maximum alternative performance score for each entry as the right element of that entry's interval number. The weighted matrix C is shown in Table 2.

Table 2. Weighted Matrix C

Perfomance	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Antônio	[0.13 0.30]	[0.22 0.52]	[0.01 0.14]	[0.11 0.43]	[0.01 0.20]	[0.02 0.23]	[0.18 0.55]
Fábio	[0.05 0.16]	[0.01 0.14]	[0.03 0.26]	[0.07 0.33]	[0.04 0.34]	[0.00 0.08]	[0.01 0.25]
Alberto	[0.09 0.23]	[0.06 0.44]	[0.03 0.26]	[0.12 0.40]	[0.05 0.41]	[0.01 0.14]	[0.18 0.55]
Fernando	[0.17 0.35]	[0.10 0.41]	[0.03 0.26]	[0.03 0.33]	[0.04 0.32]	[0.02 0.24]	[0.09 0.39]
Isabel	[0.10 0.33]	[0.19 0.52]	[0.02 0.20]	[0.16 0.48]	[0.01 0.23]	[0.02 0.24]	[0.12 0.50]
Rafaela	[0.13 0.30]	[0.04 0.19]	[0.02 0.20]	[0.06 0.38]	[0.01 0.20]	[0.02 0.24]	[0.02 0.30]

The next step of the grey related method is to obtain reference number sequences based on the optimal weighted interval number value for every alternative. This is defined as the interval number for each attribute defined as the maximum left interval value over all alternatives, and the maximum right interval value over all alternatives.

For **C1**, this would yield the interval number [0.17, 0.35]. This reflects the maximum weighted value obtained in the data set for attribute **C1**. Table 3 gives this vector, which reflects the range of value possibilities (entries are not rounded):

Table 3. Reference Number Vector

~	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Max(Min)	0.1700	0.2250	0.0325	0.1625	0.0525	0.0225	0.1875
Max(Max)	0.3500	0.5225	0.2550	0.4750	0.4050	0.2400	0.5500

Distances are defined as the maximum between each interval value and the extremes generated. Table 4 shows the calculated distances by alternative.

Table 4. Distances From Alternatives to Reference Number Vector

Distances	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Antônio	[.04, .0525]	[0, 0]	[.02, .12]	[.05, .05]	[.045, .2025]	[0, .015]	[0, 0]
Fábio	[.12, .1925]	[.21, .385]	[0, 0]	[.0875, .15]	[.0075, .0675]	[.02, .165]	[.175, .3025]
Alberto	[.08, .1225]	[.165, .0825]	[0, 0]	[.0375, .075]	[0, 0]	[.0125, .105]	[0, 0]
Frenando	[0, 0]	[.12, .11]	[0, 0]	[.125, .15]	[.0075, .09]	[0, 0]	[.1, .165]
Isabel	[.07, .0175]	[.03, 0]	[.01, .06]	[0, 0]	[.045, .18]	[0, 0]	[.0625, .055]
Rafaela	[.04, .0525]	[.18, .33]	[.01, .06]	[.1, .1]	[.045, .2025]	[0, 0]	[.1625,.2475]

The maximum distance for each alternative to the ideal is identified as the largest distance calculation in each cell of Table 4. These maxima are shown in Table 5.

Table 5. Maximum Distances

Distances	**C1**	**C2**	**C3**	**C4**	**C5**	**C6**	**C7**
Antônio	0.0525	0	0.12	0.05	0.2025	0.015	0
Fábio	0.1925	0.385	0	0.15	0.0675	0.165	0.3025
Alberto	0.1225	0.165	0	0.075	0	0.105	0
Fernando	0	0.12	0	0.15	0.09	0	0.165
Isabel	0.07	0.03	0.06	0	0.18	0	0.0625
Rafaela	0.0525	0.33	0.06	0.1	0.2025	0	0.2475

A reference point is established as the maximum of entries in each column of Table 7. This point has a minimum of 0 and a maximum of 0.3850. Thus the reference point is [0, 0.385]. Next the method calculates the maximum distance between the reference point and each of the Weighted Matrix C values. Based upon weight interval number standardizing index value of every plan and reference number sequence.

Results by alternative are given in Table 6:

Table 6. Weighted Distances to Reference Point

Distances	1	2	3	4	5	6	7	Averages
Antônio	.785714	1	.616000	.793814	.487342	.927711	1	**.801512**
Fábio	.500000	.333333	1	.562044	.740385	.538462	.388889	**.580445**
Alberto	.611111	.538462	1	.719626	1	647059	1	**0.788037**
Frenando	1	.616000	1	.562044	.681416	1	.538462	**.771132**
Isabel	.733333	.865169	.762376	1	.516779	1	.754902	**.804651**
Rafaela	0.785714	.368421	0.762376	0.658120	0.487342	1	.437500	**.642782**

The average of these weighted distances is used as the reference number to order alternatives. These averages reflect how far away each alternative is from the nadir, along with how close they are to the ideal, much as in TOPSIS. This set of numbers indicates that Isabel is the preferred alternative, although Antônio is extremely close, with Alberto and Fernando close behind. This closeness demonstrates that the fuzzy input may reflect a case where there is not a clear winner. Simulation provides a tool capable of picking up the probability of each alternative being preferred.

3 Monte Carlo Simulation

To deal with both fuzzy weights and fuzzy alternative performance scores over attributes, we develop a Monte Carlo simulation model of this decision. The simulation was controlled, using ten unique seed values to ensure that the difference in simulation output due to random variation was the same for each alternative.

3.1 Trapezoidal Fuzzy Simulation

The trapezoidal fuzzy input dataset can also be simulated.

X is random number (0 < rn < 1)

Definition of trapezoid:
a1 is left 0
a2 is left 1
a3 is right 1
a4 is right 0

Contingent calculation:
J is area of left triangle
K is area of rectangle
L is area of right triangle
Fuzzy sum = left triangle + rectangle + right triangle = 1

M is the area of the left triangle plus the rectangle (for calculation of X value)

X is the random number drawn (which is the area)

If $X \le J$:

$$X = a1 + \sqrt{\frac{X \times (a2 - a1) \times (a4 - a3 + a2 - a1)}{J + L}} \tag{2}$$

If $J \le X \le J{+}K$:

$$X = a2 + \frac{X - J}{K} \times (a3 - a2) \tag{3}$$

If $J{+}K \le X$:

$$X = a4 - \sqrt{\frac{(1 - X) \times (a4 - a3) \times (a4 - a3 + a2 - a1)}{J + L}} \tag{4}$$

Our calculation is based upon drawing a random number reflecting the area (starting on the left (a1) as 0, ending on the right (a4) as 1), and calculating the distance on the X-axis. The simulation software Crystal Ball was used to replicate each model 1,000 times for each random number seed. The software enabled counting the number of times each alternative won. Probabilities given in Table 7 are thus simply the number of times each alternative had the highest value score divided by 1,000. This was done ten times, using different seeds. Therefore, mean probabilities and standard deviations (std) are based on 10,000 simulations. The Min and Max entries are the minimum and maximum probabilities in the ten replications shown in the table.

Table 7. Simulated Probabilities of Winning for Uniform Fuzzy Input

Trapezoidal	Antônio	Fábio	Alberto	Fernando	Isabel	Rafaela
seed1234	0.337	0.000	0.188	0.046	0.429	0.000
seed2345	0.381	0.000	0.168	0.040	0.411	0.000
seed3456	0.346	0.000	0.184	0.041	0.429	0.000
seed4567	0.357	0.000	0.190	0.046	0.407	0.000
seed5678	0.354	0.000	0.210	0.052	0.384	0.000
seed6789	0.381	0.000	0.179	0.046	0.394	0.000
seed7890	0.343	0.000	0.199	0.052	0.406	~ 0.000
seed8901	0.328	0.000	0.201	0.045	0.426	~ 0.000
seed9012	0.353	0.000	0.189	0.048	0.410	~ 0.000
seed0123	0.360	0.000	0.183	0.053	0.404	~ 0.000
~	~					
min	0.328	0.000	0.168	0.040	0.384	~ 0.000
mean	0.354	0.000	0.189	0.047	0.410	~ 0.000
max	0.381	0.000	0.210	0.053	0.429	~ 0.000
std	0.017	0.000	0.012	0.004	0.015	~ 0.000

3.2 Analysis of Results

The results for each system were very similar. Differences were tested by t-test of differences in means by alternative. None of these difference tests were significant at the 0.95 level (two-tailed tests). This establishes that no significant difference in interval or trapezoidal fuzzy input was detected (and any that did appear would be a function of random numbers only, as we established an equivalent transformation). A recap of results is given in Table 8.

Table 8. Probabilities Obtained

~	Antônio	Fábio	Alberto	Fernando	~	Isabel	Rafaela
Grey-Related	-		-	-	~	X	-
Interval average	**0.358**	**~ 0**	**~ 0.189**	**~ 0.047**	**~**	**0.410**	**~ 0**
Interval minimum	0.336	~ 0	~ 0.168	~ 0.040	~	0.384	~ 0
Interval maximum	0.393	~ 0	~ 0.210	~ 0.053	~	0.429	~ 0
Trapezoidal average	**0.354**	**~ 0**	**~ 0.189**	**~ 0.044**	**~**	**0.409**	**~ 0**
Trapezoidal minimum	0.328	~ 0	~ 0.171	~ 0.035	~	0.382	~ 0
Trapezoidal maximum	0.381	~ 0	~ 0.206	~ 0.051	~	0.424	~ 0

Isabel still wins, but at a probability just over 0.4. Antônio was second with a probability just over 0.35, followed by Alberto at about 0.19 and Fernando at below 0.05. There was very little overlap among alternatives in this example. However, should such overlap exist, the simulation procedure shown would be able to identify it. As it is, Isabel still appears a good choice, but Antônio appears a strong alternative. While this example is on a small set of data, the intent was to demonstrate what could be done in that context could be applied on large-scale data sets as well. Our proposal is unique to our knowledge, proposing the use of simulation to more fully use grey-related data that more accurately reflects the real problem. If this could be done with small-scale data sets, our contention is that it can also be done with large-scale data sets in a data mining context.

4 Grey Related Decision Tree Model

Grey related analysis is expected to provide improvement over crisp models by better reflecting the uncertainty inherent in many human analysts' minds. Data mining models based upon such data are expected to be less accurate, but hopefully not by very much. However, grey related model input would be expected to be stabler under conditions of uncertainty where the degree of change in input data increased.

We applied decision tree analysis to a small set (1,000 observations total) of credit card data. Originally, there was one output variable (whether or not the account defaulted, a binary variable with 1 representing default, 0 representing no default) and 65 available explanatory variables. These variables were analyzed and 26 selected as representing ideas that might be important to predicting the outcome. The original data set was imbalanced, with 140 default cases and 860 not defaulting. Initial decision tree models were almost all degenerate, classifying all cases as not defaulting. When differential costs were applied, the reverse degenerate model was obtained (all cases predicted to default). Therefore, a new dataset containing all 140 default cases and 160 randomly selected not default cases was generated, from which 200 cases were randomly selected as a training set, with the remaining 100 cases used as a test set.

The explanatory variables included five binary variables and one categorical variable, with the remaining 20 being continuous. To reflect fuzzy input, each variable (except for binary variables) was categorized into three categories based upon analysis of the data, using natural cutoff points to divide each variable into roughly equal groups.

Decision tree models were generated using the data mining software PolyAnalyst. That software allows setting minimum support level (the number of cases necessary to retain a branch on the decision tree), and a slider setting to optimistically or pessimistically split criteria. Lower support levels allow more branches, as does the optimistic setting. Every time the model was run, a different decision tree was liable to be obtained. But nine settings were applied, yielding many overlapping models. Three unique decision trees were obtained, reflected in the output to follow. There were a total of eight explanatory variables used in these three decision trees. The same runs were made for the categorical data reflecting grey related input. Four unique decision trees were obtained, with formulas again given below. A total of seven explanatory variables were used in these four categorical decision trees.

These models were then entered into a Monte Carlo simulation (supported by Crystal Ball software). A perturbation of each input variable was generated, set at five different levels of perturbation. The intent was to measure the loss of accuracy for crisp and grey related models.

The model results are given in the seven model reports in the appendix. Since different variables were included in different models, it is not possible to directly compare relative accuracy as measured by fitting test data. However, the means for the accuracy on test data for each model given in Table 9 show that the crisp models declined in accuracy more than the categorical models. The column headings in Table 9 reflect the degree of perturbation simulated.

The fuzzy models were expected to be less accurate, but here they actually averages slightly better accuracy. This, however, can simply be attributed to different variables being used in each model. The one exception is that models Continuous

Table 9. Mean Model Accuracy

Model	**crisp**	**.25**	**.50**	**.00**	**.00**	**.00**	**.00**	**.25**
Continuous 1	.70	.70	.70	.068	.67	.66	.65	.70
Continuous 2	.67	.67	.67	.67	.67	.66	.66	.67
Continuous 3	.71	.71	.70	.69	.67	.67	.66	.71
Continuous	**.693**	**.693**	**.690**	**.680**	**.670**	**.667**	**.657**	**.693**
Categorical 1	.70	.70	.68	.67	.66	.66	.65	.70
Categorical 2	.70	.70	.70	.69	.68	.67	.67	.70
Categorical 3	.70	.70	.70	.69	.69	.68	.67	.70
Categorical 4	.70	.70	.70	.69	.68	.67	.67	.70
Categorical	**.700**	**.700**	**.695**	**.688**	**.678**	**.670**	**.665**	**.700**

2 and Categorical 3 were based on one variable, V64, the balance-to-payment ratio. The cutoff generated by model Continuous 2 was 6.44 (if V64 was < 6.44, prediction 0), while the cutoff for Categorical 3 was 4.836 (if V64 was > 4.835, the category was "high", and the decision tree model was that if V64 = "high", prediction 1, else prediction 0). The fuzzy model here was actually better in fitting the test data (although slightly worse in fitting the training data). The important point of the numbers in Table 9 is that there clearly was greater degradation in model accuracy for the continuous models than for the categorical (grey related) models.

5 Conclusions

This paper has discussed the integration of grey-related analysis and decision making with uncertainty. Grey related analysis, a method for the multiattribute decision making problem, is demonstrated by a case study. Results based on Monte Carlo simulation as a data mining technique offers more insights to assist our decision making in fuzzy environments by incorporating probability interpretation. Analysis of decision tree models through simulation shows that there does appear to be less degradation in model fit for grey related (categorical) data than for decision tree models generated from raw continuous data. It must be admitted that this is a preliminary result, based on a relatively small data set of only one type of data. However, it is intended to demonstrate a point meriting future research. This decision making approach can be applied to large-scale data sets, expanding our ability to implement data mining and large-scale computing.

References

1. Aouam T., Chang S.I., Lee E.S.: Fuzzy MADM: An outranking method, European Journal of Operational Research 145:2 (2003) 317-328
2. Hu Y., Chen, R.; Tzeng, G.H.: Finding fuzzy classification rules using data mining techniques. Pattern Recognition Letters Volume: 24, Issue: 1-3, (2003) 509-519
3. Deng J.L.: Control problems of grey systems. Systems and Controls Letters 5, (1982) 288-294
4. Dubois D., Prade H.: Fuzzy Sets and Systems: Theory and Applications, New York: Academic Press, Inc., (1980)
5. Pedrycz, Witold : Fuzzy set technology in knowledge discovery. Fuzzy Sets and Systems 98 (1998) 279-290
6. Rocco S., Claudio M.A rule induction approach to improve Monte Carlo system reliability assessment. Reliability Engineering and System Safety Volume: 82, Issue: 1, October, (2003) 85-92
7. Pawlak, Z.: Rough sets, International Journal of Information & Computer Sciences 11 (1982) 341 -356
8. Pearl, J.: Probabilistic reasoning in intelligent systems, Networks of Plausible inference, Morgan Kaufmann,San Mateo,CA (1988)
9. Gau W.L., Buehrer D.J.: Vague sets. IEEE Trans, Syst. Man, Cybern, 23(1993) 610-614
10. Olson D.L., Wu D.: Simulation of Fuzzy Multiattribute Models for Grey Relationships. Accepted by European Journal of Operational Research(2005)

Complex Networks and Networked Data Mining

Deyi Li, Guisheng Chen, and Baohua Cao

China Institute of Electronic System Engineering,
Beijing, 100840
ziqinli@public2.bta.net.cn

There have been numerous and various complex networks with the development of science, technology and human society, such as the Internet, the World Wide Web, the network of air lines, large-scale electric power networks, the structure of a piece of Very Large-Scale Integration (VLSI), the human social relationships, the neural networks, and the spreading path net of an infectious disease, etc. Even in the study on semantics of human language, the relationship between synonyms can also be represented and analyzed via complex networks. Most researchers widely apply the parameters of degree distribution, clustering coefficient and average distance to analyze efficiently the uncertainty of complex networks.

Watts and *Strogatz* proposed the Small World Model of complex networks to describe the transition from completely regular graphs to completely random graphs in their paper on Nature in 1998, and the related approach to configure small world feature of complex networks has been studied for years. In 1999, *Barabási Albert* proved empirically and theoretically that, considering the topology of most complex networks, the node degree distribution follows a power-law distribution, and then illustrated the significant scale-free feature of complex networks by using the BA model, in which two essential clues, growth and preferential attachment, lead to the generation of a scale free network via self-organization. Due to the general characteristics of small world effect and scale free feature, more and more researchers in distinguished fields all focus on the study of complex networks. However, the presupposition of forming a network with both small world effect and scale free feature, is not theoretically and mathematically described so far.

In this paper, the general laws on uncertainties in complex networks are proposed, such as the varying trend of degree distribution from homogenously to non-homogenously, or from democratization to centralization, while considering the evolutional relationships among regular graphs, random graphs, small world networks, scale free networks, hierarchical networks and star-networks. Furthermore, the corresponding methods and algorithms are discussed and designed for simulation, which aims at keeping the scale free feature while changing the topology by dynamically increasing or decreasing nodes and links in networks. Depending on the innovative operations, the network scale can be rapidly reduced, therefore the study cost over a complex network is saved via decaying those nodes and links and maintaining the original essential characteristics. Contrarily, via adding up certain nodes and links, the growing trend of an initial small network can be predicated and planned.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 10 – 12, 2005.

Considering the complex networks in the real world, studying mathematically the mere two factors of node and link can not meet the variety of demands. For example, the Next Generation Internet, beyond the merely nodes and links, there exist some other significant factors which can not be omitted, such as geographical distance, bandwidth of a link, and the number of data warehouse on a node. In order to manage and control the computational resource, communication resource, software resource, storage resource and information resource, to deal with the coexistence of grid computing and peer-to-peer computing, the sequential character in a disorder network state, the certainty feature in a uncertain process, the collaboration in competitive activities are need to be discovered.

Consequently, methods for mapping some more important factors abstracted from a real complex network into the topology of nodes and links, are proposed, in which the effect of node is denoted with the computable quality, such as a city scale with traffic network, node throughput of communication network, the hit rates of a web site, and the individual prestige of human relationship, etc., and meanwhile, the interaction between nodes may be denoted by the distance or length of links, such as the geographic distance between two cities in the traffic network, the bandwidth between two communication nodes, the number of hyperlinks in WWW and the friendship intensity of human relationship. That is, topologically, two-factor operations with node and link are generally expanded to four-factor operations with node, link, distance, and quality. Explicitly, using this four-factor method, we analyze the networked data and simulate the optimization of web mining to form a mining engine by excluding those redundant and irrelevant nodes,exploring the clustering community and then reducing complicated topology structures to a new informative concise graph. Following the model design and implementation of prototype for mining informative structure, several experiments based on real networked data sets have been showing the encouraging results on both discovered knowledge and discovered rate (See Fig. 1 and Fig.2).

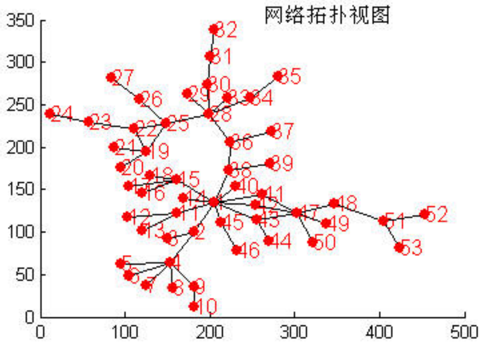


Fig. 1. Networked data

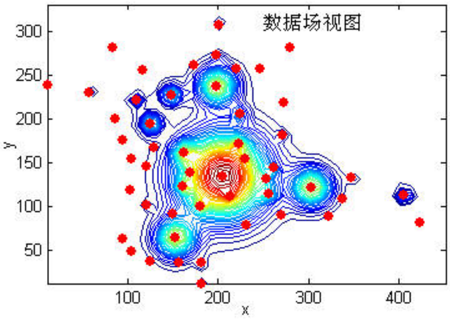


Fig. 2. The informative structure of the Networked data in Fig.1

References

1. Albert R., Barabási A-L.: Statistical mechanics of complex networks. Review of Modern Physics, 74 (2002) 47~91.
2. Cancho R. F., Sole R. V.: The small-world of human language. Proc. R. Soc. London, Ser. B268,(2001) 2261 - 2265
3. Watts D.J., Strogatz S.H..: Collective dynamics of "small-world" networks. Nature, 393(1998) 440~442
4. Barabási A-L., Albert R.. Emergence of scaling in random networks. Science, 286(1999), 509 ~ 512

In-Depth Data Mining and Its Application in Stock Market[*]

Chengqi Zhang and Shichao Zhang

Faculty of Information Technology,
University of Technology, Sydney,
PO Box 123, Broadway, NSW 2007, Australia
{chengqi, zhangsc}@it.uts.edu.au

Abstract. Existing association rule mining algorithms are specifically designed to find strong patterns that have high predictive accuracy or correlation. Many useful patterns, for example, out-expectation patterns with low supports, are certainly pruned for efficiency in these mining algorithms. This talk introduces our ongoing research developing novel theories, techniques and methodologies for discovering hidden interactions within data, such as class-bridge rules and out-expectation patterns. These patterns are essentially different from traditional association rules, but are much more useful than traditional ones to applications such as cross-sales, trend prediction, detecting behavior changes, and recognizing rare but significant events. This delivers a paradigm shift from existing data mining techniques. In addition, the system of applying these techniques to stock market is briefly presented.

[*] This work is partially supported by Australian large ARC grants (DP0343109 and DP0559536), a China NSFC major research Program (60496321), and a China NSFC grant (60463003).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, p. 13, 2005.

Relevance of Counting in Data Mining Tasks

Osmar R. Zaïane

Department of Computing Science,
University of Alberta,
Edmonton AB, Canada
zaiane@cs.ualberta.ca

Abstract. In many languages, the English word "computer" is often literally translated to "the counting machine." Counting is apparently the most elementary operation that a computer can do, and thus it should be trivial to a computer to count. This, however, is a misconception. The apparently simple operation of enumeration and counting is actually computationally hard. It is also one of the most important elementary operation for many data mining tasks. We show how capital counting is for a variety of data mining applications and how this complex task can be achieved with acceptable efficiency.

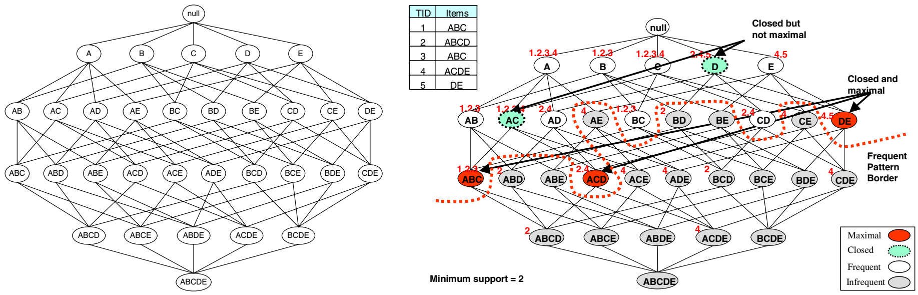
1 Introduction

Counting is an elementary computer operation. Of course for humans counting has its limitations, but for computers one might think it is a trivial task. All computer programs entail counting in one form or another. From business management programs to programs for scientific research, counting is omnipresent in the implementations of these programs. However, counting can be very complex. In particular, the scalability issue with counting can be of a major concern.

For example, consider an alphabet of 5 letters {a, b, c, d, e}. The number of all possible subsets is 32 (i.e $2^5 = 32$). All these combinations are shown in Figure 1. These 5 letters could be the 5 unique products that a specialized store sells. The combinations illustrated in Figure 1 are the possible transactions that potential customers have when visiting this store. To study the relationships between products bought together, one might take existing real transactions and for each transaction, check the combinations, and for each of them traverse the graph in Figure 1 to increment the respective counters. Let us take the example now of a more realistic department store with 10,000 different products. The graph in Figure 1 would explode to $2^{10,000}$ nodes. Traversing the graph efficiently to find and increment the exact counter is complex, but even keeping the complete graph resident in main memory is phenomenal.

To make things more complicated, let us reduce the alphabet to only 4 {A, C, G, T} and now allow the items to repeat in a combination. In a very long sequence of these 4 items, counting the existing different combinations (or subsequences) of different lengths in this sequence is almost unthinkable. This is

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 14–18, 2005.



TID	Items
1	ABC
2	ABCD
3	ABC
4	ACDE
5	DE

Fig. 1. Left:Search Space with an alphabet of 5 items. Right: Example database and the frequent pattern border at minimum support of 2

a common problem in genomics where the 4 letter alphabet is Adenine, Cytosine, Guanine, Thymine and the sequences of DNA are hundreds of thousands of elements long. In proteomics, studying the proteins, the alphabet is of 20 amino-acids making counting a more daunting task.

2 Enumeration and Counting in Data Mining

Here are some typical examples of data mining applications where counting is important and at the same time can be overwhelming.

2.1 Frequent Itemset Mining

Enumerating and counting frequent itemsets is the first and most important phase in the process of mining for association rules. The illustrative example given above is an example of market basket analysis, the typical application for association rules [1], However, the counting of itemsets is also an intricate part of many other data mining tasks and applications such as classification [8, 13] and clustering [4].

2.2 Event Sequence Analysis and Sequential Patterns

When items, events or measurements are chronologically ordered, the time element becomes relevant and should be taken into account. There are many variations of these sequences depending upon the nature of the elements in the sequence. If they are nominal symbols from a given alphabet, the sequence is known as a temporal event sequence [12] such as the events on a power or telecommunication grid or simple click-streams on a web site; if they are continuous valued elements, the sequence is known as time series [10] such as stock market feeds or meteorological data. Detecting important subsequences or building predictive models from these data require sophisticated counting.

2.3 Frequent Subsequence Analysis

In bioinformatics, identifying and counting significant sub-sequences in a set of very long sequences is important for the understanding of protein functions, the identification of transcriptor factors and even the reconstruction of genome phylogeny [11]. Given the particularly large sequences and close to infinite search space, erudite methods for counting sub-sequences were devised [7].

2.4 Contrast Sets

Contrast sets [2] are used to describe the fundamental differences between groups. Simply put, they are conjunctions of items that differ meaningfully in their distributions across groups. This again entails counting. Another variation of these are emerging patterns [5].

3 Top-Down Versus Bottom-Up Enumeration

Looking at Figure 1, it is obvious that when the alphabet is large, enumerating all nodes is onerous if not infeasible. The clever idea used in [1] and later in many other publications is based on the [illegible] property or observation. There is no need to visit a node and all its descendents if one if its ancestors is not frequent. The goal is to find the frequent pattern border (Figure 1, Right) above which itemsets are frequent and below which itemsets are irrelevant. Nevertheless, this approach may yield too many useless enumerations for nodes that are doomed to be irrelevant. This is particularly the case for datasets with long patterns (i.e. the frequent pattern border is deep in the graph). The bottom-up approach, starting from the long patterns and going toward the empty set searching for the frequent pattern border is also interesting with very clever heuristics for pruning. It is however burdensome if this border is high in the graph (i.e. the relevant patterns are short). Many attempts at reducing the enumeration of frequent patterns were done by concentrating on non redundant itemsets such as closed [9] and maximal patterns [3], but the main strategies remain the same: either top-down or bottom-up.

4 Leap Approach

There are two issues in frequent itemset mining: relevant itemset enumeration and counting the exact frequencies. Discovering all frequent patterns from the non-redundant sets such as the maximal patterns, does not necessarily mean we get de facto their exact counts. Moreover, every superfluous enumeration and counting of an itemset doomed infrequent is definitely time-consuming and useless in the final result. The idea of a leap traversal of the search space is, rather than systematically traverse the graph top-down or bottom-up, to cunningly jump from one node to the other avoiding as much as possible the superfluous nodes doomed irrelevant. The leap traversal searches for the frequent pattern border by identifying maximal patterns and collecting enough information in the process to generate exact counts for all frequent patterns at the end. The idea is that maximal patterns, which subsume all frequent patterns, are frequent sub-transactions. The process starts by marking some interesting nodes that appear as complete sub-transactions of frequent unique items. The leap is made from those marked nodes identified as non maximals and the jump goes to the node resulting from intersecting the marked nodes that are not maximals [6]. The jumps are often many levels ahead, avoiding many irrelevant nodes. On average, the leap traversal generates and tests more than one order of magnitude less candidates than other traversal strategies and produces the exact same results at the end of the process. This approach can mine millions of transactions with hundreds of thousand items on a small desktop in a reasonable time (i.e. few seconds). However, while linear scalability is achieved, with larger and larger real application datasets, physical limits are quickly reached. Current hardware and state-of-the-art algorithms can not cope realistically. More clever ideas are needed for real world enumeration and counting problems.

5 Conclusion

Mining for frequent itemsets is a canonical task, fundamental for many data mining applications. It is used to generate association rules, produce contrast sets, count frequent subsequences in event sequences and time series, estimate probabilities in a belief network, and even create a rule-based classification model or cluster data. It is the primary operation for data analysis. Yet, it is still an open problem how to achieve this counting efficiently. Many algorithms have been reported in the literature, some original and others extensions of existing techniques. While we showed some effective approaches for this task, the problem of how to improve on the existing methods remains a challenging puzzle for the future.

References

1. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. 1994 Int. Conf. Very Large Data Bases*, pages 487–499, Santiago, Chile, September 1994.
2. S. Bay and M. Pazzani. Detecting changes in categorical data: Mining contrast sets. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 302–306, San Diego, USA, 1999.
3. R. J. Bayardo. Efficiently mining long patterns from databases. In *ACM SIGMOD*, 1998.
4. F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD'02)*, Edmonton, Canada, 2002.
5. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In *ACM SIGKDD International Conference on Knowledge Discovery and Data MIning*, pages 43–52, San Diego, USA, 1999.

6. M. El-Hajj and O. R. Zaïane. Cofi approach for mining frequent itemsets revisited. In *9th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery (DMKD-04)*, pages 70–75, Paris, France, June 2004.
7. D. Gusfield. *Algorithms on Strings, Trees, and sequences: Computer Science and Coputational Biology.* Cambridge University Press, 1997.
8. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In *4th Intl. Conf. on Knowledge Discovery and Data Mining (KDD'98)*, pages 80–86, New York City, NY, August 1998.
9. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory (ICDT)*, pages pp 398–416, January 1999.
10. A. Weigend and N. Gershenfeld. *Time Series Prediction: Forecasting the Future and Understanding the Past.* Addison-Wesley, 1993.
11. X. Wu, X. Wan, G. Wu, D. Xu, and G.-H. Lin. Whole genome phylogeny construction via complete composition vectors. Technical Report TR05-06, Department of Computing Science, University of Alberta, January 2005.
12. Q. Yang, H. Wang, and W. Zhang. Web-log mining for quantitative temporal-event prediction. *IEEE Computational Intelligence Bulletin*, 1(1):10–18, December 2002.
13. O. R. Zaïane and M.-L. Antonie. Classifying text documents by associating terms with text categ ories. In *Proc. of the Thirteenth Australasian Database Conference (A DC'02)*, pages 215–222, Melbourne, Australia, January 2002.

Term Graph Model for Text Classification*

Wei Wang, Diep Bich Do, and Xuemin Lin

University of New South Wales, Australia
{weiw, s2221417, lxue}@cse.unsw.edu.au

Abstract. Most existing text classification methods (and text mining methods at large) are based on representing the documents using the traditional vector space model. We argue that important information, such as the relationship among words, is lost. We propose a term graph model to represent not only the content of a document but also the relationship among the keywords. We demonstrate that the new model enables us to define new similarity functions, such as considering rank correlation based on PageRank-style algorithms, for the classification purpose. Our preliminary results show promising results of our new model.

1 Introduction

In recent years, text mining has become one of the most popular research areas in data mining, due to the rapid growth and evolution of digital text documents, such as Web pages, office documents, and E-mails. As the demand to organize these documents automatically is constantly rising, [illegible] (or [illegible]) become one active subfields for data mining researchers. Text classification deals with the problem of automatically assigning single or multiple category (or class) labels to a new text document based after learning from a set of training documents with correct category labels.

Most existing text classification methods (and text mining methods at large) adopt the approach of transforming the text mining problem into traditional machine learning problem, where a large number of mature techniques can be applied [7, 14, 17, 18, 8]. Usually, the conversion of a text document into a relational tuple is performed using the popular [illegible] model. Intuitively, the document is parsed, cleaned and stemmed, in order to obtain a list of [illegible] with corresponding frequencies. Then a corresponding vector can be constructed to represent the document. Therefore, a collection of documents can be represented by a [illegible] matrix, which can be subsequently interpreted as a relational table.

However, the vector space model only allow preserving fundamental features of the document. Although a few alternative weighting scheme other than term frequency have been proposed, one common weakness is that they don't take into consideration the associations among terms. Recent studies have revealed that association among terms could provide rich semantics of the documents and

* This work was partially supported by ARC Discovery Grant – DP0346004.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 19–30, 2005.

serve as the basis of a number of text mining tasks [9]. However, the approach proposed in [9] discards the vector space model and uses frequently co-occurring terms only.

In this paper, we propose a novel model for text document by combining the strengths of vector space model and frequently co-occurring terms together. The result is called the [illegible]. The basic idea is to mine the associations among terms, and then capture all these information in a graph. We use text classification to illustrate the potential application of the new model. To that end, we design two novel similarity functions. One is based on Google's page-rank style algorithm [3] to discover the weights of each terms. The other is based on the distances of all pairs of terms. Our preliminary experimental results shows our new model is promising.

The rest of the paper is organized as follows. Section 2 introduces related works in text classification. Section 3 presents the proposed term graph model which is capable of capturing more information than the vector space model for text documents. In Section 4, we propose methods to classify text documents represented in the term graph model. Experimental results based on the Reuters-21578 text collection is described in section 5. We conclude the paper in Section 6.

2 Related Work

An immense amount of work has been done in the area of text classification in the past decade. We refer readers to [15] for a recent survey. In the rest of this section, we will only focus on several work that is most related to our approach in this section.

The [illegible] (SVM) technique — a popular and highly accurate machine learning method for classification problems — was first in introduced in the early 1990s [5]. In 1998, the study proposed by Joachims explored the benefits of using SVM for text categorization [8]. SVM-based approaches can handle large feature spaces with excellent classification accuracy. As a result, SVM-based system has ability to work well for the standard text corpus. The results in [8] shows that SVM-based method is more accurate than alternative approaches. SVM has also been suggested in [14, 18] as one of the most outperforming classifiers in comparison with a set of alternative text categorization methods. One weakness of SVM-based text categorization system is that it cannot scale well with the number of documents in the text collections.

Text categorization based on association rule mining is also another promising approach. [1] proposed two different methods for generating text classifier based on associating the words of a document and its pre-defined categories. These methods are called the Association Rule-based Categorizer By Category (ARC-BC) and the Association Rule-based Categorizer for All Categories (ARC-AC). The main ideas in both approaches are:

1. Present each training document as a transactions of terms.
2. Use a special association rule mining algorithm that is guided by constraints to produce the expected rules in the form of $T \Rightarrow c_i$ where T is set of terms. The results then will be used directly for classification.

The difference between the two method lies in the granularity when forming the text collection. In the ARC-AC algorithm, all the categories form a text collection and the only set of rules generated form the classifier. Meanwhile, ARC-BC algorithm considers each category as a separate text collection and a distinct set of association rules is generated for each category.

More recently, document level frequent itemsets is explored more for other problems like text clustering and learning from the web. Liu et al. [9] introduced a novel system to mine [illegible] knowledge on the Web. The intuition is that the frequent word phrases in a collection of web pages of the same topic are most likely to be the sub-topics or the salient concepts. We can find out several different methods and system for clustering transactions [16] and documents [2, 6]. These are all based on the intuition that there should be many frequent itemsets within a cluster and different clusters have little overlapping of such frequent itemsets.

3 Term Graph Model

3.1 Overview of the Term Graph Model

The term graph model is an improved version of the vector space model [13] by weighting each term according to its relative "importance" with regard to term associations. Specifically, for a text document D_i, it is represented as a vector of term weights $\boldsymbol{D_i} =< w_{1i}, \ldots, w_{|T|i} >$, where T is the ordered set of terms that occur at least once in at least one document in the collection. Each weight w_{ji} represents how much the corresponding term t_j contribute to the semantics of document d_i. Although a number of weighting schemes have been proposed (e.g., boolean weighting, frequency weighting, tf-idf weighting, etc.), those schemes determine the weight of each term [illegible]. As a result, important yet rich information regarding the relationships [illegible] the terms are not captured in those weighting schemes.

We propose to determine the weight of each term in a document collection by constructing a [illegible]. The basic steps are as follows:

1. [illegible]: For a collection of document, extract all the terms.
2. [illegible]:
   (a) For each document, we view it as a transaction: the document ID is the corresponding transaction ID; the terms contained in the document are the items contained in the corresponding transaction. Association rule mining algorithms can thus be applied to mine the frequently co-occurring terms that occur more than *minsup* times in the collection.
   (b) The frequent co-occurring terms are mapped to a weighted and directed graph, i.e., the [illegible].

We will introduce the details of each step as follows.

3.2 Preprocessing

In our term graph model, we will capture the relationships among terms using the frequent itemset mining method. To do so, we consider each text document in the training collections as a transaction in which each word is an item. However, not all words in the document are important enough to be retained in the transaction. To reduce the processing space as well as increase the accuracy of our model, the text documents need to be preprocessed by (1) remove stopwords, i.e., words that appear frequently in the document but have no essential meanings; and (2) retaining only the root form of words by stemming their affixes as well as prefixes. We use Lancaster algorithm for stemming [11].

3.3 Graph Building

As mentioned above, we will capture the relationships among terms using the frequent itemset mining method. While this idea has been explored by previous research [9], our approach distinguish from previous approaches in that we maintain all such important associations in a graph. The graph not only reveals the important semantics of the document, but also provide a basis to extract novel features about the document, as we will shown in the next section.

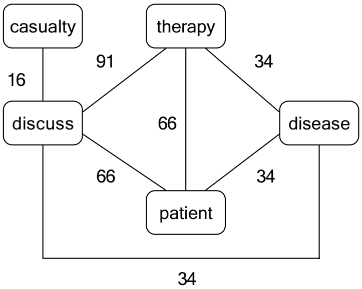
[illegible] After the preprocessing step, each document in the text collection will be stored as a transaction (list of items) in which each item (term) is represented by a unique non-negative integer. Then frequent itemset mining algorithms can be used to find all the subset of items that appeared more than a threshold amount of times (controlled by *minsup*) in the collection. In our implementation, we use the AFOPT algorithm [10].

[illegible] In our system, our goal is to explore the relationships among the important terms of the text in a category and try to define a strategy to make use of these relationships in the classifier and other text mining tasks. Vector space model cannot express such rich relationship among terms. Graph is thus the most suitable data structure in our context, as, in general, each term may be associated with more than one terms.

We propose to use the following simple method to construct the graph from the set of frequent itemsets mined from the text collections. First, we construct a node for each unique term that appear at least once in the frequent itemsets.

Itemset	Support
$\{therapy, discuss\}$	91
$\{therapy, discuss, patient\}$	66
$\{therapy, discuss, patient, disease\}$	34
$\{casualty, discuss\}$	16

(a) Frequent Itemsets



(b) The Corresponding Graph

Fig. 1. An Example Term Graph

Then we create edges between two node u and v if and only if they are both contained in one frequent itemset. Furthermore, we assign weights to the edges in the following way: the weight of the edge between u and v is the largest support value among all the frequent itemsets that contains both of them.

[illegible] Consider the frequent itemsets and their absolute support shown in Figure 1(a). Its corresponding graph is shown in Figure 1(b).

4 Text Classification

Our term graph model encapsulates richer information than the traditional vector space model. As we preserve and extract the hidden relationships among terms in the document collection, we argue that many text mining applications can benefit from this model. Specifically, we consider the classic text classification problem with our new model.

One central notion to the classification is the similarity (or distanct) function of a document and a category. We consider two different approaches based on the term graph model. In the first approach, we borrow the idea of PageRank ranking of the web pages [3] to assign weights to the nodes (i.e., terms) in the term graph; we can then measure the similarity of a document and a category using a rank correlation coefficient [12] based on the ranks of the terms. In the second approach, we define a similarity formula based on the distance matrix of the term graph. More details about those two approaches are described in the following sub-sections.

4.1 Classification Based on the Term Ranks

Ranking Terms. PageRank is a well-known method for measuring the relative importance of the web pages based on their linking information. According to [3], the basic intuition of the PageRank is that a page will have a high rank if there are many pages in the web point to it, or if there are some pages with high ranks pointing to it. By following the same idea, we can determine the "PageRank" scores for the nodes in the term graph (or a document or a category) too. Intuitively, if a word that appears frequently with many other words in the text collections, it is an important word; words that appear together with some important words may also be important.

Since the original Pagerank computation algorithm accept as input a directed, unweighted graph, we need to use the following transformation on our term graph:

- treat each node in the term graph as a web page.
- treat each edge between node u and v with weight w in the graph as $2w$ links; w of them are $u \rightarrow v$ and the other w links as $v \rightarrow u$.

The output can be directly feed into the PageRank computation algorithm. An example of the term graph with PageRank scores computed for each node can be found in Figure 4.

Rank Correlation Coefficient. After calculating the rank of the nodes in the graph using the idea of PageRank, each term a the document is assigned a PageRank value. One simple method to calculate the similarity between a document and a category is as follows: we directly use the PageRank values of the terms as their weights and existing document similarity functions (for instance, the cosine similarity) can be directly applied. In this sense, we can view the process of constructing the term graph and calculating the PageRank for each term as an preprocessing step to obtain yet another [illegible].

However, we argue that the relative order rather than the absolute values of the PageRank scores are meaningful. Therefore, we propose another similarity metric based on the concept of rank correlation. The basic idea is that if a document belongs to a category, the relative order of terms appearing in the document and the documents belonging to the category should be consistent.

To compute the rank correlation coefficient, we need to obtain the rank of each term. This can be computed by sorting the terms by their PageRank scores in the descending order. We can do this for a (testing) document as well as a category, where we treat all the documents belonging to the category in question as a single document.

It is well-known that a robust statistics to measure the correlation of two arrays of size N is the non-parametric correlation scores, for example, the Spearman Rank-Order Correlation Coefficient [12]. Specifically, let R_i be the rank of x_i among the other x's, S_i be the rank of y_i among the other y's. Then the rank-order correlation coefficient is defined to be the linear correlation coefficient of the ranks, namely,

$$r_s = \frac{\sum_i^N (R_i - \overline{R})(S_i - \overline{S})}{\sqrt{\sum_i^N (R_i - \overline{R})^2}\sqrt{\sum_i^N (S_i - \overline{S})^2}} \tag{1}$$

However, there ordered terms of a document and a category are usually of different length. Therefore, Equation 1 cannot be directly applied. We propose the following heuristics to solve this problem, as shown in Figure .

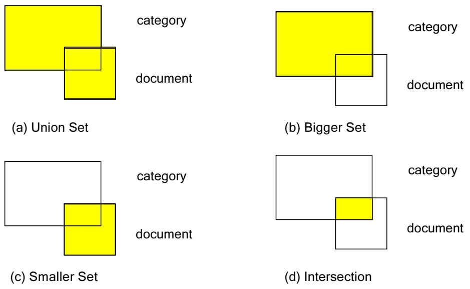


Fig. 2. Four Heuristics to Generate the Rank Correlation Cofficient

1. **Union Set.** In this heuristic, we consider all the terms from the document and the category. In order to calculate the rank correlation, we need to assign rank values to terms that do not appear in the document (or the category). We simply assign the same rank to all the terms that are unique to one input For example, suppose there are $p + n$ terms for the document and $p + m$ terms for the category (that is, there are p terms that appear in both the document and the category). We will find all the term that appear only in the document and assign a rank which is the average rank from $p + n + 1$ to $p + m + n$, i.e., $p + n + m/2$. For all the terms that appear only in the category, they are assign a rank which is the average rank from $p + m + 1$ to $p + m + n$, i.e., $p + m + n/2$.
2. **Bigger Set.** In this heuristic, we only consider all the terms from the bigger term collection, which is usually the category. Similarly, for the same example, we assign a rank which is the average rank from $p + m + 1$ to $p + m + n$, i.e., $p + m + n/2$.
3. **Smaller Set.** In this heuristic, we only consider all the terms from the smaller term collection, which is usually the document. Similarly, for the same example, we assign a rank which is the average rank from $p + n + 1$ to $p + m + n$, i.e., $p + n + m/2$.
4. **Intersection.** In this heuristic, we only consider all the terms that appear in both the document and the category. Therefore, we do not need to adjust the rank of any terms.

Two vectors of ranks of the same size will be generated after using any of the above heuristics. The vectors will be directly used as the inputs for the Spearman Rank-Order Correlation Coefficient algorithm. The result measures how similar the document and the category is, and will be used in our k-NN classifier.

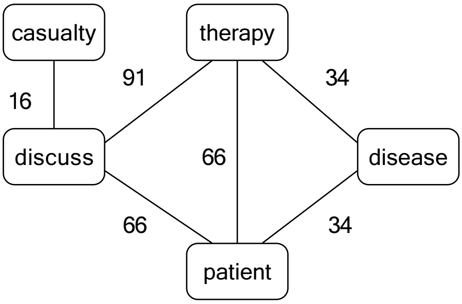
Classification. We adopt the following simple classifier to perform the text classification. We first build a set of vectors of rank values, $\{V_1, V_2, \ldots, V_n\}$, representing the categories $\{C_1, C_2, \ldots, C_n\}$ from the training set. For each testing document, a vector of rank values, F, representing the testing document D is calculated. We search for the category C such that it has the highest rank correlation coefficients with the testing document D. Then the document is assigned to the category C.

4.2 Classification Based on the Term Distances

Another similarity function we propose is based on the intuition that the distance between two terms in the term graph reflects the relations between the terms. Intuitively, terms that appear more often in the text collections will have more chances to be connected directly in the term graph.

Term Distance Matrix. Given a term graph, we can build its term distance matrix as follows. Assume the graph has n terms. Its term distance matrix T is of size $n \times n$, where $T[i][j]$ records the smallest number of hops between term i and j.

[illegible] Consider the term graph shown in Figure 3(a). Its distance matrix is shown in Figure 3(b).



(a) The Term Graph

$$\begin{bmatrix} 0 & 2 & 1 & 2 & 3 \\ 2 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 2 \\ 2 & 1 & 1 & 0 & 1 \\ 3 & 1 & 2 & 1 & 0 \end{bmatrix}$$

(b) The Corresponding Distance Matrix

Fig. 3. An Example Term Graph and Its Distance Matrix

Distance-Weighted Similarity. We propose to use the [illegible] to characterize the similarity of a document and a category. Intuitively, if a document D is similar to a category C, there will be many pairs of terms that occur in both the document and the category; in addition, the distance between terms in those pairs will not be too large. We show the algorithm in Algorithm 1. α is a parameter to adjust the effect of the distance of the terms to the similarity score. We set $\alpha = 2$ in our experiments.

Algorithm 1 Distance-Weighted-Similarity(T, D, α)

```
Input:
    $T$ is the distance matrix for the category $C$.
Description:
  $n = 0$
  $w = 0$
  for all pair of terms, $(u, v)$ , in $D$ do
      $w = w + (T[u][v])^\alpha$
      $n = n + 1$
  end for
  return $\frac{n}{w}$
```

Classification. We adopt the following simple classification method based on the distance matrix and the distance-weight similarity function. Given the set of distance matrixes $\{T_1, T_2, \ldots, T_n\}$ representing the categories $\{C_1, C_2, \ldots, C_n\}$ and a testing document D. The document will be classified to category C_i if and only if the distance-weighted similarity of C_i and D is the largest among all the categories.

5 Experimental Evaluation

In this section, we present some preliminary experiment results using classifiers build on our term graph model. We note that our current focus is to more

Table 1. Statistics of the Categories

Category	Training Set	Testing Set	*minsup*
acq	1488	643	12
corn	159	48	6
crude	349	161	10
earn	2709	1044	15
grain	394	134	8
interest	289	100	12
money-fx	460	141	20
ship	191	85	7
trade	337	112	14
wheat	198	66	6

on gaining more insight into the term graph model and exploring its potential applications in text mining.

We have implement our two classification methods in Java. We choose to use the SVM classifier with linear kernel for comparison, as it is one of the most best text classifier[8]. We build our SVM classifier built using TF-IDF weighting scheme on top of the `libsvm` library [4].

The text collections we used in the experiments are the Reuters-21578 repository, which is the standard collections in many previous studies on text categorization. There exist several modes of splitting the Reuters-21578 text collections into the training and testing parts. For comparison purpose, we choose to use "ModApte" split, which produces 9603 documents for the training set and 3299 documents for the testing set. We also follow the popular approach to choose to use only top-10 categories with the most number of training documents in the experiment [18]. We list the category names, the corresponding numbers of training and testing documents, and the minimum support thresholds in Table 1. The minimum support thresholds are set empirically such that the size of the term graphs for the categories are of similar size.

We measure the adjusted accuracy of different classifiers on the testing documents. The adjustment is necessary because a Reuters document may belong to multiple categories. We regard it as a correct classification as long as the predicted class label match one of the class labels of the testing document.

5.1 Visualization of the Graph Model

We show the plot of an example term graph (with PageRank scores) for a medical text document collections in Figure 4. We observe that many important notions have been captured in the figure, such as "patient" with "disease".

5.2 Experiments Using the Rank-Based Classification

We performed experiments using classification methods based on the notion of rank correlation. We list the results of each of the four heuristics in Table 2.

As shown in the tables above, method of using Union Set is most competitive: it has similar accuracy with SVM for four out of ten categories. Specifically, it

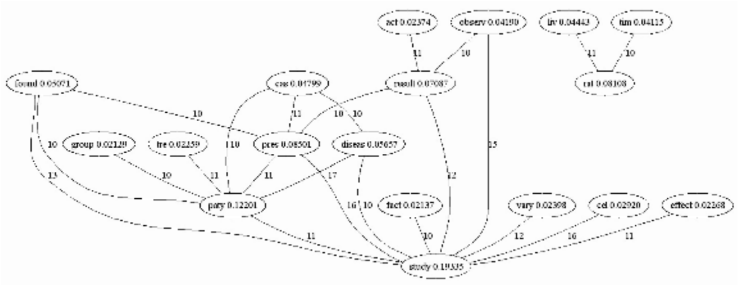


Fig. 4. Visualization of the Term Graph

Table 2. Experiments Using the Rank-based Classification

Category	Union Set	Bigger Set	Smaller Set	Intersection	SVM
acq	98.4	80.9	34.5	20.7	95.6
corn	93.7	70.8	62.5	31.2	93.8
crude	83.2	70.8	65.2	35.4	90.1
earn	95.4	95.3	64.9	27.0	98.8
grain	90.2	51.5	43.2	20.1	94.8
interest	57.0	70.0	55.0	40.0	83.0
money-fx	51.8	75.9	57.4	52.5	90.8
ship	58.8	65.9	58.8	35.3	84.7
trade	80.3	70.5	49.1	27.7	89.3
wheat	96.9	53.0	50.0	21.1	100.0

significantly outperforms SVM for acq category. Considering the simplicity of our classifier, these results are rather encouraging.

We can also observe that the methods of using Intersection Set or Smaller Set do not produce good results. The problem for these methods is that the input vectors are not able to represent the category. There are a large number of words used to present the whole category. For example, the acq category has 776 words, the earn category has 527 words. Using Intersection or Smaller sets means that we use only a small portion of those when we calculate the correlation between the category and the testing document. Therefore, the accuracy is very low. The accuracy of Bigger and Union sets are better for some categories.

5.3 Experiment with Distance Score Approach

We experimented with the classifier based on the distance-weighted similarity function and list the results in Table 3.

We can observe the similar trend between the results in this experiment and the Bigger Set and Union Set methods above. earn, acq have the highest precision points. The system does not perform well for overlapped categories such as grain, corn and wheat because those categories have lots of words in common. Unfortunately, those words also appear frequently in the testing documents and

Table 3. Experiments Using the Distance-Weighted Classification

Category	Distance-Weighted	SVM
acq	87.9	95.6
corn	60.4	93.8
crude	52.8	90.1
earn	88.3	98.8
grain	54.5	94.8
interest	68.0	83.0
money-fx	80.1	90.8
ship	22.4	84.7
trade	73.2	89.3
wheat	63.6	100.0

the distance model cannot clearly discriminate between the categories to which the documents should belong.

6 Conclusions

In this work, we introduce a new term graph model to capture more information for text document and present preliminary results on its potential application in text mining. The new model is capable of capturing the term co-occurrence information among terms. We explored ideas of using novel similarity functions, the rank correlation coefficient and the distance-weighted similarity function, both based on the new model.

There are many area our methods can be improved. As one of our future work, we are actively exploring new features based on our term graph model. Another promising direction is to use our model to complement existing classification method.

References

1. M. Antoine and O.R. Zaiane. Classifying text documents by associating terms with text categories. In *Proceedings of the 13th Australasian conference on database technologies*, volume 5, pages 215–222, Melbourne, Australia, 2002.
2. F. Beil, M. Ester, and X. Xu. Frequent term-based text clustering. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining*, 2002.
3. S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, volume 30, pages 107–117, 1998.
4. C. C. Chang and C. J. Lin. LIBSVM: A library for support vector machine, 2001. At `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`.
5. C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, September 1995.
6. B. C. M. Fung, K. Wang, and M. Ester. Hierarchical document clustering using frequent itemsets. In *Proceedings of the SIAM International Conference on Data Mining*, 2003.

7. P. Jackson and I. Moulinier. *Natural Language Processing for Online Applications: Text Retrieval, Extraction and Categorization.* John Benjamins Publishing Company, Amsterdam/Philadenphia, 2002.
8. T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Machine Learning: ECML- 98, 10th European Conference on Machine Learning, 1998*, pages 137–142, 1998.
9. B. Liu, C. W. Chin, and H. T. Ng. Mining topic-specific concepts and definitions on the web. In *Proceedings of the 12th International Conference on World Wide Web*, pages 251–260, 2003.
10. Guimei Liu, Hongjun Lu, Jeffrey Xu Yu, Wei Wang, and Xiangye Xiao. AFOPT: An efficient implemetation of pattern growth approach. In *Workshop on Frequent Itemset Mining Implementations*, Melbourne, Florida, USA, November 2003.
11. Chris D. Paice. Another stemmer. *SIGIR Forum*, 24(3):56–61, 1990. `http://www.comp.lancs.ac.uk/computing/research/stemming/`.
12. William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C: The Art of Scientific Computing.* Cambridge University Press, 2nd edition edition, 1992. ISBN 0-521-43108-5.
13. G. Salton and M. J. McGill. *An Introduction to Modern Information Retrieval.* McGraw-Hill, 1983.
14. F. Sebastiani. Machine learning in automated text categorization. Technical Report Technical Report IEI-B4-31-1999, Consiglio Nazionale delle Ricerche, Pisa, Italy, 1999.
15. Fabrizio Sebastiani. Machine learning in automated text categorization. *ACM Computing Survey*, 34(1):1–47, 2002.
16. K. Wang, C. Xu, and B. Liu. Clustering transactions using large items. In *CIKM 1999*, 1999.
17. Y. Yang. An evaluation of statistical approaches to text categorization. Technical Report Technical Report CMU-CS-97-127, Carnegie Mellon University, April 1997.
18. Y. Yang and X. Liu. A re-examination of text categorization methods. In *22nd Annual International SIGIR-99*, pages 42–49, Berkley, August 1999.

A Latent Usage Approach for Clustering Web Transaction and Building User Profile

Yanchun Zhang[1], Guandong Xu[1], and Xiaofang Zhou[2]

[1] School of Computer Science and Mathematics,
Victoria University, PO Box 14428, VIC 8001, Australia
{xu, yzhang}@csm.vu.edu.au
[2] School of Information Technology & Electrical Engineering,
University of Queensland, Brisbane QLD 4072, Australia
zxf@itee.uq.edu.au

Abstract. Web transaction data between web visitors and web functionalities usually convey users' task-oriented behavior patterns. Clustering web transactions, thus, may capture such informative knowledge, in turn, build user profiles, which are associated with different navigational patterns. For some advanced web applications, such as web recommendation or personalization, the aforementioned work is crucial to make web users get their preferred information accurately. On the other hand, the conventional web usage mining techniques for clustering web objects often perform clustering on usage data directly rather than take the underlying semantic relationships among the web objects into account. *Latent Semantic Analysis* (LSA) model is a commonly used approach for capturing semantic associations among co-occurrence observations.. In this paper, we propose a LSA-based approach for such purpose. We demonstrated usability and scalability of the proposed approach through performing experiments on two real world datasets. The experimental results have validated the method's effectiveness in comparison with some previous studies.

1 Introduction

With the popularizing and spreading of Internet application, Web has recently become a powerful platform for, not only retrieving information, but also discovering knowledge, from web data repository. Generally, web users may exhibit various types of behaviors associated with their information needs and intended tasks when they are traversing the Web. These task-oriented behaviors are explicitly characterized by sequences of clicks on different web items performed by users. As a result, these tasks are implicitly captured by inducing the underlying relationships among the click-stream data. For example, image a web site designed for information about automobiles; there will be a variety of customer groups with various access interests during their visiting such an E-commerce website. One type of customers intends to make comparison prior to shopping,; a visitor planning to purchase particular type car of wagon, for example, would have to browse the web pages of each manufacturer, compare their offering,, whereas another one will just be more interested in one spe-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 31 – 42, 2005.

cific brand car, such as "Ford", rather than one specific car category. In such circumstance, these two visitors with different interests may follow distinct access tracks to accomplish their goals and corresponding clickstream data are recorded in web sever log file as well. As a result, mining web log information may reveal user access patterns. Moreover, the discovered informative knowledge (or pattern) will be utilized for providing better web applications or web services, such as web recommendation or personalization.

Generally, web mining techniques can be defined as those methods to extract so-called "nuggets" (or knowledge) from web data repository, such as content, linkage, usage information, by utilizing data mining tools. Among such web data, user clickstream, i.e. usage data, can be mainly utilized to capture users' navigational patterns and identify user intended tasks. Once the user navigational behaviors are effectively characterized, they will provide benefits for further web applications, in turn, facilitate and improve web service quality for both web-based organizations and for end users. As a result, web usage mining recently has become one more active and hotter topic, and a variety of research communities from database management, artificial intelligence and information systems etc., have addressed this topic and achieved great success as well [1-7]. Meanwhile, with the benefits of great progress in data mining research, many data mining techniques, such as clustering[3, 8, 9] association rule mining [10, 11] and sequential pattern mining [12] are adopted widely to improve the usability and scalability of web mining.

Related work: In general, there are two types of clustering methods performed on the usage data: user transaction clustering and web page clustering [13]. One successful application of web page clustering is adaptive web site. For example, the algorithm called PageGather [3, 14] is proposed to synthesize index pages that are not existing initially, based on partitioning web pages into various groups. The generated index page is conceptually representing the various access interests of users according to their navigational history. Another example is that clustering user rating results has been successfully adopted in collaborative filtering application as a data preparing step to improve the scalability of recommendation using *K-Nearest-Neighbor* (*K*NN) algorithm [15]. Mobasher et al. [9] utilize user transaction and pageview clustering techniques, which is employing traditional *k*-means clustering algorithm to characterize user access pattern for web personalization based on mining web usage data. These proposed clustering-based techniques have been proven to be efficient from their experimental results since they are really capable of identifying the intrinsic common attributes revealed from their recently historic clickstream data. Generally, these usage patterns are explicitly captured at the level of user transaction or pageview. They, however, do not reveal the underlying characteristics of user navigational activities as well as web pageview. For example, such discovered usage patterns provide little information of why such web transactions or web pages are partitioning together, and latent relationships among the co-occurrence observation data have not been incorporate into the mining process as well. Thus, it is needed to develop LSA-based approaches that can reveal not only common trends explicitly, but also take the latent information into account implicitly during mining. In [16], an algorithm based on *Principal Factor Analysis* (PFA) model derived from statistical analysis, is proposed to generate user access pattern and uncover latent factor by clustering user transactions and analyzing principal factor involved in web usage mining. Analogous, some works [17-19] are addressed to derive user access patterns and web page segments from various types of web data, by utilizing a so-called *Probabilistic Semantic Latent Analysis* (PLSA) model, which is based on maximum likelihood principle from statistics.

Our approach: In this paper, we address these issues by proposing another alternative LSA-based approach for clustering web transaction and generating user profile. After data preprocessing, we produce a user transaction collection and a pageview corpus via user and pageview identification process respectively, in turn, construct the session-pageview matrix as usage data, in which each cell is expressed by a weight re presenting the contribution made by a specific pageview during one user transaction. In this manner, we could map the relationships among the co-occurrence observations (i.e. user transactions) into a high-dimensional space. Moreover, an improved LSA-based clustering algorithm, named latent usage information (LUI), is proposed to find out user segments with similar behaviors effectively and precisely from aforementioned usage data by using linear algebra theory, especially single value decomposition of matrix due to revealing deeper relationships among web transactions. The discovered user clusters are exploited to generate a variety of goal-oriented user profiles by calculating the centroid of corresponding cluster in the form of weighted pageview set. Experiments are conducted on two real world datasets to validate the usability and scalability of usage mining. Meanwhile, an evaluation metric is adopted to assess the quality of discovered clusters, and comparisons are made with some previous work as well. The experimental results have shown that the proposed approach is capable of effectively discovering user access pattern and revealing the underlying relationships among user visiting records.

The remainder of paper is organized as follows: in section 2, we briefly discuss how to construct session-pageview matrix during data preparation. Section 3 gives the latent usage information (LUI) algorithm. Since the LUI algorithm is based on linear algebra theory, especially the Single Value Decomposition (SVD) of a matrix, some basic background knowledge of SVD is provided in this section as well. In section 4, some experimental results derived on real world datasets are presented and comparisons with previous study are discussed as well. Finally, we conclude and give future works in section 5.

2 Latent Usage Information (LUI) Model

We start with collecting the raw web sever logs of the site and perform data cleaning, pageview identification, and user identification such data preparation measures to construct the co-occurrence observation. More detailed introduction of data preparation steps could be found in [20]. At this stage, we briefly introduce how to build up the session-pageview matrix for web usage mining

2.1 Usage Data Identification

Basically, according to W3C definition, a web pageview can be viewed as a visual rendering of a web page. In this way, the user access interest exhibited may be reflected by the varying degree of visits in different web pages during one session.

Thus, we can represent a user session as a collection of transactions, which includes a series of weighted pageviews, during the visiting period. In other words, the user session can be expressed in the form of pageview vectors. From such viewing point, we generate the following user session expression. Given *n* web pages in a web site and *m* web users visiting the web site during a period of time, after appropriate data preprocessing such as page identification and user sessionization, we built up the pageview corpus as $P = \{p_1,p_2,...,p_n\}$, and user session collection as $S = \{s_1,s_2,...,s_m\}$. Conceptually, modeling of user session in a collection of pages defined by the so-called web page corpus, which consists of all web items visited by whole users, is similar to modeling a document in terms of word frequencies by using a word dictionary in text IR. In short, each user session can be, in turn, expressed as a set of weight-pageview pairs, $s_i=\{<p_1,a_{i1}>,<p_2,a_{i2}>,...<p_n,a_{in}>\}$. By simplifying the above expression in the form of pageview vector, each user session can be considered as an *n*-dimensional vector $s_i=\{a_{i1},a_{i2},...,a_{in}\}$, where a_{ij} denotes the weight for pageview p_j in s_i user session. As a result, the whole user session data can be utilized to form web usage data represented by a session-pageview matrix $SP_{m\times n}=\{a_{ij}\}$ (Figure 1 illustrates the skeletal structure of session-page matrix).

The cell value in the session-page matrix, a_{ij}, can be represented by a weight associated with the contribution of page p_j in the user session s_i, which is usually determined by the number of hit or the amount time spent by specific user on the corresponding page. Generally, in order to eliminate the influence caused by the relative amount difference of visiting time duration or hit number, the normalization manipulation across pageviews space in same user session is performed. Figure 2 illustrates

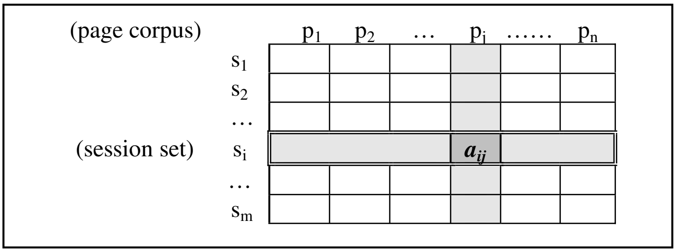


Fig. 1. Skeletal structure of session-pageview matrix

```
202.161.108.167 - - [01/Feb/2003:00:00:03 +1100]
"GET/timetables/city/2003s1/cc 4logo.gif HTTP/1.1" 206
14102 "http://www.cs.rmit.edu.au/timetables/city/2003s1/
cover.html "Mozilla/4.0 (compatible; MSIE 5.5; Windows 98)"

213.183.13.65 - - [01/Feb/2003:00:00:16 +1100]
"GET/~winikoff/palm/dev.html HTT P/1.1" 302 244
"http://www.google.de/search?q=sources+onboardc+examples&ie
=UTF-8&oe=UTF-8&hl=de&meta=" Scooter/3.3"
```

Fig. 2. Sample of web access log

two items retrieved from a web access log, in which each field is separated by space. Particularly, note that the first and fourth fields are identified as the visitor IP address and the requested URL respectively, and utilized to assist usage data collection.

2.2 Latent Usage Information Algorithm

Once usage matrix is constructed, we may applying conventional clustering on user transaction data to classify user sessions into various groups, within which the classified sessions share both common access interest exhibited from their visiting records. It is intuitive to perform clustering algorithm directly on each row vector of usage matrix to determine the relative "close" session cluster by using similarity-based measure, such as commonly adopted cosine similarity from Information Retrieval. In [9], an algorithm named PACT is proposed based on the above discussed technique. However, this kind of clustering technique only capture the mutual relationships between session data explicitly, it is incapable of revealing the "deeper" underlying characteristics of usage pattern. In this work, we propose the latent usage information (LUI) algorithm to group user sessions semantically through taking latent information into account. For better understanding the LUI algorithm, we first discuss some theoretical background of the SVD algorithm.

- **Single Value Decomposition Algorithm**

The SVD definition of a matrix is illustrated as follows[21]: For a real matrix $A=[a_{ij}]_{m\times n}$, without loss of generality, suppose $m\geq n$ and there exists SVD of A:

$$A = U_{m\times m} \sum_{m\times n} V_{n\times n}$$

where U and V are orthogonal matrices. Matrices U and V can be respectively denoted as $U_{m\times m}=[u_1,u_2,\ldots,u_m]_{m\times m}$ and $V_{n\times n}=[v_1,v_2,\ldots,v_n]_{n\times n}$, where u_i $(i=1,\ldots,m)$ is a m-dimensional vector $u_i=(u_{1i},u_{2i},\ldots,u_{mi})^T$ and v_j $(j=1,\ldots,n)$ is a n-dimensional matrix $v_j=(v_{1j},v_{2j},\ldots,v_{nj})^T$. Suppose rank(A)=r and single values of A are diagonal elements of $\sum$ as follows:

$$\sigma_1 \geq \sigma_2 \geq \cdots \sigma_r \geq \sigma_{r+1} = \cdots = \sigma_n = 0 \cdot$$

For a given threshold ε ($0<\varepsilon\leq 1$), we choose a parameter k such that $(\sigma_k-\sigma_{k+1})/\sigma_k\geq \varepsilon$. Then, we denote $U_k=[u_1,u_2,\ldots,u_k]_{m\times k}$, $V_k=[v_1,v_2,\ldots,v_k]_{n\times k}$, $\sum_k=diag(\sigma_1,\sigma_2,\ldots\sigma_k)$, and $A_k=U_k\sum_k V_k$

As known from the theorem in algebra [21], A_k is the best approximation matrix to A and conveys main and latent information among the usage data. This property makes it possible to find out relative "close" user session at the semantic latent level based on their mutual similarity.

- **Representation of User Transaction in Latent Space**

Once SVD implementation is completed, we may rewrite user sessions with the obtained approximation matrix U_k, $\sum_k$ and V_k and map them into another k-dimensional latent space. For a given session, it is represented as a coordinate vector with respect to pageviews: $s_i = \{a_{i1},a_{i2},\ldots,a_{in}\}$. The projection of coordinate vector s_i in the k-dimensional latent subspace is reparameterize as

$$s_i^{'} = s_i V_k \sum{}_k = (t_{i1}, t_{i2}, ..., t_{ik}) \quad (1)$$

where $t_{ij} = \sum_{k=1}^{n} a_{ik} v_{kj} \sigma_j$, $j = 1,2,\dots,k$ ·

- **Similarity Measure**

We adopt traditional Cosine similarity to capture common interests shared by user sessions, i.e. for two vectors $x=(x_1,x_2,\dots,x_k)$ and $y=(y_1,y_2,\dots,y_k)$ in k-dimensional space, the similarity between them is defined as

$$sim(x, y) = (x \bullet y) / (\|x\|_2 \|y\|_2), \text{ where } x \bullet y = \sum_{i=1}^{k} x_i y_i \text{ , } \|x\|_2 = \sqrt{\sum_{i=1}^{k} x_i^2} \text{ .}$$

In this manner, the similarity between two user sessions is defined as:

$$sim(s_i^{'}, s_j^{'}) = \frac{(s_i^{'} \bullet s_j^{'})}{\|s_i^{'}\|_2 \|s_j^{'}\|_2} \quad (2)$$

3 Constructing User Profile Based on Latent Usage Information

In this section, we present the algorithms for clustering web transaction and generating user profile based on the discovered clusters as well.

3.1 Clustering Web Transaction

Here we adopt a modified standard K-means clustering algorithm, named MK-means clustering, to classify user session based on the transformed SP matrix over the latent k-dimensional space. This algorithm does not need to predefine value k and k initial centroids, whereas the standard k-means has to do so to start clustering. The algorithm is described as follows:

```
Algorithm: MK-means clustering
Input: usage data SP' and similarity threshold $\varepsilon$
1. Choose the first user session $s_1'$ as the initial cluster $C_1$ and centroid of this
   cluster, i.e. $C_1=\{s_1'\}$ and $Cid_1=s_1'$.
2. For each session $s_i'$, calculate the similarity between $s_i'$ and the centroids of
   other existing cluster $sim(s_i',Cid_j)$.
3. if $sim(s_i^{'}, Cid_k) = \sum_j \max(sim(s_i^{'}, Cid_j)) > \varepsilon$ , then allocate $s_i'$ into $C_k$ and
   recalculate the centroid of cluster $C_k$ as $Cid_k = 1/|C_k| \sum_{j \in C_k} s_j^{'}$ ;
4. Otherwise, let $s_i'$ itself construct a new cluster and be the centroid of this
   cluster.
5. Repeat step 2 to 4 until all user sessions are processed and all centroids do
   not update any more.
Output: cluster set $CS=\{C_k\}$
```

3.2 Building User Profile

As we mentioned above, each user session is represented as a weight-based pageview vector. In this may, it is reasonable to derive the centroid of cluster obtained by aforementioned algorithm as the user profile. In this work, we compute the mean vector to represent the centroid. For each session cluster $C_k \in CS$, the value for each pageview in the mean vector is determined by the ratio of the sum of pageview weights in C_k to the number of sessions in the cluster. In order to eliminate the diversity in visiting quantity of each session, the weights are normalized in calculating the centroid of cluster. Thus, the maximum weight in user profile is updated to be 1, whereas other weights are divided by the maximum weight across session. Meanwhile, some low-contribution pageviews (i.e. those with mean weights below one certain limit) are filtered out. The algorithm for constructing user profile is as follows:

1. For each pageview in cluster, we compute the mean value of pageview as

$$wt(p, pf) = 1/|C_k| \sum_{s \in C_k} w(p, s) \tag{3}$$

   where $w(p,s)$ is the weight of pageview p in session $s \in C_k$.
2. For each cluster, furthermore, we construct its mean vector(i.e. centroid) as

$$mv_C = \{< p, wt(p, pf) > | p \in P\} \tag{4}$$

3. For each pageview weight within user profile, if the value < threshold μ, the corresponding item will be removed from the vector with it weight, otherwise keep it leave.
4. Sort the pageviews with their weights in descending order and output the mean vector as user profile.

$$pf_{c_k} = \{< p_{1k}, wt(p_{1k}, pf) >, < p_{2k}, wt(p_{2k}, pf) > ..., < p_{tk}, wt(p_{tk}, pf)\} \tag{5}$$

where

$$wt(p_{1k}, pf) > wt(p_{2k}, pf) > \cdots > wt(p_{tk}, pf) > \mu, PF = \{pf_{c_k}\}, k = 1, 2 \cdots, t.$$

4 Experiment and Evaluation

In order to evaluate the effectiveness of the proposed LUI-based clustering algorithm and user profile generating algorithm, and explore the discovered user access pattern, we conducted preliminary experiments on two real world data sets. Some comparisons with previous work are made as well.

4.1 Data Sets

The first dataset used is downloaded from KDDCUP (www.ecn.purdue.edu/ kddcup/). The data set is common-used data resource provided to test and compare methods (prediction algorithm, clustering approaches, etc.) for data mining purpose. Data preprocessing is needed to perform on the raw data set since there are some short user sessions existing in the data set, which mean they are of less contribution for data mining. Support filtering technique is used to eliminate these user sessions, leaving only sessions with at least four pages. After data preparation, we have setup an data set including 9308 user sessions and 69 pages, where every session consists of 11.88 pages in average. We refer this data set to “KDDCUP data”. In this data set, the entries in session-page matrix associated with the specific page in the given session are determined by the numbers of web page hits by the given user.

The second data set is from a university website log files and was made available by the author of [13]. The data is based on a random collection of users visiting this site for a 2-week period during April of 2002. After data preprocessing, the filtered data contains 13745 sessions and 683 pages. This data file is expressed as a session-page matrix where each column is a page and each row is a session represented as a vector. The entries in the table correspond to the amount of time (in seconds) spent on pages during a given session. For convenience, we refer this data as “CTI data”. For each dataset, we randomly choose 1000 transaction as the evaluation set, whereas the remainder part is selected as the training set for constructing user profiles.

4.2 Results of Generated User Profiles

We utilize aforementioned LUI method to classify user transactions. For comparison purpose, we also perform PACT approach based on standard K-means used in [9] to generate user profiles. From the results, it is found that generated profiles are “overlapping” of pageviews since some pageviews are listed in more than one user clusters. Table 1 depicts 2 user profiles generated from KDD dataset using LUI approach. Each user profile is listed in a ordered pageviews’ sequence with weights, which means the greater weight of a pageview contribute, the more likely it is to be visited. The first profile in Table 1 represents the activities involved in online-shopping circumstance such as login, shopping_cart, checkout etc., especially occurring in purchasing leg-wear products, whereas second user profile reflects customers’ concern focused on the interests with regard to the department store itself.

Table 1. Examples of generated user profiles from KDD dataset

Pageview #	Pageview content	weight
29	Main-shopping_cart	1.00
4	Products-productDetailleagwear	0.86
27	Main-Login2	0.67
8	Main-home	0.53
44	Check-expressCheckout	0.38
65	Main-welcome	0.33
32	Main-registration	0.32
45	Checkout-confirm_order	0.26

Pageview #	Pageview content	weight
11	Main-vendor2	1.00
8	Main-home	0.40
12	Articles-dpt_about	0.34
13	Articles-dpt_about_mgmtteam	0.15
14	Articles-dpt_about_broadofdirectors	0.11

Analogously, some informative finding can be obtained in Table 2, which is derived from CTI dataset. In this table, three profiles are generated: the first one reflects the main topic of international student concerning issues regarding applying for admission, and second one involves in the online applying process for graduation, whereas the final one indicates the most common activities happened during students browsing the university website, especially while they are determining course selection, i.e. selecting course, searching syllabus list, and then going through specific syllabus.

Table 2. Examples of generated user profiles from CTI dataset

Pageview #	Pageview content	weight
19	Admissions-requirement	1.00
3	Admissions-costs	0.41
15	Admissions-intrnational	0.24
13	Admissions-I20visa	0.21
387	Homepage	0.11
0	Admission	0.11

Pageview #	Pageview content	weight
349	Gradapp-tologin	1.00
20	Admissions-statuscheck	0.35
340	Gradapp-login	0.32
333	Gradapp-appstat_shell	0.13
0	Admissions	0.11

Pageview #	Pageview content	weight
387	Homepage	1.00
59	Courses	0.78
71	Course-syllabilist	0.40
661	Program-course	0.17
72	Course-syllabisearch	0.12

4.3 Evaluation of User Transaction Clusters

In order to evaluate the quality of clusters derived from LUI approach, we adopt one specific metric, named the *Weighted Average Visit Percentage* (WAVP)[9]. This evaluation method is based on assessing each user profile individually according to the likelihood that a user session which contains any pageviews in the transaction cluster will include the rest pageviews in the cluster during the same session. The principle of WAVP metric is discussed as follows: suppose T is one of transaction set within the evaluation set, and for s specific cluster C, let T_c denote a subset of T whose elements contain at least one pageview from C. Moreover, the weighted average visit percentage of T_c may conceptually be determined by the similarity between T_c and the cluster C if we consider the T_c and C as in the form of pageview vector. As a result, the WAVP is computed as:

$$WAVP = \left(\sum_{t \in T_c} \frac{\vec{t} \bullet \vec{C}}{|T_c|} \right) \Big/ \left(\sum_{p \in Pf} wt(p, pf) \right) \qquad (6)$$

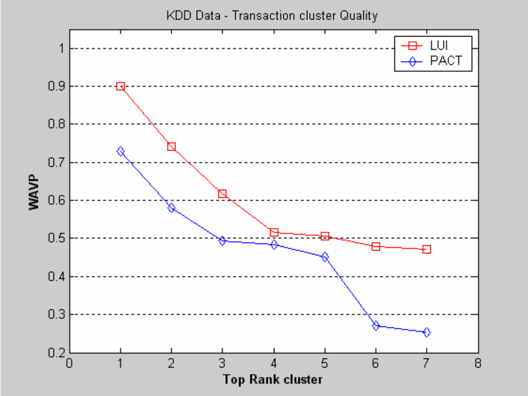


Fig. 3. User cluster quality analysis results upon WAVP comparison for KDD dataset

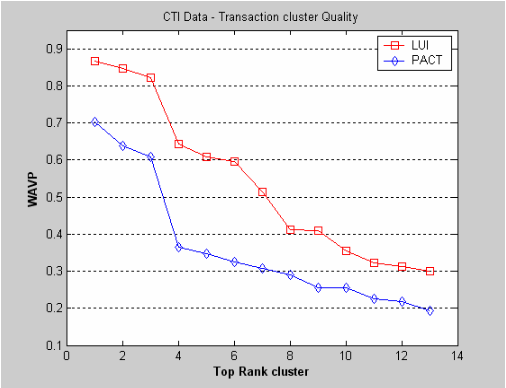


Fig. 4. User cluster quality analysis results upon WAVP comparison for CTI dataset

From the definition of WAVP, it is known that the higher WAVP value is, the better quality of obtained transaction cluster possesses.

As mentioned above, for comparison purpose, we conduct data simulations upon two real world datasets by using two approaches. Figure 3 and Figure 4 depict the comparison results of WAVP values for KDD and CTI datasets with PACT respectively. In each figure, the obtained user profiles are arrayed in the descending rank according to their WAVP values, which reflect the quality of various clustering algorithms. From these two curve lines, it is easily concluded that the proposed LUI-based technique overweighs standard K-means based algorithm in term of WAVP parameter. Moreover, LUI approach is capable of capturing the latent relationships among user transaction and discovering user profiles representing the actual navigational behaviors more effectively and accurately.

5 Conclusion and Future Work

In this paper, we proposed a LSA-based approach, named LUI, for grouping web transaction and generating user profile. Firstly, we mapped the relationships among the co-occurrence observations (i.e. user transactions) into a high-dimensional space to construct the usage data in the form of session-pageview matrix. Then, an dimension reducing algorithm (i.e. single value decomposition) was employed on the usage matrix to capture the latent usage information for partitioning user transaction. Based on the decomposed latent usage information, we proposed a modified *k*-means clustering algorithm to generate user session clusters. Moreover, the discovered user groups are utilized to construct user profiles expressed in the form of a weighted pageview collection, which represents the common usage pattern associated with one kind of specific visitors' access interests. The constructed user profiles corresponding to various task-oriented behaviors are represented as a set of pageview-weight pairs' collection, in which each weight reflects the significance contributed by the page. Experiments are conducted on two real world datasets to validate the usability and scalability of usage mining. Meanwhile, an evaluation metric is adopted to assess the quality of discovered clusters, and comparisons are made with some previous works as well. The experimental results have shown that the proposed approach is capable of effectively discovering user access pattern and revealing the underlying relationships among user visiting records as well.

The future works will be focused on the research issues, such as performing experiments over more datasets, broadening comparison and make use of discovered user profiles for further web application, for example, web recommendation and personalization.

Acknowledgement

This research has been partly supported through ARC Discovery Project Grant (DP0345710) and National Natural Science Foundation of China (No 60403002).

References

1. Joachims, T., D. Freitag, and T. Mitchell. Webwatcher: A tour guide for the *world wide web*. in *The 15th International Joint Conference on Artificial Intelligence (ICJAI'97)*. 1997. Nagoya, Japan.
2. Lieberman, H. *Letizia: An agent that assists web browsing*. in *Proc. of the 1995 International Joint Conference on Artificial Intelligence*. 1995. Montreal, Canada: Morgan Kaufmann.
3. Perkowitz, M. and O. Etzioni. *Adaptive Web Sites: Automatically Synthesizing Web Pages.* in *Proceedings of the 15th National Conference on Artificial Intelligence*. 1998. Madison, WI: AAAI.
4. Ngu, D.S.W. and X. Wu. *Sitehelper: A localized agent that helps incremental exploration of the world wide web*. in *Proceedings of 6th International World Wide Web Conference*. 1997. Santa Clara, CA: ACM Press.

5. Cohen, E., B. Krishnamurthy, and J. Rexford. *Improving end-to-end performance of the web using server volumes and proxy lters*. in *Proc. of the ACM SIGCOMM '98*. 1998. Vancouver, British Columbia, Canada: ACM Press.
6. Büchner, A.G. and M.D. Mulvenna, *Discovering Internet Marketing Intelligence through Online Analytical Web Usage Mining.* SIGMOD Record, 1998. **27**(4): p. 54-61.
7. Mobasher, B., R. Cooley, and J. Srivastava. *Creating adaptive web sites through usage-based clustering of URLs*. in *Proceedings of the 1999 Workshop on Knowledge and Data Engineering Exchange*. 1999: IEEE Computer Society.
8. Han, E., et al., *Hypergraph Based Clustering in High-Dimensional Data Sets: A Summary of Results.* IEEE Data Engineering Bulletin, 1998. **21**(1): p. 15-22.
9. Mobasher, B., et al., *Discovery and Evaluation of Aggregate Usage Profiles for Web Personalization.* Data Mining and Knowledge Discovery, 2002. **6**(1): p. 61-82.
10. Agarwal, R., C. Aggarwal, and V. Prasad, *A Tree Projection Algorithm for Generation of Frequent Itemsets.* Journal of Parallel and Distributed Computing, 1999. **61**(3): p. 350-371.
11. Agrawal, R. and R. Srikant. *Jorge B. Bocca and Matthias Jarke and Carlo Zaniolo*. in *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB).* 1994. Santiago, Chile: Morgan Kaufmann.
12. Agrawal, R. and R. Srikant. *Mining Sequential Patterns*. in *Proceedings of the International Conference on Data Engineering (ICDE).* 1995. Taipei, Taiwan: IEEE Computer Society Press.
13. Mobasher, B., *Web Usage Mining and Personalization*, in *Practical Handbook of Internet Computing*, M.P. Singh, Editor. 2004, CRC Press.
14. Perkowitz, M. and O. Etzioni, *Adaptive Web sites.* Communications of the ACM, 2000. **43**(8): p. 152 - 158.
15. O'Conner, M. and J. Herlocker. *Clustering Items for Collaborative Filtering*. in *Proceedings of the ACM SIGIR Workshop on Recommender Systems.* 1999. Berkeley, CA: ACM Press.
16. Zhou, Y., X. Jin, and B. Mobasher. *A Recommendation Model Based on Latent Principal Factors in Web Navigation Data*. in *Proceedings of the 3rd International Workshop on Web Dynamics*. 2004. New York: ACM Press.
17. Xu, G., et al. *Discovering User Access Pattern Based on Probabilistic Latent Factor Model*. in *Proceeding of 16th Australasian Database Conference*. 2004. Newcastle, Australia: ACS Inc.
18. Xu, G., Y. Zhang, and X. Zhou. *Using Probabilistic Semantic Latent Analysis for Web Page Grouping*. in *15th International Workshop on Research Issues on Data Engineering: Stream Data Mining and Applications (RIDE-SDMA'2005).* 2005. Tyoko, Japan.
19. Jin, X., Y. Zhou, and B. Mobasher. *A Unified Approach to Personalization Based on Probabilistic Latent Semantic Models of Web Usage and Content*. in *Proceedings of the AAAI 2004 Workshop on Semantic Web Personalization (SWP'04).* 2004. San Jose.
20. Cooley, R., B. Mobasher, and J. Srivastava, *Data Preparation for Mining World Wide Web Browsing Patterns.* Journal of Knowledge and Information Systems, 1999. **1**(1): p. 5-32.
21. Datta, B.N., *Numerical Linear Algebra and Application*. 1995: Brooks/Cole Publishing Company.

Mining Quantitative Association Rules on Overlapped Intervals

Qiang Tong[1,3], Baoping Yan[2], and Yuanchun Zhou[1,3]

[1] Institute of Computing Technology,
Chinese Academy of Sciences, Beijing, China
{tongqiang, yczhou}@sdb.cnic.cn
[2] Computer Network Information Center,
Chinese Academy of Sciences, Beijing, China
ybp@mimi.cnc.ac.cn
[3] Graduate School of the Chinese Academy of Sciences, Beijing, China

Abstract. Mining association rules is an important problem in data mining. Algorithms for mining boolean data have been well studied and documented, but they cannot deal with quantitative and categorical data directly. For quantitative attributes, the general idea is partitioning the domain of a quantitative attribute into intervals, and applying boolean algorithms to the intervals. But, there is a conflict between the minimum support problem and the minimum confidence problem, while existing partitioning methods cannot avoid the conflict. Moreover, we expect the intervals to be meaningful. Clustering in data mining is a discovery process which groups a set of data such that the intracluster similarity is maximized and the intercluster similarity is minimized. The discovered clusters are used to explain the characteristics of the data distribution. The present paper will propose a novel method to find quantitative association rules by clustering the transactions of a database into clusters and projecting the clusters into the domains of the quantitative attributes to form meaningful intervals which may be overlapped. Experimental results show that our approach can efficiently find quantitative association rules, and can find important association rules which may be missed by the previous algorithms.

1 Introduction

Mining association rules is a key data mining problem and has been widely studied [1]. Finding association rules in binary data has been well investigated and documented [2, 3, 4]. Finding association rules in numeric or categorical data is not as easy as in binary data. However, many real world databases contain quantitative attributes and current solutions to this case are so far inadequate.

An association rule is a rule of the form $X \Rightarrow Y$, where X and Y are sets of items. It states that when X occurs in a database so does Y with a certain probability. X is called the antecedent of the rule and Y the consequent. There

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 43–50, 2005.

are two important parameters associated with an association rule: support and confidence. Support describes the importance of the rule, while confidence determines the occurrence probability of the rule. The most difficult part of an association rule mining algorithm is to find the frequent itemsets. The process is affected by the support parameter designated by the user.

A well known application of association rules is in market basket data analysis, which was introduced by Agrawal in 1993 [2]. In the problem of market basket data analysis, the data are boolean, which have values of "1" or "0". The classical association rule mining algorithms are designed for boolean data. However, quantitative and categorical attributes widely exist in current databases. In [5], Srikant and Agrawal proposed an algorithm dealing with quantitative attributes by dividing quantitative attributes into equi-depth intervals and then combining adjacent partitions when necessary. In other words, for a depth d, the first d values of the attribute are placed in one interval, the next d in a second interval, and so on. There are two problems with the current methods of partitioning intervals: MinSup and MinConf [5]. If a quantitative attribute is divided into too many intervals, the support for a single interval can be low. When the support of an interval is below the minimum support, some rules involving the attribute may not be found. This is the minimum support problem. Some rules may have minimum confidence only when a small interval is in the antecedent, and the information loss increases as the interval size becomes larger. This is the minimum confidence problem.

The critical part of mining quantitative association rules is to divide the domains of the quantitative attributes into intervals. There are several classical dividing methods. The equi-width method divides the domain of a quantitative attribute into n intervals, and each interval has the same length. In the equi-depth method, there are equal number of items contained in each interval. The equi-width method and the equi-depth method are so straightforward that the partitions of quantitative attributes may not be meaningful, and cannot deal with the minimum confidence problem. In [5], Srikant and Agrawal introduced a measure of partial completeness which quantified the information lost due to partitioning, and developed an algorithm to partition quantitative attributes. In [6], Miller and Yang pointed out the pitfalls of the equi-depth method, and presented several guiding principles for quantitative attribute partitioning. In selecting intervals or groups of data to consider, they wanted to have a measure of interval quality to reflect the distance among data points. They took the distance among data into account, since they believed that putting closer data together was more meaningful. To achieve this goal, they presented a more general form of an association rule, and used clustering to find subsets that made sense by containing a set of attributes that were close enough. They proposed an algorithm which used birch [7] to find clusters in the quantitative attributes and used the discovered clusters to form "items", then fed the items into the classical boolean algorithm apriori [3]. In their algorithm, clustering was used to determine sets of dense values in a single attribute or over a set of attributes that were to be treated as a whole.

Although Miller and Yang took the distance among data into account and used a clustering method to make the intervals of quantitative attributes more meaningful, they did not take the relations among other attributes into account by clustering a quantitative attribute or a set of quantitative attributes alone. We believe that their technique still falls shot of a desirable goal.

Based on the above analysis, we find that the partitioning method can be further improved. On the one hand, since clustering an attribute or a set of attributes alone is not good enough, we believe that the relations among attributes should be taken into account. We tend to cluster all attributes together, and project the clusters into the domains of the quantitative attributes. On the other hand, the projection of the clusters on a specific attribute can be overlapped. We think this is reasonable. Moreover, this is a good resolution to the conflict between the minimum support problem and the minimum confidence problem. A small interval may result in the minimum support problem, while a large interval may lead to the minimum confidence problem. When several overlapped intervals coexist in the domain of a quantitative attribute, and different intervals are used for different rules, the conflict between the two problems which confuses the quantitative attributes partitioning does not exist. In this paper, we propose an approach which first applys a clustering algorithm to all attributes, and projects the discovered clusters into the domains of all attributes to form intervals (the intervals may be overlapped), then uses a boolean association rule mining algorithm to find association rules.

The rest of the paper is organized as follows. In Section 2, we introduce some definitions of the quantitative association rule mining problem and review the background in brief. In Section 3, we present our approach and our algorithm. In Section 4, we give the experimental results and our analysis. Finally, in Section 5, we give the conclusions and the future work.

2 Problem Description

Now, we give a formal statement of the problem of mining quantitative association rules and introduce some definitions.

Let $I = \{i_1, i_2, \ldots, i_n\}$ be a set of attributes, and R be the set of real numbers, $I_R = X \times R \times R$, that is $I_R = \{(x, l, u) | x \in I, l \in R, u \in R, l \leq x \leq u\}$. A triple $(x, l, u) \in I_R$ denotes either a quantitative attribute x with a value interval $[l, u]$, or a categorical attribute with a value l $(l = u)$. Let D be a set of transactions, where each transaction T is a set of attribute values. $X \subset I_R$, if $\forall(x, l, u) \in X, \exists(x, v) \in T, l \leq v \leq u$, we say transaction T supports X. A quantitative association rule is an implication of the form $X \Rightarrow Y$, where $X \in I_R$, $Y \in I_R$, and $attribute(X) \bigcap attribute(Y) = \emptyset$. If s percent of transactions in D support $X \bigcup Y$, and c percent of transactions which support X also support Y, we say that the association rule has support s and confidence c respectively. The problem of mining quantitative association rules is the process of finding association rules which meet the minimum support and the minimum confidence at a given transaction database which contains quantitative and/or categorical attributes.

Clustering can be considered the most important unsupervised learning technique, which deals with finding a structure in a collection of unlabeled data. A cluster is therefore a collection of objects which are "similar" to each other and are "dissimilar" to the objects belonging to other clusters [8]. In this paper, a cluster is a set of transactions. An important component of a clustering algorithm is the distance measure among data points. In [6], Miller and Yang defined two thresholds on the cluster size and the diameter. First, the diameter of a cluster should be less than a specific value to ensure that the cluster is sufficiently dense. Second, the number of transactions contained in a cluster should be greater than the minimum support to ensure that the cluster is frequent. Since our clustering approach is different, our definition of the diameter of a cluster is also different.

Definition 1. [illegible]

$$d(I_i, I_j) = \sqrt{\sum_{k=1}^{n} (I_{ik} - I_{jk})^2} \tag{1}$$

Definition 2. [illegible] $C = \{I_1, I_2, \ldots, I_m\}$ [illegible] C [illegible]

$$C_g = \frac{1}{m} \sum_{i=1}^{m} I_i \tag{2}$$

Definition 3. [illegible] $C = \{I_1, I_2, \ldots, I_m\}$ [illegible]

$$D_g(C) = \frac{1}{m} \sum_{i=1}^{m} (I_i - C_g)^T (I_i - C_g) \tag{3}$$

Definition 4. [illegible] C [illegible] $|C|$ [illegible] d_0 [illegible] s_0 [illegible] $|C|$ [illegible] $D_g(C)$ [illegible]

$$|C| \geq s_0,\ D_g(C) \leq d_0 \tag{4}$$

[illegible]

3 The Proposed Approach

In this section, we describe our approach of mining quantitative association rules. We divide the problem of mining quantitative association rules into several steps:

1. Map the attributes of the given database to $I_R = I \times R \times R$. For ordered categorical attributes, map the values of the attribute to a set of consecutive integers, such that the order of the attributes is preserved. For unordered categorical attributes, we define the distance between two different attributes as a constant value. For boolean attributes, map the values of the attributes to “0” and “1”. For quantitative attributes, we keep the original values or transform the values to a standard form, such as Z-Score. We adopt various mapping methods to fit the clustering algorithm. For different data sets, we may use different mapping methods.
2. Apply a clustering algorithm to the new database produced by the first step. In the clustering algorithm, by dealing with the transactions as n-dimension vectors, we take all attributes into account. In this paper, we adopt a common clustering algorithm k-means to identify transaction groups that are compact (the distance among transactions within a cluster is small) and isolated (relatively separable from other groups). By clustering all attributes together, the relations among all attributes are considered, and the clusters may be more meaningful. Besides, we also use Definition 4 as the principle for evaluating the quality of the discovered clusters.
3. Project the clusters into the domains of the quantitative attributes. The projections of the clusters will form overlapped intervals. We make an interval $x \in [l, u]$ a new boolean attribute. The two-dimension example of the projection is shown in Figure 1.
4. Mine association rules by using a classical boolean algorithm. Since the quantitative attributes have been booleanized, we can use a boolean algorithm (such as apriori) to find frequent itemsets, and then use the frequent itemsets to generate association rules.

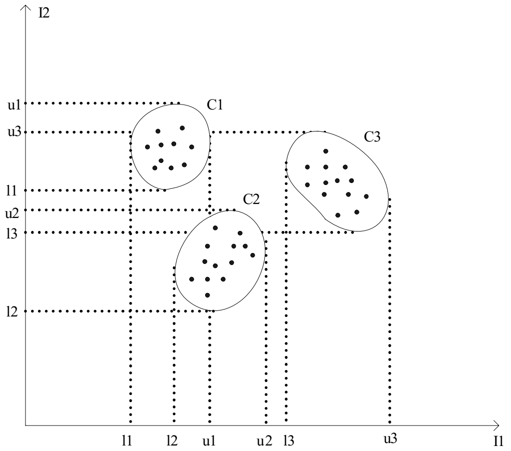


Fig. 1. Projecting the clusters into the domains of quantitative attributes to form intervals, which may be overlapped

4 Experimental Results

Our experimental environment is an IBM Netfinity 5600 server with dual PIII 866 CPUs and 512M memory, which runs Linux operating system. The experiment has been done over a real data set of bodyfat [10]. The attributes in the bodyfat dataset are: density, age, weight, height, neck, chest, abdomen, hip, thigh, knee, ankle, biceps, forearm and wrist. All of the attributes are quantitative attributes. There are 252 records of various people in the dataset. Our purpose is to find association rules over all attributes.

For our algorithm, the parameters needed from the user are the minimum support, the minimum confidence, and the number of clusters. In our experiment, we use minimum support of 10%, minimum confidence of 60%, and the clusters of six. We first use a common clustering algorithm (k-means) to find clusters, then project the clusters into the domains of the quantitative attributes, and finally use a boolean association rule mining algorithm (apriori) to find association rules. Some of the rules which we have found are listed in Figure 2.

From the above rules, we can see that the intervals are overlapped, which cannot be discovered by the previous partitioning methods. The equi-width method cannot divide some quantitative attributes properly (such as density), because the attributes range only in a very small domain, while the equi-depth method may put far apart transactions into the same interval. As shown in Figure 1, our partitioning method projects the clusters into the domains of the quantitative attributes, and forms overlapped intervals. Our method considers both the distance among transactions and the relations among attributes. For previous methods, if an interval is small, it may not meet the minimum support; if an interval is large, it may not meet the minimum confidence. In our method, since the intervals can be overlapped, we can avoid the conflict between the minimum support problem and the minimum confidence problem. Moreover, since our intervals tend to be less than those of the previous methods, the boolean association rule mining algorithm works more efficiently.

ID	Rules
1	Age[40, 74]&Weight[178, 216] ⇒ Abdomen[88.7, 113.1]
2	Age[34, 42]&Weight[195.75, 224.75] ⇒ Chest[99.6, 115.6]
3	Weight[219, 363.15] ⇒ Hip[105.5, 147.7]&Chest[108.3, 136.2]
4	Weight[154, 191]&Height[65.5, 77.5] ⇒ Density[1.025, 1.09]
5	Abdomen[88.6, 111.2]&Hip[101.8, 115.5] ⇒ Weight[196,224]
6	Biceps[24.8, 38.5] & Forearm[22, 34.9] ⇒ Wrist[15.8, 18.5]
7	Thigh[54.7, 69]&Knee[34.2, 42.2] ⇒ Ankle[21.4, 33.9]
8	Weight[118.5, 159.75]&Height[64, 73.5] ⇒ Density[1.047, 1.11]

Fig. 2. Some of the rules discovered by our algorithm with the parameters (minimum support = 10%, minimum confidence = 60%, and the number of clusters k = 6)

5 Conclusions and Future Work

In this paper, we have proposed a novel approach to efficiently find quantitative association rules. The critical part of quantitative association rule mining is to partition the domains of quantitative attributes into intervals. The previous algorithms dealt with this problem by dividing the domains of quantitative attributes into equi-depth or equi-width intervals, or using a clustering algorithm on a single attribute (or a set of attributes) alone. They cannot avoid the conflict between the minimum support problem and the minimum confidence problem, and risk missing some important rules. In our approach, we treat a transaction as an n-dimension vector, and apply a common clustering algorithm to the vectors, then project the clusters into the domains of the quantitative attributes to form overlapped intervals. We finally use a classical boolean algorithm to find association rules. Our approach takes the relations and the distances among attributes into account, and can resolve the conflict between the minimum support problem and the minimum confidence problem by allowing intervals to be overlapped. Experimental results show that our approach can efficiently find quantitative association rules, and can find important association rules which may be missed by the previous algorithms.

Since our approach adopts a common clustering algorithm and a classical boolean association rule mining algorithm rather than integrates the two algorithms together, we believe that our approach can be further improved by integrating the clustering algorithm and the association rule mining algorithm tightly in our future work.

Acknowledgement

This work is partially supported by the National Hi-Tech Development 863 Program of China under grant No. 2002AA104240, and the Informatization Project under the Tenth Five-Year Plan of the Chinese Academy of Sciences under grant No. INF105-SDB. We thank Mr. Longshe Huo and Miss Hong Pan for their helpful suggestions.

References

1. Han, J., Kamber, M.: Data Mining Concepts and Techniques. China Machine Press and Morgan Kaufmann Publishers (2001)
2. Agrawal, R., Imielinski, T. and Swami, A.: Mining Association Rules between Sets of Items in Large Databases. In Proc. of the 1993 ACM SIGMOD International Conf. on Management of Data, Washington, D.C., May (1993) 207–216
3. Agrawal, R. and Srikant, R.: Fast Algorithms for Mining Association Rules in Large Databases. In Proc. of 20th International Conf. on Very Large Data Bases, Santiago, Chile, September (1994) 487–499
4. Han, J., Pei, J., Yin, Y. and Mao, R.: Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach. Data Mining and Knowledge Discovery (2004) 8, 53–87

5. Srikant, R. and Agrawal, R.: Mining Quantitative Association Rules in Large Relational Tables. In Proc. of the 1996 ACM SIGMOD International Conf. on Management of Data, Montreal, Canada, June (1996) 1–12
6. Miller, R. J. and Yang, Y.: Association Rules over Interval Data. In Proc. of the 1997 ACM SIGMOD International Conf. on Management of Data, Tucson, Arizona, United States, May (1997) 452–461
7. Zhang, R., Ramakrishnan, R. and Livny, M.: BIRCH: An Efficient Data Clustering Method for Very Large Databases. In Proc. of the 1996 ACM SIGMOD International Conf. on Management of Data, Montreal, Canada, June (1996) 103–114
8. Jain, A. K., Dubes, R. C.: Algorithms for Clustering Data. Prentice Hall, Englewood Cliffs, New Jersey (1988)
9. Kaufman, L. and Rousseeuw, P. J.: Finding Groups in Data: An Introduction to Cluster Analysis. New York: John Wiley and Sons (1990)
10. Bailey, C.: Smart Exercise: Burning Fat, Getting Fit. Houghton-Mifflin Co., Boston (1994) 179–186

An Approach to Mining Local Causal Relationships from Databases

Yang Bo He[1], Zhi Geng[2], and Xun Liang[1]

[1] Institute of Computer Science and Technology,
Peking University, Beijing 100871, China
{heyangbo, liangxun}@icst.pku.edu.cn
[2] School of Mathematical Sciences,
Peking University, Beijing 100871, China
zgeng@math.pku.edu.cn

Abstract. Mining association rules and correlation relationships have been studied in the data mining field for many years. However, the rules mined only indicate association relationships among variables in an interested system. They do not specify the essential underlying mechanism of the system that describe causal relationships. In this paper, we present an approach for mining causal relationships among attributes and propose a potential application in the field of bioinformatics. Based on the theory of causal diagram, we show the properties of our approach.

1 Introduction

Data mining, which is also referred to as knowledge discovery in databases, is a process of nontrivial extraction of implicit, previously unknown and potentially useful information from databases [8]. There are two primary goals for mining data. One is to understand the interested system; the other is to predict the future behavior. In scientific data mining, say the gene microarray data mining, understanding how the genes regulate each other is the crux of the research. While in business data mining, it is crucial to obtain the useful knowledge that can be used to take action to improve the performance of the business.

Three different kinds of knowledge could be used to achieve the purposes of data mining, association, correlation, and causality. In mining association rules, a major concern is to find appropriate definitions of the "interest" for specific applications [1]. There are many kinds of interest measures for different applications, but in the view of statistics, the concept of association is similar to the conditional probability. It is commonly used to describe the knowledge such as "the presence of item A implies the presence of item B". But it does not involve other three rules: "the absence of item A implies the absence of item B", "the presence of item A implies the absence of item B" and "the absence of item A implies the presence of item B". Brin et al. [2] proposed a correlation measure to represent this kind of relationship about item A and item B in their market basket mining research. Correlation mining is an important technology to

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 51–58, 2005.

discover the correlation relationships among variables. But the association rules and correlation relationships are not equivalent to casual relationships, such as whether the event A causes the event B. The knowledge of such causal relationships is very useful for the purposes of understanding and predicting. For example, suppose that we have a database with thousands of gene expressions from microarray data. The network of gene regulation represents causal relationships among genes, and it describes how genes regulate each other. Furthermore the conditional probability of an effect gene given its causal genes can be used to predict the expression level of the effect gene when its causal genes are intervened externally.

Correlation between variables may come from many possibilities, such as the presence of a common causal variable, selection bias of data, and the causation between them. We call the correlation induced by the first two reasons as a spurious correlation. On the other hand, although a correlation relationship may be spurious, it can also contain the causal information about the variables. The information about the independence and dependence relationships among variables can be used to constrain the possible causal relationships among a subset of those variables. For example, if two variables are independent, we can say that there are no causal relationships between them. This kind of algorithms using constrained information to learn causal relationships are called the constraint-based algorithms. All algorithms discussed in this paper are the constraint-based algorithms.

In this paper we consider the problem of mining causal relationships from statistical correlations that can be easily measured from observational databases. Cooper [4] presented a simple constraint-based algorithm, the simple local causal discovery (LCD) algorithm, for learning the causal relationships. Verstein et al. [10] applied it to discover the causal relationships from the market basket data. But the simple LCD algorithm presented by Cooper can discover causal relationships just in the simplest causal system, in which all variables occur sequentially in a single causal chain. In this paper, we will extend this local causal mining approach such that it could be applicable to the general cases and show the properties of those algorithms.

2 Notation of Causation and Its Interpretation

Causal diagram is a powerful tool to represent the causal relationships among a large number of variables. A causal diagram model contains two parts, a directed acyclic graph (DAG) $G = (V, E)$ and its joint distribution P where G represents a causal structure among variables and E is the set of directed edges that represent the causal relationships among variables. We denote the parents of the variable V_i as pa_i. The parent set pa_i of V_i is a set of direct causes of V_i.

For example, the causal diagram G_1 in Fig.1 represents a causal system with five variables. Node W is the father of A and B, so W is the direct cause of A and B, A and B are the direct effects of W. The variable C has two parents, A and B, so A and B are two direct causes of C.

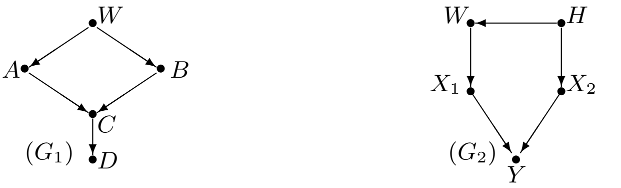


Fig. 1. Two Causal diagrams G_1 and G_2 with five variables. In the diagram G_1, the variable W is a root. In the diagram G_2, the variable W is not the descendent of observational variables X_1, X_2 and Y

In general, we interpret the causal relationships as functions [9]

$$V_i = f_i(pa_i, \varepsilon_i), \tag{1}$$

where ε_i are mutually independent, arbitrarily distributed random disturbances. This kind of child-parent relationship as a function leads a conditional probability between child and its parent, denoted as $P(V_i|pa_i)$. The joint distribution of V could be represented by the product of the conditional probabilities [5, 7] as

$$P(v_1, \cdots v_n) = \prod_i P(v_i|pa_i). \tag{2}$$

For the causal diagram G_1 in Fig. 1, we interpret the causal structure as underlying functions, $W = f_1(\varepsilon_1), A = f_2(W, \varepsilon_2), B = f_3(W, \varepsilon_3), \quad C = f_4(A, B, \varepsilon_4)$ and $D = f_5(C, \varepsilon_5)$. In fact, we do not know the underlying "mechanism", but we can get that the joint distribution $P(W, A, B, C, D)$ satisfies the recursive factorization $P(W, A, B, C, D) = P(W)P(A|W)P(B|W)P(C|A, B)P(D|C)$. The purpose of causal mining is to discover the causal relationships from observational data with the joint distribution $P(W, A, B, C, D)$.

3 The LCD Algorithms and Their Applications

We consider the causal system with discrete variables $V = \{V_1, \cdots, V_l\}$. Suppose that V_i has u_i different values for $i = 1, \cdots, l$. Let $P(v_i, v_j)$ be the joint probability of variables V_i and V_j, n be the sample size denoting the total number of data cases, $n_{v_i} = n(V_i = v_i)$ be the frequency of $V_i = v_i$, and $n_{v_i,v_j} = n(V_i = v_i, V_j = v_j)$ be the frequency of $V_i = v_i$ and $V_j = v_j$.

The chi-squared statistic is widely used to test independence of variables. If the null hypothesis is that V_i and V_j are independent, then the chi-squared statistic is defined as

$$\chi^2_{v_i v_j} = \sum_{v_i, v_j} \frac{\left(n(v_i, v_j)n - n(v_i)n(v_j)\right)^2}{n(v_i)n(v_j)n}. \tag{3}$$

The degree of freedom of the χ^2 distribution is $(u_i - 1)(u_j - 1)$. If the null hypothesis is that V_i and V_j are independent conditionally on a variable (or a vector) S, then the chi-squared statistic is given by

$$\chi^2_{v_i v_j s} = \sum_{v_i, v_j, s} \frac{(n(v_i, v_j, s)n(s) - n(v_i, s)n(v_j, s))^2}{n(v_i, s)n(v_j, s)n(s)}. \tag{4}$$

The degree of freedom of the χ^2 distribution is $(u_i - 1)(u_j - 1)(u_s - 1)$, where u_s is the number of levels of the variable (or the vector) S. We can get the critical value δ of the chi-squared statistic at any significance level. When $\chi^2_{v_i v_j}$ is smaller than a certain δ, we say that V_i and V_j may be marginally independent, denoting as $ID(V_i, V_j)$, otherwise we say that V_i and V_j are dependent, denoting as $D(V_i, V_j)$. When $\chi^2_{v_i v_j s}$ is smaller than δ, we say that V_i,V_j are conditionally independent given S, denoting by $CI(V_i, V_j|S)$, otherwise we say that V_i,V_j are dependent conditionally on S, denoting by $CD(V_i, V_j|S)$.

The simple LCD causal discovery algorithm was introduced to data mining field by Cooper [4]. Cooper presented a simple algorithm to discover causal relationships over triplets of variables as shown in Table 1.

Table 1. The simple LCD algorithm proposed by Cooper [4]

```
Simple LCD algorithm
Input:
    A variable $W$, which is assumed not to be caused by any other variable in
    the set $V$ of discrete variables in a complete database.
Output:
    A list of possible causal relationships and the conditional probability.

    For each variable $X$ other than $Y$
      If $D(X, W)$
        For all variables $Y \notin \{X, W\}$
        If $D(X, Y)$ and $CI(Y, W|X)$
        Output "$X$ might cause $Y$" and the conditional probability $P(Y|X)$
        End{for};
    End{for};
End{LCD}.
```

If there are l variables in V and n cases in database, then the computational complexity of the simple LCD is $O(l^2n)$ and the space complexity is $O(ln)$. We also give the proof of correctness of this algorithm in the next section. So, if all conditions hold, we can conclude strictly that "X should cause Y" in theory.

The focus on searching causal relationships over only triplets of variables makes the simple LCD algorithm computational efficiency, but also loses the ability to identify many causal relationships that cannot be encoded in a single causal chain.

The variables W, H, X_1, X_2 and Y have underlying causal relationships represented by G_2 in Fig. 1. We have the database with measured variables W, X_1, X_2, Y and we also know that W is not caused by V, where $V = \{X_1, X_2, Y\}$. From the Markov property of the causal diagram G_2, we have $D(W, X_1)$,$D(Y, X_1)$, $CD(W, Y|X_1)$, $D(W, X_2)$, $D(Y, X_2)$ and $CD(W, Y|X_2)$. There is no one-order conditional independence, where the order denotes the

Table 2. The Enhanced LCD (ELCD) algorithm

```
ELCD algorithm
Input:
    A variable $W$, which is assumed not to be caused by any other variable
    in the set $V$ of discrete variables in the complete database. An integer $K$,
    which is the highest order for testing conditional independence.
Output:
    A list of possible causal relationships and the causal conditional probability.

    Step one: Let $V_1 = \Phi$. If $D(W,X)$ set $V_1 = V_1 \cup \{X\}$ for all $X \in V$.
    Step two:
      For any $Y \in V_1$
        Let $V_Y = \Phi$. If $D(Y,X)$, set $V_Y = V_Y \cup \{X\}$ for all $X \in V_1, X \neq Y$.
        Let $\chi_k\{V_Y\}$ be the set of all subsets of $V_Y$ with $k$ variables and let $H = \Phi$.
        For $k = 1$ to $K$
          For any subset $S \in \chi_k\{V_Y\}$
            If any subset of $S$ is not in $H$ and $CI(Y,W|S)$
            Do
            Output "all variables in $S$ are causes of $y$" and the conditional
            probability $P(Y|S)$;
            Let $H = H \cup \{S\}$;
            End{Do}
          End{for};
        End{for};
      End{For};
    End{Step two};
End{ELCD}.
```

number of conditional variables. So, from the simple LCD algorithm, we cannot discover any causal relationship among V.

In order to enhance the ability to mine causal relationships, we present the enhanced LCD algorithm as shown in Table 2.

The ELCD outputs that the data support S as causes of Y, and the algorithm also could output the causal distribution which equals the conditional probability $P(Y|S)$. Let y denote some value of Y and s denote a value of vector S. The causal probability which equals $P(Y = y|S = s)$ represents the probability that Y takes the value y given that we manipulate S to have the value s.

In the next section, we discuss the properties of the ELCD algorithm. Especially in the case of $K = 2$, if we get that X_1 and X_2 are causes of Y, we can further conclude that X_1 cannot intersect the causal influence of X_2 on Y, and X_2 cannot intersect the causal influence of X_1 on Y either. Thus, we can discover some causal relationships shown as G_2 in Fig. 1. There is no any marginal independence or conditional independence given one variable, and thus we can get that X_1 and X_2 are causes of Y from $CI(W,Y|X_1,X_2)$. Moreover, we can also obtain the local structure $X_1 \rightarrow Y \leftarrow X_2$.

The choice of K in this algorithm depends on the requirement of applications. In general, higher order tests of independence can be relatively less reliable. A small integer K, say two or three, is preferable. For the worst situation, the computational complexity of ELCD is $O(l^{K+1}n)$, where l^K is the number of testing conditional independencies and $O(n)$ is the time complexity of one test of conditional independence.

We consider the causal relationships shown as G_1 in Fig 1. If variables W, A, B, C and D are measured in database, we can get following causal relationships: $C \to D$, $A \to D \leftarrow B$ and $A \to C \leftarrow B$. Further research can reconstruct the complete causal structure of A, B, C and D, but they are beyond the scope of this paper.

Causal diagrams have been used to represent an interested system in many fields such as sociology, epidemiology and business [10, 9]. Especially, in bioinformatics, they are useful for extracting meaningful biological insights from the resulting data sets. Friedman [3] used them to provide a concise representation of complex cellular networks by composing simpler submodels. Jansen et al. [6] developed an approach using them to predict protein-protein interactions genome-wide in yeast. In some biological studies, we have the knowledge that some variables could not be influenced by the other variables, say gene promoter in gene regulation networks. Thus, we can use the ELCD algorithm to learn the regulation relationships and interactional regulation relationships of those variables. For example, suppose that the underlying gene regulation network shown in Figure 1 as G_1. We know that W is a gene promoter, that is, W could not be regulated by any other gene. So, using ELCD algorithm, we can get the regulation relationships, $C \to D$, $A \to D \leftarrow B$ and $A \to C \leftarrow B$. They imply that genes A and B regulate gene C, and then gene C regulates gene D.

4 The Properties of Algorithms

D-separation and faithfulness condition are two important conceptions in constrain-based learning. The faithfulness condition describes the relationships between the structure of causal diagram and the joint distribution. The d-separation describes the properties of paths in a causal diagram. The definitions of those two conceptions can be found in [9].

Let $G =< V, E >$ be a causal diagram and $P(V)$ be the joint distribution of V. Suppose that the diagram G and the distribution P satisfy the Markov condition and faithfulness condition. Theorem 1 shows the properties of the simple and enhanced LCD algorithms. For a special case of $K = 2$, theorem 2 gives a stronger property of outputs of the ELCD algorithm.

Theorem 1. $X_1, \cdots, X_k$ Y W W $X_1, \cdots, X_k$, $X_1, \cdots, X_k$ Y , $CI(W, Y|X_1, \cdots, X_k)$ $CD(W, Y|S)$ S $X_1, \cdots, X_k$ $X_1, \cdots, X_k$ Y

From $CI(W, Y|X_1, \cdots, X_k)$, we have that $X_1, \cdots, X_k$ d-separate W and Y in the causal diagram G. Lauritzen [7] showed that if $X_1, \cdots, X_k$ d-separate W and Y in the moral subgraph induced by ancestors of W and Y, then $CI(W, Y|X_1, \cdots, X_k)$. Moreover, $CI(W, Y|X_1, \cdots, X_k)$ also implies that $X_1, \cdots,$

X_k must d-separate W and Y in the moral subgraph induced by ancestors of W and Y under the faithfulness. So there is a subset S of ancestors of W and Y such that $CI(W, Y|S)$ and $S \subset X_1, \cdots, X_k$. From conditions of this theorem, we have $S = \{X_1, \cdots, X_k\}$. Thus $X_1, \cdots, X_k$ are the ancestors of W and Y. Furthermore, because W is not the descendant of $X_1, \cdots, X_k$, we have that $X_1, \cdots, X_k$ must be the ancestors of Y.

Theorem 2. X_1, X_2 Y W $D(W, X_1)$, $D(W, X_2)$, $D(X_2, Y)$, $D(X_1, Y)$, $CD(W, Y|X_1)$, $CD(W, Y|X_2)$ $CI(W, Y|X_1, X_2)$ X_1 X_2 Y X_1 (X_2) Y X_2 (X_1)

From $D(W, X_1)$, $D(W, X_2)$, $D(X_1, Y)$ and $D(X_2, Y)$, there are *d*-connective paths between W and X_1, X_2, and between X_1, X_2 and Y. Let $S^W_{\rightarrow X_1}$ denotes the set of paths between W and X_1 in which all edges adjacent to X_1 point to X_1 and $S^W_{\leftarrow X_1}$ denotes the paths out of X_1. In the same way, $S^Y_{\rightarrow X_1}$ and $S^Y_{\leftarrow X_1}$, $S^W_{\rightarrow X_2}$and $S^W_{\leftarrow X_2}$, $S^Y_{\rightarrow X_2}$ and $S^Y_{\leftarrow X_2}$, are used to denote the sets of paths between Y and X_1, W and X_2, Y and X_2 respectively. We have that X_1 and X_2 are the ancestors of Y from Theorem 1. It implies that $S^Y_{\leftarrow X_1} \neq \Phi$ and $S^Y_{\leftarrow X_2} \neq \Phi$.

Without loss of generality, we assume that there is at least one path from X_2 to Y such that $X_2 \rightarrow \cdots \rightarrow Y$ does not go through X_1. Now, we only need to show that there is at least one path from X_1 to Y such that $X_1 \rightarrow \cdots \rightarrow Y$ does not go through X_2. There are two relationships between X_1 and X_2, X_1 is the ancestor of X_2 or X_1 is not the ancestor of X_2. If X_1 is not the ancestor of X_2, we get that every path in $S^Y_{\leftarrow X_1}$ is a directed path from X_1 to Y that does not go through X_2.

When X_1 is the ancestor of X_2, $X_1 \rightarrow \cdots \rightarrow X_2 \rightarrow \cdots \rightarrow Y$ occurs in the causal diagram. Let us consider the paths in $S^W_{\rightarrow X_2}$. We partition it to two sets, one contains the paths that go through X_1 and the other contains the paths that do not go through X_1, denoted as S_1 and S_2 respectively. If S_2 is empty, we have that there is at least one direct path from X_1 to Y that does not go through X_2. Otherwise X_1 *d*-separates W and Y, it's contradictory to the condition $CD(W, Y|X_1)$. If S_2 is not empty, we also have that there is at least one direct path from X_1 to Y that does not go through X_2, otherwise $CI(W, Y|X_1, X_2)$ does not hold or $CI(W, Y|X_2)$ holds.

5 Conclusion

This paper addresses the problem of mining causal relationships from data. An approach that learns causal relationships by testing the independencies among variables has been developed. We discussed two algorithms, the simple LCD algorithm and the ELCD algorithm. The simple LCD algorithm introduced by Cooper [4] is an efficient method to learn causal relationships, but it can discover the causal relationships only in a single causal chain. The ELCD algorithm proposed in this paper improves the ability to learn causal relationships for a more general case with multiple causal chains. We have presented how to use the ELCD algorithm to mine causal relationships and also proposed a potential application of the ELCD algorithm in bioinformatics. Further, we discussed the properties of the ELCD algorithm.

Acknowledgements. The authors would like to thank the Editor and referees for their helpful comments. This research was supported by the 36th postdoctoral science fund 2004036180, NSFC and NBRP 2003CB715900.

References

1. Agrwal, R., Imielinski, T., and Swami, A.: Mining association rules between sets of items in large databases . In: Proceedings of the 1993 ACM SIGMOD Conference on Management of Data (1993) 207-216
2. Brin, S., Motwani, R., Silverstein C.: Beyond Market Baskets: Generalizing Association Rules to Correlations. In: Proceedings of the ACM SIGMOD Conference on Management of Data, Tucson, AZ (1997) 267-276
3. Friedman, N.: Inferring cellular networks using probabilistic graphical models. Science 303 (5659) (2004) 799-805
4. Cooper, G. F. : A Simple Constraint-based Algorithm for Efficiently Mining Observational Databases for Causal Relationships. Data Mining and Knowledge Discovery 1(1997) 203-224
5. Hecherman, D.: Bayesian Networks for Data Mining. Data Mining and Knowledge Discovery 1(1997) 79-119
6. Jansen R., Yu H. Y., Greenbaum, D.: A Bayesian networks approach for predicting protein-protein interactions from genomic data. Science 302(5644) (2003) 449-453
7. Lauritzen, S. L.: Graphical Models. Clarendon Press Oxford (1996)
8. Piatetsky, S. G., Frawley, W. J.: Knowledge Discovery in Databases. AAAI/MIT Press (1991)
9. Spirtes, P. , Glymour, C. , Scheines, R.: Causation, Prediction and Search. Springer-Verlag New York Inc (1993)
10. Verstein, C. S. , Brin, S. , Motwani, R., andUllman, J.: Scalable Techniques for Mining Causal Structures. Data Mining and Knowledge Discovery 4(2000) 163-192

Mining Least Relational Patterns from Multi Relational Tables

Siti Hairulnita Selamat[1], Mustafa Mat Deris[2],
Rabiei Mamat [1], and Zuriana Abu Bakar[1]

[1] Department of Computer Science,
University College of Science and Technology,
21030 Kuala Terengganu, Malaysia
{rab, zuriana}@kustem.edu.my
[2] Faculty of Information Technology and Multimedia,
College University Technology Tun Hussein Onn,
86400 Parit Raja, Batu Pahat, Johor, Malaysia
mmustafa@kuittho.edu.my

Abstract. Existing mining association rules in relational tables only focus on discovering the relationship among large data items in a database. However, association rule for significant rare items that appear infrequently in a database but are highly related with other items is yet to be discovered. In this paper, we propose an algorithm called Extraction Least Pattern (ELP) algorithm that using a couple of predefined minimum support thresholds. Results from the implementation reveal that the algorithm is capable of mining rare item in multi relational tables.

1 Introduction

Nowadays, the quantity of data is expanding rapidly. Most of the data are stored in a relational table in order to support a variety of administrative management where it provides valuable input for organizational decision-making [11]. It will be extracted from the tables through various techniques to obtain valuable information and knowledge from those vast amounts of data. Currently, mining association rules in relational tables only focus on discovering the relationship among large itemsets in the tables that satisfy the support and confidence set by the users [4]. Nevertheless, the existing association rule discovery techniques do not consider the occurrence frequency pattern of data, and discover the association rules using the same support on the whole data, so the discovered rules with regard to rare data may be redundant and as a result, unnecessary rules may be generated.

In this paper, a new algorithm called Extraction Least Pattern (ELP) algorithm to extract least relational patterns of data items from multi relational tables is proposed. Least data items are referred to the data items that its frequency in the relational tables does not satisfy the minimum support but are highly associated with the specific [16]. This enables us to identify significant rare data associated with specific data in a way that rare data occur simultaneously with specified data more frequently that the

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 59–66, 2005.

average co-occurrence frequency in the relational tables. A range of predefined minimum support thresholds are used to discover the least data. By using a couple of minimum support thresholds, it captured more meaningful data to discover interesting patterns. This paper is organized as follows. In section 2, the related wok will be discussed. In section 3, we discuss the background of this project. ELP (Extraction Least Pattern) algorithm and its detail experiments are present in section 4. Finally, we conclude this paper in section 5.

2 Related Work

The concept of relational patterns and the utilized elements of Apriori [10] algorithm to extract the relational patterns from multiple relational tables have been proposed in [1], which used bottom-up approach. Another previous works focus on advanced association rules problems that involved relational tables have been briefly discussed in [2], [3], [8], [20] and [21]. However, the model used in these studies only focus on providing an approach to generate the large datasets that satisfied predefined minimum support threshold. In other cases, there are also researches discovering on significant rare data in the table that have been studied extensively in [14], [15] and [16] but, unfortunately all these models only discovering the rare data in a single table instead of multi relational tables.

3 Background

Association rule mining is one of the processes of discovering hidden patterns in data. It also known as finding association, correlation or causal structures among sets of data items or objects in transaction tables, relational tables or other information repositories. Thereupon, one of its mining algorithms, Apriori algorithm is an influential algorithm used to mine all frequent data items in a table that satisfy the user predefined minimum support and minimum confidence constraints. A frequent data item is the data whose support is greater than user predefined minimum support threshold.

Relational data mining is one of data mining techniques for relational tables. Most existing traditional data mining approaches which are look for patterns in a single table are called propositional patterns. In contrast, relational data mining approaches that are seek for patterns among multiple tables are called relational patterns. That is, relational pattern involve multiple relations that represent the information as a set of relations. This is because, a relational table consists of a set of multiple tables and a set of associations (i.e. constraints) between pairs of tables describing how records in one table relate to records in another table. An association between multiple tables describes the relationships between records in these tables. In relational model, the association between these relational tables is defined through primary and foreign key attributes. If relation R_j includes, among its attributes, relation $\mathrm{R}_{\mathrm{j+1}}$'s primary key, then a tuple t_1 in R_j and a tuple t_2 in R_{j+1} refer to one another if t_1[Foreign_Key] = t_2[Primary_Key] [1].

4 Approach

The bottom-up approach used in this paper to extract the least patterns beginning from the leaf relation R_i up to relation R_{i-n}, where the leaf relation R_i is n levels downward in the path. In addition, based on hierarchical concept, the leaf relation R_i is a leaf tuple. Thus, the relations composing the path are considered as R_{i-n}, R_{j-n+1}, …, R_i. Our approach is only considered the least data items that occur infrequently but appear simultaneously with specific data items in high proportion. In brief, the least items are data items that rarely occur in a table. Hence, the least data items can only be found in the data if the predefined minimum support threshold has to be set very low. However, this situation would cause too many rules generated, which most of them are not important. If a higher minimum support threshold is used, we might miss out on generating important association rules. This problem is known as the rare item problem. Despite these drawbacks, our approach introduces usage of a range of two predefined minimum support thresholds that may overcome these problems. The extracted least data items must be satisfied the range of predefined minimum support thresholds, that is data items must be contained in between first and second user-predefined minimum support threshold. Four phases are involved in ELP algorithm in mining least data items on multiple relational tables. Those are *Extract least data items*, *Extract sibling patterns*, *Extract join patterns*, and *Extract least relational patterns*.

Phase 1: Extract Least Data Items

Basically, in normalized relational tables design, it would be to have three tables that is first table for contact, second table for groups, and the third table called 'joiner' table depicting what groups a contact belongs to. In this phase, we select the related attributes from the 'joiner' table, and construct a table called JoinTable as shown in Table 2. For example, we extract relational patterns from the sample hospital database as shown in Table 1, where the sample database has two main tables, i.e., Department Table, and Procedures Table. Assume that the first minimum support parameter, *fminsup* is 25%, and the second minimum support parameter, *sminsup* is 10%. Those least data items are only extracted if they are satisfying a range of two predefined minimum support thresholds. Using bottom-up approach starting from the leaf level tuple L_j^{i-n}, each extracted least data item is mapped to a unique key and stored in a set that is split up into a few tables according to the field name. Eventually, from this example, two tables as shown in Table 3 and Table 4 will be generated.

Algorithm 1. Algorithm applied to JoinTable in order to extract least data items matched

```
for each $f_j \in D$ do        // each field, f in table, D
    for each $i_n \in I$     // $I = i_1, i_2, \dots, i_n$ (A set of data items)
      if ($sminsup \le i_n$.support < $fminsup$) then
            $i_n \in I$
              $k=k+1$     // increase unique key, $k$
    end
  $L=L \cup L_j$          // L Least data items
end
```

Table 1. Two tables from Hospital Database

R_2 : Department Table

ID	Department	LOS
100	ER	1 day
100	internal	2-3 days
200	pediatric	2-3 days
300	ER	3-6 hour
400	ER	1 day
400	pediatric	1 day
400	surgery	7-10 hour
500	surgery	3-6 hour
600	ICU	1-2 hour

R_3 : Procedures Table

ID	Department	Procedures	Cost($)
100	ER	BC	2-5K
100	ER	ECG	1-2K
200	pediatric	BC	1-2K
200	pediatric	X-ray	5-7K
300	ER	ECG	1-2K
400	pediatric	BC	1-2K
400	pediatric	ECG	5-7K
400	surgery	operation	7-9K
500	surgery	operation	2-5K
600	ICU	fixation	1-2K

Table 2. JoinTable table

ID	Department	Procedures
100	ER	BC
100	ER	ECG
200	pediatric	BC
200	pediatric	X-ray
300	ER	ECG
400	pediatric	BC
400	pediatric	ECG
400	surgery	operation
500	surgery	operation
600	ICU	fixation

Table 3. Procedures table

ID	Procedures	Key
200	X-ray	1
400	operation	2
500	operation	2
600	fixation	3

Table 4. Department table

ID	Department	Key
400	surgery	4
500	surgery	4
600	ICU	5

Phase 2: Extract Sibling Patterns

At this phase, either Descendant or Transformed, both tables are constructed in order to extract any sibling pattern that exists in '*sibling_collection'* field. These tables transform relation R_j that each least data items of tuple in R_j is replaced with set of unique keys depending on the parent tuple (i.e., ID) that representing all least data items contained in that tuple's sub-tree. For instance, the Descendant table as shown in Table 5 consists of joining of the least data items in Table 3, and Table 4 but the least data items from Table 4 have been replaced with matched unique keys. As a consequence, in the Transformed table, contained all patterns in Least table, Join table and Sibling table, as these patterns constitute all possible least relational patterns with respect to relation R_{i-n}, contained in sub-trees of tuples in R_j. Each extracted least $(n \geq 2)$-Sibling pattern is mapped to a unique key and stored in set S_j^{i-n}. Algorithm 2 is the algorithm used for extracting any sibling pattern that exists in generated table. For example, based from result of our implementation, there is no sibling pattern extracted from Table 5 but there is one sibling pattern extracted from Table 7.

Table 5. Descendant table

ID	Data_Item	Sibling_Collection
200	pediatric	1
400	surgery	2
500	surgery	2
600	ICU	3

No Sibling Pattern

Algorithm 2. Algorithm that used to generate sibling pattern

```
for each $l_j \in L$      do
     for each $t_m \in T$ do    // $T$ data items in sibling collection
          if ($PID$= $t_m.ID$) and ($t_m$.duplicate = True)   then
                $t_m \in S_j$
                $k=k+1$     // increase unique key,$k$
     end
     $S=S \cup S_j$
end
```

Phase 3: Extract Join Patterns

If the tuple is a leaf relation, its tuples have no join pattern and thus, this phase is skipped. These join patterns generated by joining all descendant patterns in Descendant table with all the least data items in least table using Algorithm 3. Each generated least join pattern is then mapped to a unique key and stored in a set J_j^{i-n} as shows in Table 6. An extracted join pattern is represented with an ordered list of two members $< l, ds >$, where $l \in L_j^{i-n}$ and $ds \in DS_{j+1}^{i-n}$, which these data items are contained in tuple *t*'s sub tree and following criteria are satisfying:

1. *t* contains *l*
2. there exist $< PID, \{P_{PID}\} > \in DS_{j+1}^{i-n}$, such that PID = *t*.ID, and $ds \subseteq \{P_{PID}\}$, where *t*.ID is tuple *t*'s ID.

Table 6. JoinPattern table

Join_Pattern	Key
4 , 2	6
5 , 3	7

Algorithm 3. Algorithm that used to generate join pattern

```
for each $l_j \in L$ do
    for each $t_m \in T$ do
        if ($PID= t_m.ID$) then
            $<l_j, t_m> \in J_j$
            $k=k+1$    // increase unique key
    end
    $J=JUJ_j$
end
```

Table 7. Transformed table

ID	Sibling_Collection	
400	{4 , 2 , 6}	*Sibling Pattern Extracted*
500	{4 , 2 , 6}	
600	{5 , 3 , 7}	

Table 8. S2Pattern table

Sibling_Pattern	Key
{4 , 2 , 6}	8

Phase 4: Extract Least Relational Patterns

This final phase construct the sets that contained all least patterns in the Department table, Procedures table, SiblingPattern table and JoinPattern table. In other form of results generated from phase one to four are: $\hbar_j L_j^{i-n}, \hbar_j S_j^{i-n}, \hbar_j J_j^{i-n}$, where *j* starting from the leaf relation's index *i* to *i-n*, as shown in Table 9. These set contains unique keys representing all least relational patterns that have been removed all redundant patterns. In addition, these patterns are encapsulated in their respected parent tuples (i.e., ID tuple).

Table 9. LeastPattern table

ID	Set_Patterns
400	8 , 4 , 2 , 6
500	8 , 4 , 2 , 6
600	5 , 3 , 7

Based on the implementation results, the extracted least relational patterns contains unique keys that represented each of its nodes having a number pointer to its parent data items, which indicate that the data items are related in each other. Specifically, the least relational patterns captured relationships between the tuples across multi relational tables from which co-occurrence of attributes were extracted. Although these least relational patterns are rarely occur in a database, it is special interesting cases to be discovered. Therefore, the least data items should not totally ignore to avoid potentially valuable information loss. These extracted least relational patterns can be used to improve and support variety of organizational decision-making tasks such as hospitalization administrative databases. For instance, least relational patterns may be used to support hospitalization's decision making by identifying their patient behavior. More precisely, this implementation may used to discover unexpected data into an interesting pattern.

5 Conclusion

In this paper, we presented an ELP algorithm and discussed the approach used to discover the least relational patterns from multi relational tables. The ELP algorithm generated all least data items that satisfied a couple of predefined minimum support thresholds. Specifically, we used a couple of predefined minimum support threshold to extract least patterns be more meaningful and avoid valuable 'nuggets' of information from loss. The implementation results indicate that the introduced algorithm is capable of mining rare items for multi relational tables.

References

[1] M. Saar-Tsechansky, N. Pliskin, G. Rabinowitz, A. Porath, "Mining Relational Patterns from Multiple Relational Tables". Decision Support Systems, v.27, n.1-2, 177-195.

[2] R. Agrawal, R. Srikant, "Mining Qualitative Association Rules in Large Relational Tables". SIGMOD'96, Montreal, Canada, June 1996.

[3] X. Shang, K. Sattler, I. Geist, "Efficient Frequent Pattern Mining in Relational Databases". Workshop GI Working Group of the Knowledge Discovery (AKKD) in the context of the LWA 2004.

[4] R. Agrawal, R. Srikant, "Mining Association Rules Between Sets of Items in Large Databases". Proceedings of ACM SIGMOD, 207-216.

[5] R. Agrawal, R. Srikant, "Mining Sequential Patterns". In Proc. Of the 11th International Conference Data Engineering 1995.

[6] R. Agrawal, R. Srikant, "Fast Algorithms for Mining Association Rules". Proceedings of the VLDB Conference, 487-499.

[7] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, "From data mining to knowledge discovery: an overview", in: U.M. Fayyad, G. Piastetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996.

[8] A Swami, M. Houtsma, "Set-oriented Data Mining in Relational Databases". In International Conference Management of Data Engineering, Taipei, Taiwan, March 1995.

[9] R. Agrawal, R. Srikant, "Mining Generalized Association Rules". In Proc. Of the 21st International Conference on VLDB, Zurich, Switzerland, September 1995.

[10] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, A. I. Verkamo, “Fast Discovery of Association Rules”, in: U.M. Fayyad, G. Piastetsky-Shapiro, P. Smyth, R. Uthurusamy (Eds.), Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press.

[11] U.M. Fayyad, G. Piatetsky-Shapiro, P. Smyth: “From Data Mining to Knowledge Discovery in Databases”. In: Fayyad et al: Advances in Knowledge Discovery and Data Mining. AAAI Press/The MIT Press, Menlo Park 1996.

[12] M.H. Dunham, “Data Mining: Introductory and Advanced Topics”, New Jersey, Prentice Hall, 2003.

[13] B. Liu, W. Hsu, Y. Ma, “Mining Association Rules with Multiple Minimum Supports”, Proceedings of the 5th ACM SICKDD International Conference on Knowledge Discovery and Data Mining, San Deigo, California, United States, 337-341, Aug 1999.

[14] H. Yun, D. Ha, B. Hwang, K.H. Ryu, “Mining Association Rules On Significant Rare Data Using Relative Support”, The Journal of Systems and Software, Elsevier, v.67. n.3. 181-191, Sept 2003.

[15] N.F. Nabila, M.M. Deris, M. Y. Saman, A. Mamat, “Association Rules On Significant Rare Data Using Second Support”, (forthcoming).

[16] P.S.M. Tsai, C.-M. Chen, “Mining Interesting Association Rules From Customer Databases and Transaction Databases”, Information Systems, Elsevier, v.29. n.8. 685-696, Dec 2004.

[17] S. Sarawagi, S. Thomas, R. Agrawal, “Integrating Association Rule Mining with Relational Database Systems: Alternatives and Impilcations”, SIGMOD Record (ACM Special Interest Group on Management of Data), v.27. n.2. 343-355.

Finding All Frequent Patterns Starting from the Closure[*]

Mohammad El-Hajj and Osmar R. Zaïane

Department of Computing Science,
University of Alberta, Edmonton AB, Canada
{mohammad, zaiane}@cs.ualberta.ca

Abstract. Efficient discovery of frequent patterns from large databases is an active research area in data mining with broad applications in industry and deep implications in many areas of data mining. Although many efficient frequent-pattern mining techniques have been developed in the last decade, most of them assume relatively small databases, leaving extremely large but realistic datasets out of reach. A practical and appealing direction is to mine for closed itemsets. These are subsets of all frequent patterns but good representatives since they eliminate what is known as redundant patterns. In this paper we introduce an algorithm to discover closed frequent patterns efficiently in extremely large datasets. Our implementation shows that our approach outperforms similar state-of-the-art algorithms when mining extremely large datasets by at least one order of magnitude in terms of both execution time and memory usage.

1 Introduction

Discovering frequent patterns is a fundamental problem in data mining. Many efficient algorithms have been published on this problem in the last 10 years. Most of the existing methods operate on databases made of comparatively small database sizes. Given different small datasets with different characteristics, it is difficult to say which approach would be a winner. Moreover, on the same dataset with different support thresholds different winners could be proclaimed. Difference in performance becomes clear only when dealing with very large datasets. Novel algorithms, otherwise victorious with small and medium datasets, can perform poorly with extremely large datasets. The question that we ask in this work is whether it is possible to mine efficiently for frequent itemsets in extremely large transactional databases, databases in the order of millions of transactions and thousands of items such as those for big stores and companies similar to Wal-Mart, UPS, etc. With the billions of radio-frequency identification chips (RFID) expected to be used to track and access every single product sold in the market, the sizes of transactional databases will be overwhelming even to current

[*] This research is partially supported by a research grant from the National Sciences and Engineering Research Council of Canada.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 67–74, 2005.

state-of-the-art algorithms. There is obviously a chasm between what we can mine today and what needs to be mined. It is true that new attempts toward solving such problems are made by finding the set of frequent closed itemsets (FCI) [6, 7, 8]. A frequent itemset X is closed if and only if there is no X' such that $X \subseteq X'$ and the support of X equals to the support of X'.

Finding only the closed item patterns reduces dramatically the size of the results set without loosing relevant information. Closed itemsets reduce indeed the redundancy already in the set of frequent itemsets. From the closed itemsets one can derive all frequent itemsets and their counts. Directly discovering or enumerating closed itemsets can lead to huge time saving during the mining process.

While there are myriad algorithms to discover the closed patterns, their performances are indistinguishable for small and medium size databases. Experimental results are typically reported with few hundred thousand transactions. A recent study [9] showns that with real datasets, [illegible], the oldest algorithm for mining frequent itemsets, outperforms the newer approaches. Moreover, when results are discovered in few seconds, performance becomes almost irrelevant. The problem of performance becomes a real issue when the size of the database increases significantly (in the order of millions of transactions) or when the dimensionality of the problem increases (i.e. the number of distinct items in the database).

We present in this paper a new algorithm for discovering closed frequent itemsets, and report on a study illustrating the importance of such algorithm when mining very large databases.

The remainder of this paper is organized as follows: To put our algorithm in the context, we explain our new traversal approach in Section 2. Since we adopt some data-structures from the literature, FP-tree and COFI-trees, we briefly describe them in Section 3 where the new COFI-closed algorithm is also explained with illustrative examples. Section 4 depicts the performance evaluation of this new algorithm comparing it with existing state-of-the-art algorithms in particular for its speed, scalability, and memory usage on dense and sparse data. Section 5 concludes and highlights our observations.

2 Leap-Traversal Approach

In this paper we introduce a new leap-traversal approach that looks ahead at the nature of the transactional database, and suggests a set of patterns from different sizes to test where the frequent patterns (all, closed, or maximals) are subset of this suggested set. To illustrate the traversal, we take the case of closed itemsets. Step one of this approach is to look at the nature of the distribution of frequent items in the transactional database. Figure 1.A presents a transactional database where we can see that there are only 4 distributions of frequent items. Indeed, {A, B, C, D, E, F, G, H, I} occurs 3 times, we call this as branch-support of 3; {A, B, C, D, E, J, K, L} occurs also 3 times; {A, F, G, H, I} occurs twice; and {A, J, K, L} also occurs twice. We call each one of these

A	B	C	D	E	F	G	H	M
A	B	C	D	E	F	G	H	M
A	B	C	D	E	F	G	H	M
A	B	C	D	E	J	K	L	
A	B	C	D	E	J	K	L	
A	B	C	D	E	J	K	L	
A	J	K	L					
A	J	K	L					
A	F	G	H	I				
A	F	G	H	I				

(A)

number	frequent-path-bases	B. Support	intersect	Candidate patterns	Support
1	A B C D E F G H I	3	1∩2	A B C D E	6 √
2	A B C D E J K L	3	1∩3	A	10 √
3	A J K L	2	1∩4	A F G H I	5 √
4	A F G H I	2	2∩3	A J K L	5 √
			~~2∩4~~	~~A~~	~~10~~
			~~3∩4~~	~~A~~	~~10~~

(B)

Fig. 1. (A): transactional database. (B): Steps needed to generate closed patterns using the leap-traversal approach ($\surd$ indicates a discovered closed pattern. Barred entries are the eliminated candidates)

patterns a . Step 2 of this process intersects each one of these patterns with all other to get a set of potential candidates. Step 3 counts the support of each one of the generated patterns. The support of each one of them is the summation of supports of all its supersets of patterns. Step 4 scans these patterns to remove non-frequent ones or frequent ones that already have a frequent superset with the same support. The remaining patterns can be declared as closed patterns. Figure 1.B illustrates the steps needed to generate the closed patterns of our example from Figure 1.A. The major goals of this approach are the followings: 1. Avoid the redundancy of testing patterns either from size 1 until patterns of size k, where k is the size of the longest frequent pattern or from patterns of size n until patterns of size k, where n is the size of the longest candidate pattern. 2. We only check the longest potential patterns that already exist in the transactional database, even if they are of different lengths. From Figure 1.A we can find that there is no need to test patterns such as ABJ or AFC since they never occur in the transactional database. We also do not need to test patterns such as AB since they never occur alone without any other frequent items in the transactional databases.

The main question in this approach is whether we could efficiently find the . The answer is yes, by using the FP-tree [4] structure to compress the database and to avoid multiple scans of the databases and COFI-trees [2] to partition the sub-transactions as we wish to do, to generate the as illustrate in the next section.

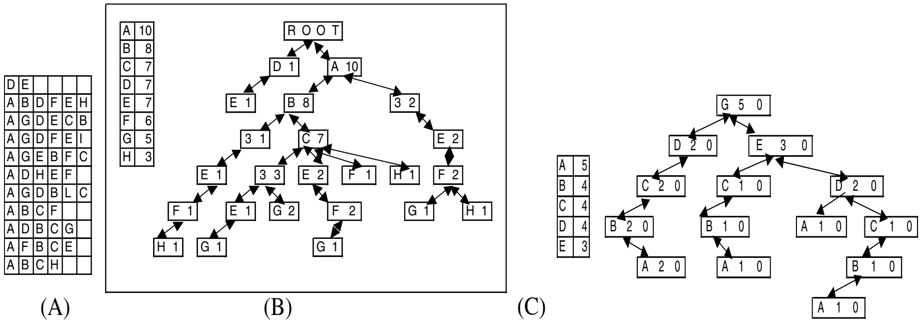
3 FP-Tree and COFI-Trees

The well-known FP-tree [4] data-structure is a prefix tree. The data structure presents the complete set of frequent itemsets in a compressed fashion. The construction of FP-Tree requires two full I/O scans. The first scan generates the frequent 1-itemsets. In the second scan, non-frequent items are stripped off the transactions and the sub-transactions with frequent ones are ordered based on their support, forming the paths of the tree. Sub-transactions that share the same prefix share the same portion of the path starting from the root. The FP-tree has also a header table containing frequent items and holds the head link for each item in the FP-tree, connecting nodes of the same item to facilitate the item traversal during the mining process [4].

A COFI-tree [2] is a projection of each frequent item in the FP-tree. Each COFI-tree, for a given frequent item, presents the co-occurrence of this item with other frequent items that have more support than it. In other words, if we have 4 frequent items A, B, C, D where A has the smallest support, and D has the highest, then the COFI-tree for A presents co-occurrence of item A with respect to B, C and D, the COFI-tree for B presents item B with C and D. COFI-tree for C presents item C with D. Finally, the COFI-tree for D is a root node tree. Each node in the COFI-tree has two main variables, [illegible] and [illegible]. [illegible] indicates the number of patterns the node has participated in at a given time during the mining step. Based on the difference between these two variables, [illegible] and [illegible] are generated. The COFI-tree has also a header table that contains all locally frequent items with respect to the root item of the COFI-tree. Each entry in this table holds the local support, and a link to connect its item with its first occurrences in the COFI-tree. A link list is also maintained between nodes that hold the same item to facilitate the mining procedure.

3.1 COFI-Closed Algorithm

The COFI-Closed algorithm is explained by a running example. The transactional database in Figure 2. A needs to be mined using a support greater or equal to 3. The first step is to build the FP-tree data-structure in Figure 2.B. This FP-tree data structure reveals that we have 8 frequent 1-itemsets. These are (A:10, B:8, C:7, D:7, E:7, F:6, G:5, H:3). COFI-trees are built after that, one at a time starting from the COFI-tree of the frequent item with lowest support, which is H. Since, in the order imposed, no other COFI-tree has item H then any closed pattern generated from this tree is considered globally closed. This COFI-tree generates the first closed pattern HA: 3. After that, H-COFI-tree is


D	E				
A	B	D	F	E	H
A	G	D	E	C	B
A	G	D	F	E	I
A	G	E	B	F	C
A	D	H	E	F	
A	G	D	B	L	C
A	B	C	F		
A	D	B	C	G	
A	F	B	C	E	
A	B	C	H		

A	10
B	8
C	7
D	7
E	7
F	6
G	5
H	3

A	5
B	4
C	4
D	4
E	3

Fig. 2. (A) A Transactional database. (B) FP-Tree built from (A). (C) G-COFI-tree pointers from header tables are not presented

discarded and G-COFI-tree, in Figure 2.C, is built and it generates (GABCD:3, GABC:4, GAE:3, GAD:4, and GA:5), a detailed explanation of the steps in generating these frequent patterns are described later in this section. F-COFI-tree is created next and it generates all its closed patterns using the same method explained later.

Mining a COFI tree starts by finding the [illegible]. As an example, we will mine the G-COFI-tree in Figure 2.C for closed patterns. We start from the most globally frequent item, which is A, and then traverse all the A nodes. If the [illegible] is greater than [illegible], the third counter on the node, then the complete path from this node to the COFI-root is built with [illegible] equals to the difference between the [illegible] and [illegible] of that node. All values of [illegible] for all nodes in these paths are updated with the [illegible] of the original node A. [illegible] (A, B, C, D: 2), (A, B, C, E: 1), (A, D, E: 1), and (A, B, C, D, E: 1) are generated from this tree. From these bases we create a special data structure called Ordered-Partitioning-Bases (OPB). The goal of this data structure is to partition the patterns by their length. Patterns with the same length are grouped together. This, on one hand allows dealing with patterns of arbitrary length, and on the other hand allows traversing the pattern space from the longest ones to the shortest ones and directly prunes the short ones if a frequent superset with same support is discovered as a candidate closed pattern.

This OPB structure is an array of pointers that has a size equal to the length of the largest [illegible]. Each entry in this array connects all [illegible] of the same size. The first entry links all [illegible] of size 1, the second one refers to all [illegible] of size 2, the n^{th} one points to all [illegible] of size n. Each node of the connected link list is made of 4 variables which are: the pattern, a pointer to the next node, and two number variables that represent the [illegible] and [illegible] of this pattern. The [illegible] reports the number of times this pattern occurs in the database. The [illegible] records the number of times this pattern occurs alone without other frequent items, i.e. not part of any other superset of frequent patterns. This [illegible] is used to identify the [illegible] from [illegible] as [illegible] have [illegible] equal to 0, while a [illegible] has [illegible] equal to the number of times this pattern occurs independently. The [illegible] is also used to count the support of any pattern in the OPB. The support of any pattern is the summation of the [illegible] of all its supersets of [illegible]. For example, to find the support for pattern X that has a length of k, all what we need to do is to scan the OPB from $k+1$ to n where n is the size of OPB, and sum the [illegible] of all supersets of X that do not have a [illegible] equal to 0, i.e. the [illegible]. The superset of X, as explained before are easily found using the prime codes.

In our example above, the first step is to build the OPB structure. The first pointer of this OPB structure points to 5 nodes which are (A, 5, 0), (B, 4, 0), (C, 4, 0), (D, 4, 0), and (E, 3, 0) which can be taken from the local frequent array of the G-COFI-tree (Figure 2.C). The first number after the pattern presents the [illegible] while the second number presents the [illegible]. The Second entry in this array points to all [illegible] of size two. A null pointer is being linked to this node since no [illegible] of size two are created. The third pointer points to one node which is (ADE, 1,1), the fourth points to (ABCD: 2: 2) and (ABCE: 1, 1), the fifth and last points to (ABCDE: 1:1). The leap-traversal approach is applied in the second step on the 4 [illegible], which are (ABCDE: 1: 1, ABCD:2 :2, ABCE: 2: 1, and ADE: 2: 1). Intersecting ABCDE with ABCD gives ABCD, which already exists, so nothing needs to be done. Same occurs when interesting ABCDE with ABCE. Intersecting ABCDE with ADE gives back ADE, which also already exists. Intersecting ABCD with ABCE gives ABC. ABC is a new node of size three. It is added to the OPB data structure and linked from the third pointer as it has a pattern size of 3. The [illegible] and the [illegible] of this node equal 0. [illegible] equals 0 indicates that this pattern is a result of intersecting between [illegible] and a [illegible]. Intersecting ABCD with ADE gives AD. AD is a new node of size two. It is added to the OPB data structure and linked from the second pointer. The [illegible] and the [illegible] of this node equal 0. Intersecting, ABCE with ADE gives AE. AE is also a new node of size two and is also added to the OPB structure, at this stage we can detect that there is no need to do further intersections. The third step in the mining process is to find the global support for all patterns. Applying a top-down traversal between these nodes does this. If node X is a subset of a [illegible] Y then its support is incremented by the [illegible] of node Y. By doing this we can find that ABCD is a subset of ABCDE, which has a [illegible] equals to 1. The ABCD support becomes 3 (2+1). ABCE support becomes 2, as it is a subset of ABCDE. At level 3 we find that ADE is a subset of only ABCDE so its support becomes 2. ABC support equals to 4. AD support equals to 4, and AE support equals to 3. At this stage all non-frequent patterns and frequent patterns that have a local frequent superset with same support are removed from OPB. The remaining nodes (ABCD:3, ABC:4, AE:3, AD:4, and A:5) are potentially global closed. We test to see if they are a subset of already discovered closed patterns with the same support from previous COFI-trees. If not then we declare them as closed patterns and add them to the pool of closed patterns. The G-COFI-tree and its OBP data structure are cleared from memory as there is no need for them any more. The same process repeats with the remaining COFI-trees for F and E, where any newly discovered closed pattern is added to the global pool of closed patterns.

4 Performance Evaluations

We present here a performance study to evaluate our new approach COFI-Closed against most of the state-of-art algorithms that mine closed patterns which are FP-Closed [3] and MAFIA-closed [1], CHARM [8]. Their respective authors provided us with the source code for these programs. All our experiments were conducted on an IBM P4 2.6GHz with 1GB memory running Linux

2.4.20-20.9 Red Hat Linux release 9. Timing for all algorithms includes the preprocessing cost such as horizontal to vertical conversions. The time reported also includes the program output time. We have tested these algorithms using synthetic datasets [5] on very large datasets. All experiments were forced to stop if their execution time reached our wall time of 5000 seconds. We made sure that all algorithms reported the same exact set of frequent itemsets on each dataset.

4.1 Experiments on Large Datasets

Mining extremely large databases is the main objective of this research work. We used five synthetic datasets made of 5M, 25M, 50M, 75M, 100M transactions, with a dimension of 100K items, and an average transaction length of 24 items. To the best our knowledge, these data sizes have never been reported in the literature before. CHARM could not mine these datasets. MAFIA could not mine the smallest dataset 5M in the allowable time frame. Only FP-Closed and COFI-Closed algorithms participated in this set of experiments. All results are depicted in Figure 3.A. From these experiments we can see that the difference between FP-Closed implementations and the COFI-Closed algorithm become clearer once we mine extremely large datasets. COFI-Closed saves at least one third of the execution time and in some cases goes up to half of the execution time compared to FP-Growth approach. The last recorded time for FP-Closed was mining 50 millions transactions while COFI-Closed was able to mine up to 100 millions transactions in less than 3500 seconds.

4.2 Memory Usage

We also tested the memory usage by FP-Closed, MAFIA and our approach. In many cases we noticed that our approach consumes one order of magnitude less memory than FP-Closed and two orders of magnitude less memory than MAFIA. Figure 3.B illustrates these results. We conducted experiments with the database size, the dimension and the average transaction length 1 million transactions, 100K items and 12 items respectively. The support was varied from 0.1% to 0.01%.

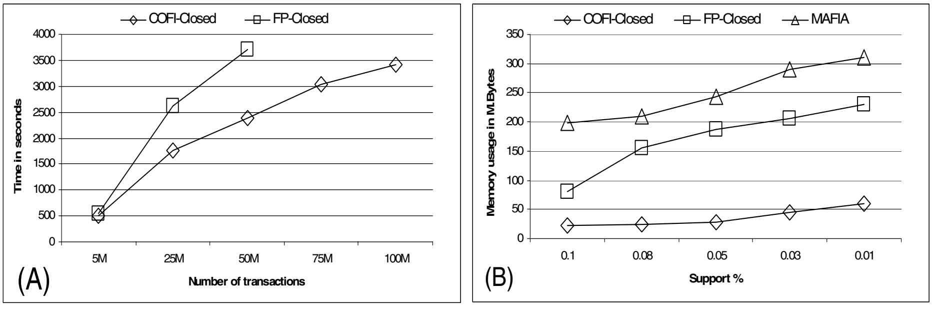


Fig. 3. (A) Scalability testing, (B) Disparity in memory usage T = 1000K, D = 100K

5 Conclusion

Mining for frequent itemsets is a canonical task, fundamental for many data mining applications. Many frequent itemset mining algorithms have been reported in the literature, some original and others extensions of existing techniques. They either model transactions horizontally or vertically and traverse the pattern space either bottom-up or top-down. However, most of the solutions assume to work on relatively small datasets in comparison to what we are expected to deal with in real applications such as affinity analysis in very large web sites, basket analysis for large department stores, or analysis of tracking data from radio-frequency identification chips on merchandize.

In this work we presented COFI-Closed, an algorithm for mining frequent closed patterns. This novel algorithm is based on existing data structures FP-tree and COFI-tree. Our contribution is a new way to mine those existing structures using a novel traversal approach. Using this algorithm, we mine extremely large datasets, our performance studies showed that the COFI-Closed was able to mine efficiently 100 million transactions in less than 3500 seconds on a small desktop while other known approaches failed.

References

1. D. Burdick, M. Calimlim, and J. Gehrke. Mafia: A maximal frequent itemset algorithm for transactional databases. In *ICDE*, pages 443–452, 2001.
2. M. El-Hajj and O. R. Zaïane. Non recursive generation of frequent k-itemsets from frequent pattern tree representations. In *In Proc. of 5th International Conference on Data Warehousing and Knowledge Discovery (DaWak'2003)*, September 2003.
3. G. Grahne and J. Zhu. Efficiently using prefix-trees in mining frequent itemsets. In *FIMI'03, Workshop on Frequent Itemset Mining Implementations*, November 2003.
4. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *2000 ACM SIGMOD Intl. Conference on Management of Data*, pages 1–12, 2000.
5. IBM_Almaden. Quest synthetic data generation code. http://www.almaden.ibm.com/cs/quest/syndata.html.
6. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In *International Conference on Database Theory (ICDT)*, pages pp 398–416, January 1999.
7. J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'03)*, Washington, DC, USA, 2003.
8. M. Zaki and C.-J. Hsiao. ChARM: An efficient algorithm for closed itemset mining. In *2nd SIAM International Conference on Data Mining*, April 2002.
9. Z. Zheng, R. Kohavi, and L. Mason. Real world performance of association rule algorithms. In *7th ACM SIGKDD International Conf. on Knowledge Discovery and Data Mining*, pages 401–406, 2001.

Multiagent Association Rules Mining in Cooperative Learning Systems

Reda Alhajj[1,2] and Mehmet Kaya[3]

[1] Department of Computer Science, University of Calgary, Calgary, Alberta Canada
[2] Department of Computer Science, Global University, Beirut, Lebanon
[3] Department of Computer Engineering, Firat University, 23119 Elazig, Turkey
alhajj@cpsc.ucalgary.ca, kaya@firat.edu.tr

Abstract. Recently, multiagent systems and data mining have attracted considerable attention in the computer science community. This paper combines these two hot research areas to introduce the term *multiagent association rule mining* on a cooperative learning system, which investigates employing data mining on a cooperative multiagent system. Learning in a partially observable and dynamic multiagent systems environment still constitutes a difficult and major research problem that is worth further investigation. Reinforcement learning has been proposed as a strong method for learning in multi-agent systems. So far, many researchers have proposed various methods to improve the learning ability in multiagent systems. However, reinforcement learning still has some drawbacks. One drawback is not modeling other learning agents present in the domain as part of the state of the environment. Another drawback is that even in learning case, some state-action pairs are experienced much less than others. In order to handle these problems, we describe a new action selection model based on association rules mining. Experimental results obtained on a well-known pursuit domain show the applicability, robustness and effectiveness of the proposed learning approach.

Keywords: multiagent systems, association rules, reinforcement learning, pursuit domain, data mining.

1 Introduction

Multiagent systems related research is an emerging subfield of distributed artificial intelligence, which aims at providing both: principles for the construction of complex systems involving multiple agents and mechanisms for the coordination of independent agents' behavior. The most important reason to use multiagent systems is to have a more natural modeling for real-life domains that require the cooperation of different parties. In particular, if there are different people with different perspectives or organizations with different goals and proprietary information, then a multiagent system is needed to handle their interaction [1]. Multiple agents are recognized as crucial for many real-world problems, such as engineering design, intelligent search, medical diagnosis, robotics, etc [2].

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 75–87, 2005.

Multiagent systems are different from single agent systems in the sense that there is no global control and globally consistent knowledge. So, limitations on the processing power of a single agent are eliminated in a multiagent environment. In other words, since data and control are distributed, multiagent systems include the inherent advantages of distributed systems, such as scalability, fault-tolerance and parallelism, among others.

Recently, there has been a considerable amount of interest in multi-agent systems. As a result, multiagent systems have been successfully utilized in many disciplines, including reinforcement learning, which is a learning technique that requires knowing almost nothing about the dynamics of the environment under consideration. An agent with its goal embedded in an environment learns how to transform one environmental state into another that contains its goal. An agent that has the ability of doing this task with minimal human supervision is called *autonomous* [3]. Autonomous agents learn from their environment by receiving reinforcement signals after interacting with the environment. Learning from an environment is robust because agents are directly affected by the dynamics of the environment. So far, many researchers have proposed various methods of reinforcement learning to improve the learning ability in multiagent systems [4]-[7]. The behavior of each agent changes as it learns, especially when considering a multiagent environment in which the agents autonomously learn. The state-transition function changes with time, after the behavior of each other agent is included in the environment. Mainly, the environment cannot be modeled as a Markov Decision Process (MDP). However, many multiagent reinforcement learning studies, like those mentioned above, apply reinforcement learning methods based on MDP without much modification.

In the literature, there have been several studies in which the internal model of each other learning agent is explicitly considered. Littmann [4] introduced 2-player zero-sum stochastic games for multiagent reinforcement learning. In that method, called mini-max Q-learning, one agent estimates the action of the other agent and learns based on the estimation, which is done according to the agent's own internal model. In zero-sum games, one agent's gain is always the other agent's loss, thus agents have strict opposite interests. Hu and Wellman [8] introduced a different multiagent reinforcement learning method for 2-player general-sum games. Since, one agent's gain is not necessarily the other agent's loss in a general-sum game, the mini-max Q-learning method is not always appropriate. However, according to both methods, while estimating the other agent's Q-function, the agent under consideration should observe the other agent's actions and the actual rewards received from the environment. Also, the former agent must know the parameters used in Q-learning of the latter agent. Finally, Nagayuki et al. [9] proposed another approach to handle this problem. In their learning method, which is also based on Q-learning, one agent estimates the other agent's policy instead of Q-function.

In this paper, we introduce the term *multiagent association rule mining*, which covers employing data mining techniques for a cooperative multiagent system. In other words, the novelty of this paper is in using association rules mining technique in order to estimate the action of the other agent in a multiagent system and describe a new action selection method.

Data mining is the discovery of previously unknown, potentially useful and hidden knowledge in databases. Inducing association rules is one of the important research issues in data mining. Association rules mining is an exploratory learning task to discover some hidden, dependency relationships among items, such as state and action, in a database.

In this study, we create an internal model database holding the past actions of the other agent. Then, by mining association rules, we estimate the next action of the other agent. As a result, there is no need to observe the other agent's actual rewards received from the environment, and to know the parameters that the other agent uses for Q-learning. One of the important drawbacks of Q-learning is that even in learning phase, some state-action pairs are experienced much less than others. In other words, although some states are not experienced sufficiently, it is expected that an agent should select an appropriate action in the corresponding state. In order to handle this problem, we also describe a new action selection model based on association rules that give more flexibility to action selection.

The main contributions of our work described in this paper can be stated as follows. Multiagent association rules mining in real time from: 1) internal model database to estimate the action of the other agent; and 2) lookup table used for Q-learning to select an appropriate action.

The rest of the paper is organized as follows. Section 2 provides the necessary background on association rules. Q-learning algorithm is explained in Section 3, after a review of the reinforcement learning theory. Section 4 describes a variant of the pursuit problem, to be used as a platform for experiments throughout this study. Section 5 presents how to mine association rules in multiagent learning systems. The results of the conducted experiments are reported in Section 6. Section 7 is summary and conclusions.

2 Association Rules

Association rules form an important class of regularities that exist in databases. Since it was first introduced in [10], the problem of association rules mining has received a great attention. The classic application is market basket analysis, which analyzes how the items purchased by customers are associated. An example of an association rule is: *bread* $\rightarrow$ *butter [sup=20%, conf=85%]*

This rule says that 20% of customers buy *bread* and *butter* together, and those who buy *bread* also buy *butter* 85% of the time.

The basic model of association rules is as follows: $A \rightarrow B$, where A and B denote subsets of a set I of all available items. As already mentioned above, the intended meaning of $A \rightarrow B$ is that a transaction $T \subset I$ which contains the items in A is likely to contain the items in B as well. Finally, A and B are called antecedent and consequent of the rule, respectively.

Of course, the development of algorithms for finding "interesting" association rules in a database D pre-assumes a formal definition of this qualification. A rule $A \rightarrow B$ is generally rated according to several criteria, none of which should fall below a certain (user-defined) threshold. In common use are the following measures:

($D_X=\{T \in D \mid X \subset T\}$ denotes the transactions in the database D which contain the items $X \subset I$, and $|D_X|$ is its cardinality):

- A measure of support defines the absolute number or the proportion of transactions in D containing $A \cup B$:

$$\sup(A \to B) = |D_{A \cup B}| \text{ or } \sup(A \to B) = \frac{|D_{A \cup B}|}{|D|}$$

- The confidence is the proportion of correct applications of the rule:

$$conf(A \to B) = \frac{|D_{A \cup B}|}{|D_A|}$$

3 Reinforcement Learning

As a machine-learning paradigm, reinforcement learning dates back to early days of cybernetics and work in statistics, psychology, neuroscience, and computer science. It addresses the question of how an autonomous agent that senses and acts in a given environment can learn to choose optimal actions to achieve its goal(s) [11]. It is the problem faced by agents that have no prior knowledge about the environment –tabula-rasa agents- and must learn behavior through trial-and-error interaction with the environment.

It is reported in [12] that there are two main strategies to solve reinforcement learning problems. The first strategy is to search in the space of behaviors in order to find one that performs well in the environment. The second strategy is to use statistical and dynamic programming methodologies to estimate the utility of taking actions in the state space of the problem. Other important approaches to reinforcement learning are game theory and functional approximation. In the former approach, the learning problem is modeled as a stochastic game and the player or player teams try to converge to Nash Equilibrium in terms of the learned information. On the other hand, the latter approach employs neural network to solve the problem in polynomial space in terms of the employed neurons, but lacks exhibiting convergence to optimal behavior.

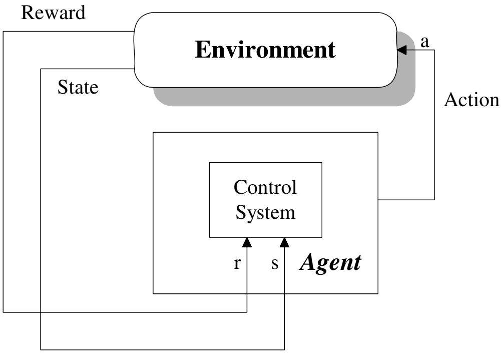


Fig. 1. Reinforcement learning model

In the standard reinforcement learning model, an agent interacts with its environment via perception and action links as shown in Figure 1. During each interaction step, the agent receives as input some indication of the current state s of the environment, and the agent chooses an action a, to generate as output. The action may or may not change the state of the environment. Finally, the agent receives back the value of its action as reinforcement r. Reinforcement can be either positive (called reward) or negative (called punishment). Further, the agent's action control system should choose actions that tend to increase the long-sum values of the reinforcement signal. An agent can learn to perform this process with time by systematic trial-and-error interaction with the environment.

3.1 Q-Learning Algorithm

Q-Learning is one of the most commonly used reinforcement learning methods, which does not need a model for its application and can be used online [13]. It is an incremental reinforcement learning method. Q-learning algorithms store the expected reinforcement value associated with each situation-action pair, usually in a look-up table. According to Q-learning, the agent selects an action based on an action-value function, called the Q-function. The Q-function is updated using the agent's experience. It is formalized next in Definition 1. Finally, the Q-learning process is described next in Algorithm 1.

Definition 1 (Q-function)**.** Given action a in state s, the Q-function, denoted $Q(s,a)$ is formally defined as:

$$Q(s,a) = (1-\alpha)Q(s,a) + \alpha(r + \gamma \max_{a' \in A} Q(s',a'))$$

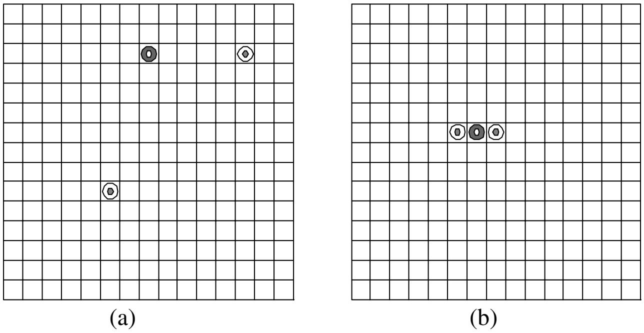
where α ($0 \leq \alpha < 1$) is the learning rate, γ ($0 \leq \gamma \leq 1$) is the discount parameter, $Q(s^l, a^l)$ is the value of action a^l in state s^l. Simply, the Q-function:

- defines the expected sum of the discounted reward attained by executing $a(\in A)$ in $s(\in S)$, where A is a finite set of actions and S is a finite set of states.
- determines the subsequent actions by the current policy π.

Algorithm 1: (Q-Learning Process)
The learning process proceeds according to the following steps:

1. Observe the current state s
2. Select an action a_i with respect to selection policy
3. Observe the new state s^l
4. Receive a reward r from the environment
5. Update the corresponding Q value for action a and state s according to Definition 1.
6. If the new state s^l satisfies a terminal condition, then terminate the current trial.

Otherwise let $s^l \rightarrow s$ and go back to step 1.



(a) (b)

Fig. 2. a) An initial position in 15x15 pursuit domain; b) A goal state

4 Pursuit Domain

In this paper, we consider a variant of the well-known pursuit problem. The characteristics of the environment are as follows:

- The environment is fully dynamic, partially observable, non-deterministic and has a homogeneous structure.
- Two hunter agents and a prey agent exist in a 15×15 grid world as shown in Figure 2. The initial position of each agent is determined randomly.
- At each time step, agents synchronously execute one out of five actions: staying at the current position or moving from the current position north, south, west, or east. More than one hunter agent can share the same cell. However, a hunter agent cannot share a cell with the prey. Also, an agent is not allowed to move off the environment. The latter two moves are considered illegal and any agent that tries an illegal move is not allowed to make the move and must stay in its current position. Finally, hunters are learning agents and the prey agent selects its own action with respect to a particular strategy, such as random or Manhattan-distance measure.
- The prey is captured when the two hunter agents are positioned at two of its sides. Then, the prey and the two hunter agents are relocated at new random positions in the grid world and the next trial starts.

5 Association Rules Mining in Multiagent Learning Systems

5.1 Mining Association Rules from Internal Model Database

So far, most of the work already done on multiagent learning assumes a stationary environment, i.e., the behavior of the other agent is not considered in the environment. Whereas it is more natural to consider a dynamic environment in the sense that an agent always learns and each other agent may change its behavior with time too. In such a case, the standard Q-learning approach is not appropriate.

In the work described in this paper, as an agent executes an action, the other agent's action is also considered. For this purpose, we must have an internal model database that holds the actions of the other hunter agent.

In other to explicitly express the dependency of the other agent's action, the hunter's Q-function is adjusted according to the formalism given next in Definition 2.

Definition 2 (Hunter's Q-Function)**.** Given a hunter h_1, which tries to estimate the action of an agent h_2. The corresponding Q-function is $Q(s, a_{self}, a_{other})$, where:

- s is an environment that contains the states of both the prey and h_2.
- $a_{self} (\in A_{self})$ and $a_{other} (\in A_{other})$ are actions of h_1 and h_2, respectively.
- A_{self} and A_{other} are the possible sets of actions for h_1 and h_2, respectively.

Table 1. The number of occurrences of state-action pairs

state/action	$a^{1}_{other}(\rightarrow)$	$a^{2}_{other}(\leftarrow)$	$a^{3}_{other}(\uparrow)$	$a^{4}_{other}(\downarrow)$	$a^{5}_{other}(stay)$	*count*
S_0	322	154	48	415	128	**1067**
S_1	211	64	103	21	82	**481**
S_2	78	145	294	124	462	**1103**
			• • •			
S_n	56	317	241	18	154	**786**

Table 2. The lookup table at any moment of learning process

		$a^{1}_{self}(\rightarrow)$	$a^{2}_{self}(\leftarrow)$	$a^{3}_{self}(\downarrow)$	$a^{4}_{self}(\uparrow)$	$a^{5}_{self}(stay)$	*count*
S_0	$a^{1}_{other}(\rightarrow)$	52.125	94.657	24.696	83.232	34.763	**1236**
	$a^{2}_{other}(\leftarrow)$	74.632	19.632	13.698	92.820	69.834	**3234**
	$a^{3}_{other}(\downarrow)$	91.852	64.633	25.367	41.140	80.978	**2487**
	$a^{4}_{other}(\uparrow)$	69.854	78.012	96.325	53.901	12.658	**2274**
	$a^{5}_{other}(stay$	58.632	23.736	84.132	47.263	97.689	**4256**
				• • •			
S_n	$a^{1}_{other}(\rightarrow)$	23.478	87.412	74.987	54.410	92.103	**5140**
	$a^{2}_{other}(\leftarrow)$	8.954	91.789	73.587	45.031	76.657	**978**
	$a^{3}_{other}(\downarrow)$	72.683	54.312	37.189	87.205	10.002	**2140**
	$a^{4}_{other}(\uparrow)$	94.127	51.978	7.014	46.879	34.478	**3106**
	$a^{5}_{other}(stay)$	66.798	30.412	87.631	12.163	86.156	**1358**

According to the above definition of Q-function, deciding on whether the action of a given agent is good or not depends on the action of the other agent. In other words, a_{other} is a hidden and major variable in selecting the action a_{self}. In this study, a_{other} is estimated based on the association rules extracted from the internal model database. If one hunter observes the other hunter in its visual environment, then the association rule $s \rightarrow a_{other}$ is mined from observations of the other hunter's past actions. On the other hand, if one hunter could not perceive the other hunter, for the unseen hunter in state *s*, a random association rule is assigned, i.e., a random action.

Table 1 shows the interval model database at one moment of the learning process. While each cell shows the number of occurrences of the corresponding state-action pair, the column *count* in Table 1 gives the number of occurrences of each state.

At the beginning of the learning process, the user gives a minimum support value for the variable *count*. If the count value of a state reaches the specified minimum support value, it is assumed that the state was experienced sufficiently. In such a case, the hunter agent under consideration estimates the action of the other agent with respect to the highest confidence value. If a state is not experienced sufficiently, the agent estimates the action of the other agent with respect to the confidence value given by the user. If the number of occurrences of a state-action pair is less than the user specified minimum confidence value, this action is not selected in the corresponding state. If there is action more than once exceeding the minimum confidence value in a state, then the possibility of selecting an action a_i is found by the following formula:

$$p(a_i \mid s) = \frac{conf\,(s \rightarrow a_i)}{\sum_{a_j \in A(MinConf\,)} conf\,(s \rightarrow a_j)}$$

where $conf(s \rightarrow a_i)$ is the confidence value of the rule $s \rightarrow a_i$, $A(MinConf)$ is the possible set of actions that exceed minimum confidence value for the corresponding agent.

For example, if we assume the minimum support value as 1000, since the count value of the state S_0 exceeds this threshold, the hunter agent under consideration estimates that the other agent will select the action a_4. However, if the state S_1 is observed and minimum confidence value is determined as 15%, then the action of the other agent is estimated as one of the three actions: a^1_{other}, a^3_{other} and a^5_{other}. However, as it can be seen easily from the table that the chance of selecting a^1_{other} is greater.

5.2 Mining Association Rules from Lookup Table

In this section, we introduce how to mine from the lookup table association rules that show the relationship between state and action. The multiagent learning method developed for this purpose is described in Algorithm 2.

Algorithm 2: (Learning Process) The proposed learning process is based on association rules mining and involves the following steps:

1. The hunter agent under consideration observes the current state *s* and estimates the other agent's action a_{other} based on association rules.
2. According to the estimated a_{other}, the action a_{self} is selected. This process is similar to mining the association rules from internal model database. Table 2 shows the lookup table at one moment of learning process. While each cell gives the Q-value of corresponding state-action pair, the count variable indicates the number of occurrences of corresponding state in case of a_{other}. In a way similar to previous association rule mining process, if the count value of a state-a_{other} pair is greater than or equal to minimum support value determined before, it is assumed that the relevant state and a_{other} was experienced sufficiently. In this case, the hunter agent under consideration selects the action with the highest confidence value.
3. If the state-a_{other} pair is not experienced sufficiently, the agent selects its action with respect to $p(a_i \mid s)$ formula in a way similar to the previous one.
4. The hunter under consideration executes the action a_{self} selected in Step 2 or 3.
5. Simultaneously, the other hunter executes an action.
6. The environment changes to a new state s^1
7. The hunter under consideration receives a reward *r* from the environment and updates internal model database and the lookup table.
8. If the new state s^1 satisfies a terminal condition, then terminate the current trial. Otherwise, $s^1 \rightarrow s$ and go back to step 1.

In case the value of minimum support is set to 3000, the following rules for state S_1 can be obtained from Table 2: $S_0 \wedge a^2_{other} \rightarrow a^4_{self}$, $S_0 \wedge a^5_{other} \rightarrow a^5_{self}$

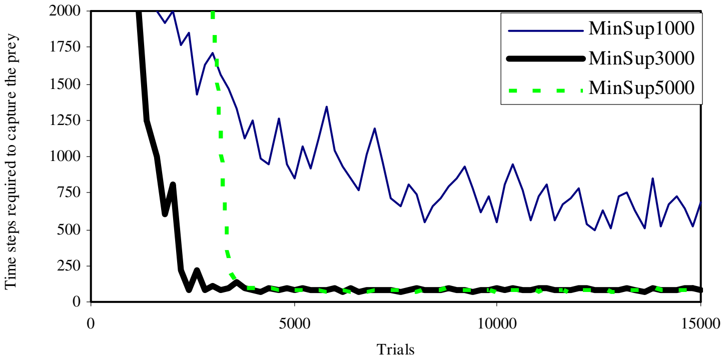


Fig. 3. Learning curves of the hunter agents with respect to randomly escaping prey

6 Experimental Results

We conducted some experiments to evaluate our algorithm learning by extracting association rules among states and actions. In all the experiments, the learning process consists of a series of trials. Each trial begins with a single prey and two hunter agents placed at random positions inside the domain and ends when either the prey is captured or at 2000 time steps. Upon capturing the prey, individual hunters immediately receive a reward of 100. Finally, the following parameters are used for the Q-learning process: 1) the learning rate α=0.8; 2) the discount factor γ=0.9; and 3) the initial value of the Q-function is 0.1,

The result is the average value over 10 distinct runs and the visual depth of the agent is set to 3nless specified otherwise. We run three different sets of experiments. In the first set of experiments, the escaping policy of the prey is random. Figure 3 shows the learning curves of the steps required to capture the prey with respect to different values of minimum support in the first set of experiments. In this experiment, the minimum confidence value is set to 0% until the number of occurrences of each state reaches the minimum support value. As can be seen easily from the figure, the learning curve (MinSup3000) in the case where minimum support value was set to 3000 converge to the near optimal solution faster than that of

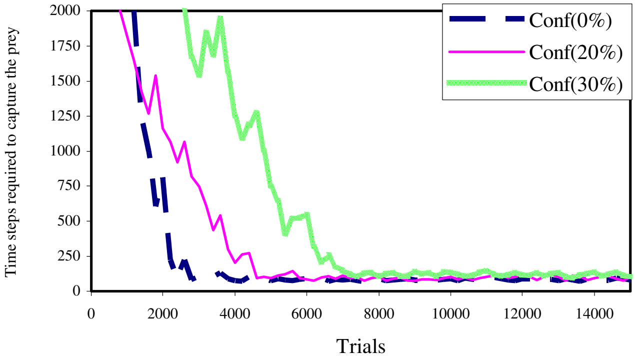


Fig. 4. Learning curves of the hunter agent with respect to different values of minimum confidence at MinSup3000

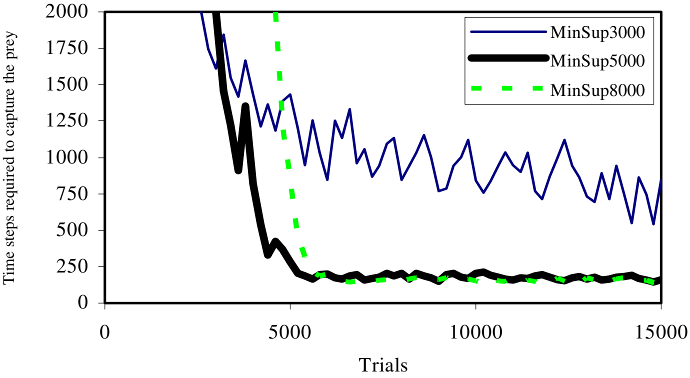


Fig. 5. Learning curves of the hunter agents with respect to the escaping prey with Manhattan-distance

MinSup5000, although the required number of steps for both curves are approximately the same at the learning moment. In addition, in case minimum support value was set to 1000, it is seen that the hunter agent almost learned nothing.

The learning curves found with respect to different values of minimum confidence at minimum support 3000 are given in Figure 4. It can be easily seen from the figure that the discovery of the environment is important up to a particular level of support value. In other words, the agent should be given the opportunity to discover its environment. Also, the solution with Conf(0%) not only converges faster but also captures the prey in less steps.

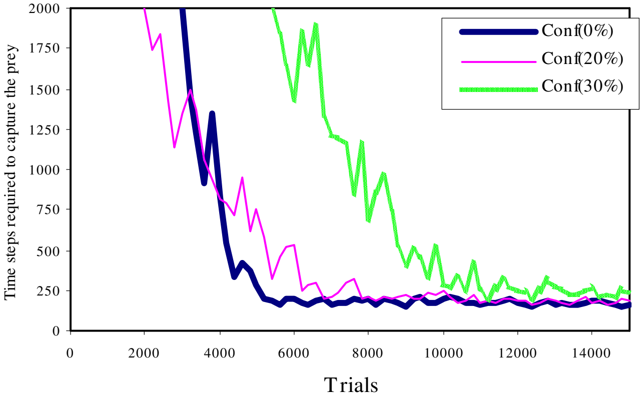


Fig. 6. Learning curves of the hunter agents with respect to the escaping prey with Manhattan-distance and different values of minimum confidence at MinSup5000

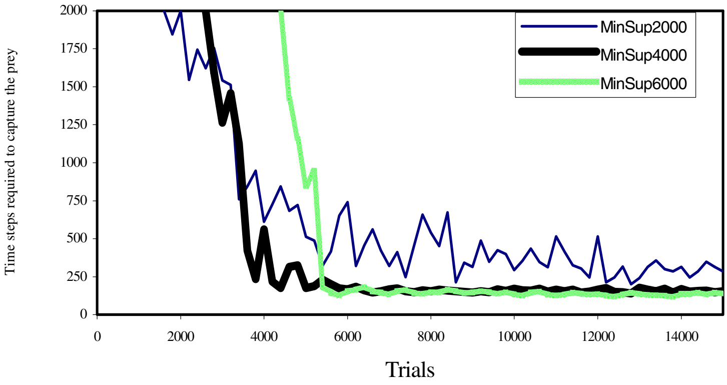


Fig. 7. Learning curves of the hunter agents with respect to the escaping prey with Manhattan-distance in case the visual depth of the hunter agents is set to 4

In the next set of experiment, we have determined the escaping policy of the prey to be Manhattan-distance to the located hunters. The results of this experiment are shown in Figure 5 and 6. Comparing with the previous experiment, it is seen that each state have to be experienced more in case of more complex multiagent environments. Also, it is enough to select minimum support value to be 5000 while the hunter agent cannot learn in less minimum support value, it is unnecessary to increase the minimum support value. As can be seen from Figure 5, a better solution with respect to the others is that of MinSup5000.

Figure 6 show the learning curves of the steps required to capture the prey with respect to different values of minimum confidence in the second set of experiments. Similarly, the curve with Conf(0%) gives a better solution.

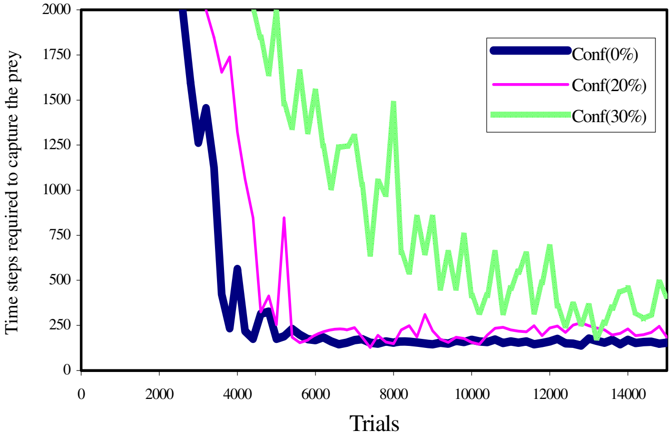


Fig. 8. Learning curves of the hunter agents with respect to Manhattan-distance and different values of minimum confidence at MinSup4000 in case the visual depth of the hunter agents is set to 4

In the final set of experiments, while the prey escapes by using Manhattan-distance, the visual depth of the hunter agents set to 4. In such a case, the hunter visualizes a larger area and hence captures the prey faster. The results of this experiment are shown in Figure 7 and Figure 8. Both figures are similar to the previous set of experiments.

7 Conclusions

In this paper, we proposed a novel multiagent reinforcement learning algorithm based on data mining. For this purpose, we introduced the term multiagent association rule mining to the literature. The agents within a multiagent environment collaboratively work updating joint internal model table and lookup table. Both tables are indeed a database where the past actions of the agents are hold. In this study, we extracted some rules among states and actions of the agents in real time. The aim of these rules is to estimate the actions of the other learning agents present in the environment and to ensure more appropriate action by considering the numbers of experiments of each state. Experimental results showed that the proposed learning approach could be used more effectively to achieve high quality optimal solutions.

References

[1] P. Stone and M. Veloso, “Multiagent systems: A Survey from a Machine Learning Perspective,” *Autonomous Robots*, Vol.8 No.3, 2000.

[2] F. Polat and A. Guvenir, “A Conflict Resolution Based Decentralized Multi-Agent Problem Solving Model," *Artificial Social Systems*, LNAI 130, Springer-Verlag, 1994.

[3] S. Benson, “Reacting, Planning and Learning in an Autonomous Agent," *Ph.D. thesis, Stanford University, Computer Science Department*, 1995.

[4] M. L. Littman, “Markov games as a framework for multi agent reinforcement learning,” *Proceedings of ICML*, pp.157-163. San Francisco, CA, 1994.

[5] T .W. Sandholm and R. H. Crites, “Multi agent reinforcement learning in the Iterated Prisoner’s Dilemma”, *Biosystems*, Vol.37, pp.147-166, 1995.

[6] M. Tan, “Multi-agent reinforcement learning: independent vs. cooperative agents,” *Proceedings of ICML,* pp.330-337, 1993.

[7] Abul, F. Polat and R. Alhajj, “Multiagent reinforcement learning using function approximation,” *IEEE TSMC*, Vol.30, No.4, 2000.

[8] J. Hu and M. P. Wellman, “Multiagent reinforcement learning: theoretical framework and an algorithm,” *Proceedings of ICML,* pp.242-250, 1998.

[9] Y. Nagayuki, S. Ishii and K. Doya, “Multi-Agent reinforcement learning: An approach based on the other agent’s internal model,” *Proceedings of the IEEE International Conference on Multi-agent system*s, pp.215-221, Boston, 2000.

[10] R. Agrawal, T. Imielinski and A. Swami. “Mining association rules between sets of items in large databases”, *Proceedings of ACM SIGMOD,* pp. 207-216, 1993.

[11] L. P. Kaelbling, M. L. Littman and A. W. Moore, “Reinforcement learning: A survey,” *Artificial Intelligence Research*, Vol.4, pp.237-285, 1996.

[12] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, Cambridge, MA: MIT Press, 1998.

[13] C. J. C. H. Watkins and P. Dayan. “Technical Note: Q-Learning,” *Machine Learning*, Vol.8, pp.279-292, 1992.

VisAR : A New Technique for Visualizing Mined Association Rules

Kesaraporn Techapichetvanich and Amitava Datta

School of Computer Science & Software Engineering,
University of Western Australia, Perth, WA 6009, Australia
{kes, datta}@csse.uwa.edu.au

Abstract. Many business organizations generate a huge amount of transaction data. Association rule mining is a powerful analysis tool to extract the useful meanings and associations from large databases and many automated systems have been developed for mining association rules. However, most of these systems usually mine many association rules from large databases and it is not easy for a user to extract meaningful rules. Visualization has become an important tool in the data mining process for extracting meaningful knowledge and information from large data sets. Though there are several techniques for visualizing mined association rules, most of these techniques visualize the entire set of discovered association rules on a single screen. Such a dense display can overwhelm analysts and reduce their capability of interpretation. In this paper we present a novel technique called *VisAR* for visualizing mined association rules. VisAR consists of four major stages for visualizing mined association rules. These stages include *managing association rules*, *filtering association rules of interest*, *visualizing selected association rules*, and *interacting with the visualization process*. Our technique allows an analyst to view only a particular subset of association rules which contain selected items of interest. VisAR is able to display not only many-to-one but also many-to-many association rules. Moreover, our technique can overcome problems of screen clutter and occlusion.

Keywords: Visualization, Visual Exploration, Association Rules, Market Basket.

1 Introduction

Many business organizations generate a huge amount of transaction data. Association rule mining is a powerful analysis tool to extract the useful meanings and associations for such data. Information visualization plays a major role to enhance the capability of analysts for analyzing large amounts of data. There is some existing research on visualization of association rules [1, 2, 3]. However, one of the drawbacks of these visualization techniques is that visualizing all mined association rules on a single screen reduces the interpreting and understanding ability of analysts. Typically, association rules generated by mining algorithms are difficult for users to understand due to their large number. Visualization allows users to visually analyze and understand the generated association rules.

This paper presents a new technique called *VisAR* for visualizing association rules derived from the mining process. We focus on reducing the complexity of visualizing

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 88–95, 2005.

large number of association rules on a single screen so that users are able to effectively understand and interpret information from a large number of association rules. We have also designed our system to eliminate occlusion. Users can explore association rules through their specified items of interest using an interactive visualization scheme. Our technique is able to represent both many-to-one and many-to-many association rules. Moreover, our technique allows users to select items appearing in the antecedents of association rules so that the users can view only the association rules containing their items of interest.

The rest of the paper is organized as follows. We discuss some previous work in Section 2, we explain the VisAR system in Section 3, the advantages of the VisAR system are presented in Section 4 and finally we conclude with some comments in Section 5.

2 Previous Work

2.1 Association Rules

An association rule [4] is formally described as a rule of the type A $\Rightarrow$ B where A is an item set called *antecedent*, *body*, or *left-hand side* (LHS) and B is an item set called *consequent*, *head*, or *right-hand side* (RHS). Each item set consists of items from a transactional database. Items existing in the antecedent are not in the consequent. In other words, an association rule is A $\Rightarrow$ B where A, B $\subset$ I and A $\cap$ B $= \phi$. I $= \{i_1, i_2, ..., i_n\}$ is a set of items in the transaction database where i_j, $1 \leq$ j $\leq$ n, is an item in the database that may appear in a transaction. The two common measures of interestingness are *support* and *confidence*. The support of a rule is defined as the percentage of frequency with which all items in the rule appear together. The confidence of the rule is the ratio of frequency of items in both antecedent and consequent (frequency of A and B) to frequency of items in antecedent appearing together. The probability of both support and confidence is, Support (A $\Rightarrow$ B) $=$ P (A $\bigcup$ B), Confidence (A $\Rightarrow$ B) $=$ P (B$|$A).

A term, *frequent itemset* [5] or *large itemset* [6] is used to define item sets whose number of co-appearances in the database is greater than a user specified support. In other words, these items are frequently purchased together and their occurrence is higher than the specified minimum support.

2.2 Related Work

Visualization techniques are integrated into data mining to help users understand the data as well as discover associations and pattern in the data. Various methodologies have been developed for visualizing association rules that are generated by an automated data mining algorithm. Prior research for visualizing association rules can be categorized into three main groups: *Table-based*, *Matrix-based*, and *Graph-based*.

First, Table-based techniques are the most common and traditional approaches to represent association rules in the form of a table. In general, the columns of a rule table represent the items, the antecedents and consequents, the support, and the confidence of association rules. Each row represents an association rule. Some examples of Table-based techniques can be found in SAS Enterprise Miner [7] and DBMiner [5].

Second, Matrix-based techniques such as MineSet [8] (2-D matrix), 3-D matrix [3], and grid represent the antecedents and consequents on a square grid based on the coordinate axes. In 2-D matrix, the height and colour of columns are used to represent the properties of the association rules such as support and confidence. Unlike 2-D matrix, 3-D matrix represents the relationships of rule-to-item rather than item-to-item. Although the 3-D matrix technique for association rule can visualize many-to-one association rules, it is difficult to display a large number of association rules due to occlusion. The higher columns representing antecedents and consequents of items with higher support and confidence can occlude the columns of low support and confidence.

The last group is Graph-based techniques such as Directed Graph. This technique uses nodes to represent the items and edges to represent the associations of items in the rules. For example, a rule A $\Rightarrow$ B is represented by a directed graph in which A and B are the nodes. The edge connecting A and B has the arrow pointing from the antecedent to the consequent of the rule. DBMiner [5] uses this technique, called *Ball graph*. The nodes in Ball graph are called *balls* whose sizes vary depending on the number of items represented by a ball.

The other techniques, not being specified above, are Interactive Mosaic plots [2], CrystalClear [1], a technique proposed by Zhao and Liu [9] and an integrated visual data mining tool by Techapichetvanich and Datta [10]. As the name suggests, the first technique applies Mosaicplot visualization technique to represent the relationships among items in each association rule from a contingency table instead of visualizing the results of association rule mining. CrystalClear uses an integrated technique of grid with tree based display to view the number of items and the lists of antecedents and consequents. The visualization technique proposed by Zhao and Liu [9] for association rules uses a line to represent each association rule. The x-axis represents time data and the y-axis represents the support or confidence value. Although this technique is designed to help users to understand discovered association rules through visual analysis of time, their visualization uses a technique similar to the Parallel Coordinates technique [11]. In practice, this technique generates occlusion and screen clutter [12] when visualizing a large number of association rules. Techapichetvanich and Datta [10] present an integrated framework for mining as well as visualizing association rules. Though their technique is useful for visual mining of association rules, it is not suitable for visualizing a large number of association rules.

Although Table-based, Matrix-based, and Graph-based techniques as well as some commercial visualization systems are capable of displaying mined association rules, they visualize all mined association rules in a single view. Typically, visualizing all association rules at once produces too much information and might also generate screen clutter. It is difficult for users to interpret and extract interesting association rules from such a dense display.

To effectively handle user interaction, an interactive tool must deal with many human factors such as consistency and feedback [13]. Our interactive technique takes into account some principles of interactive design such as consistency, providing feedback, reducing memorization, and ease of use or simplicity without extensive training of the user. In addition, an analyst has complete control in choosing the antecedents and consequents of each rule and the whole process is intuitively simple for the analyst.

3 The VisAR System

We divide the process of visualizing association rules in the VisAR system into four major stages including *Managing association rules*, *Filtering association rules of interest*, *Visualizing selected association rules*, and *Interactive visualization*.

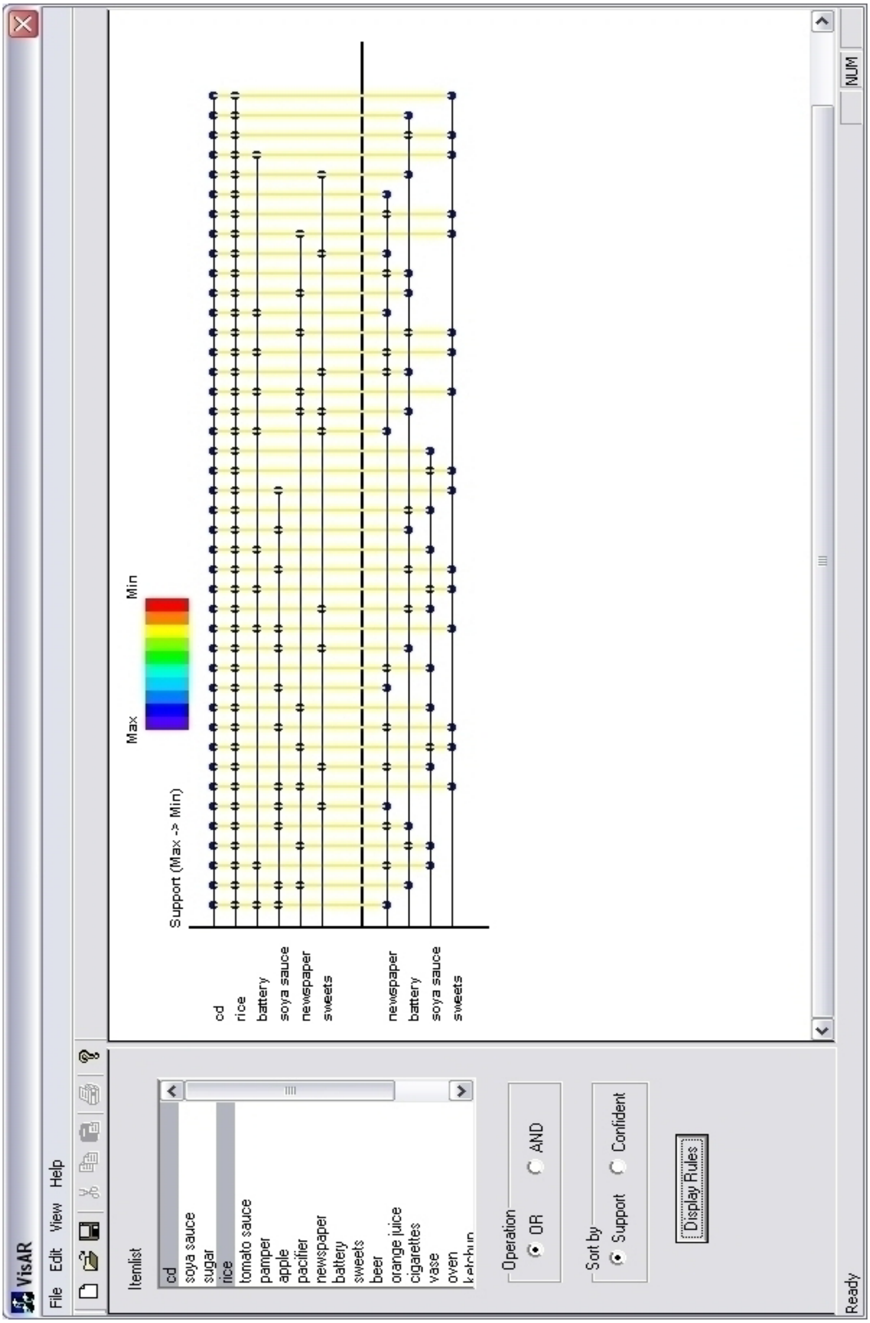
The first stage includes two processes: specifying and loading association rules that have been generated by an automated data mining tool. The specified association rules are first loaded into memory. The system counts all provided association rules and the number of distinct items in both antecedents and consequents as well as manages lists of items in antecedents and consequents. Then the system sorts the association rules according to the support values of individual association rules. We use the support as a default for sorting association rules.

The purpose of the second stage is to specify the items of interest in association rules and filter association rules according to the specified items. The user specifies the items of interest and the system filters the association rules for which the user-specified items exist in the antecedents.

The aim of the third stage is to visualize the association rules containing the selected items from the previous stage. Figure 1 shows the visualization result of the selected items, namely *cd* and *rice*, and the user interface of VisAR. After the user selects the items of interest, all association rules containing the specified items are visualized on the right panel. All antecedents and consequents of all qualified association rules are displayed along the y-axis. The antecedents are placed above and the consequents below the x-axis which is displayed as a bold and black line. The selected items of interest are displayed above other unselected items in the antecedents along the y-axis. In Figure 1, items in antecedents of all association rules are *cd*, *rice*, *battery*, *soya sauce*, *newspaper*, and *sweets*. The items in consequents of association rules are *newspaper*, *battery*, *soya sauce*, and *sweets*. The selected items are *cd* and *rice*. These two items are listed above battery, soya sauce, newspaper, and sweets. The system displays all association rules parallel to the y-axis by the sorted support values. Each rule is visualized by a vertical line parallel to the y-axis with circular dots representing items in each association rule. For example, in Figure 1, the first vertical line represents an association rule with five circular dots. Four dots representing cd, rice, battery, and soya sauce are in the antecedent section and another dot in the consequent section represents the newspaper item. The association rule is $\{$cd, rice, battery, soya sauce$\} \Rightarrow \{$newspaper$\}$. This is the rule with highest support among all rules that include *cd* and *rice* in the antecedent. Each confidence of an association rule is mapped to a color ramp so that the user can identify and group similar association rules by employing colors of confidences.

We use ten different colors for representing ten equal scales of either support or confidence in terms of percentage from zero to hundred. This color range has been designed to enhance the human ability of grouping items according to color. Blue represents the maximum value range, $90 - 100\%$, while red represents the minimum value range, $0 - 10\%$. All association rules in Figure 1 are in the same range and are mapped to yellow or the third colour range, i.e., $20 - 30\%$.

The last stage in VisAR is the interaction stage. This stage allows users to view details of each association rule and provides flexible adjustment to view association



Operation: ● OR ○ AND

Sort by: ● Support ○ Confident

Fig. 1. The left panel displays all antecedent items of association rules with the interactive options (operation and sorting) for visualizing association rules. The right panel visualizes association rules whose antecedent items are selected. *cd* and *rice* are the selected items in this figure. This visualization represents a selected OR operation

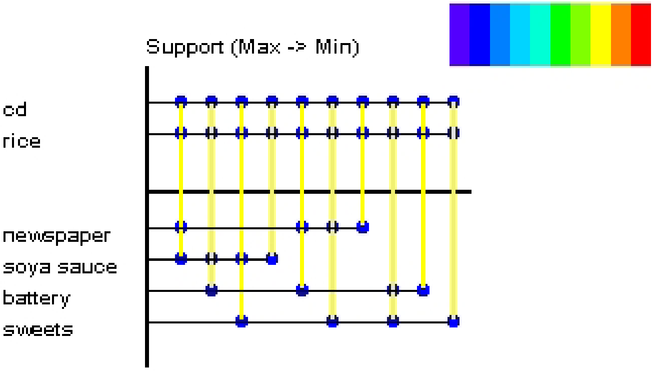


Fig. 2. Visualization of association rules from the selected items of interest in Figure 1. This visualization represents the selected operation AND which shows only association rules containing exactly the selected items, cd and rice in the antecedent

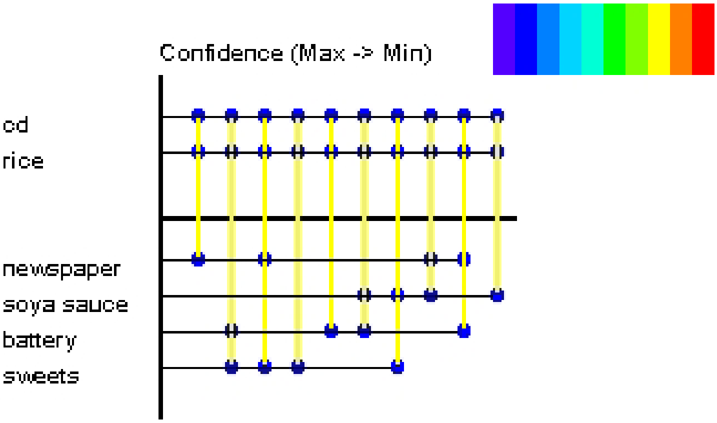


Fig. 3. This visualization represents the sorting of confidence which shows only association rules containing exactly the selected items, cd and rice. The color of vertical lines represent the support value of the association rules

rules. The support and confidence values of an association rule are shown when the user moves the mouse over the vertical line representing the rule. The user can change both defaults of the system to visualize association rules. The first option is to change the viewing of association rules to display only association rules containing exactly the specified items of interest. Figure 2 and Figure 3 shows association rules in which only *cd* and *rice* appear in the antecedents of the association rules. The default of our system is set to display association rules containing both specified items and all other items in each antecedent. The second option is to change the sorting order from support to confidence. The default of sorting in our system is according to support.

4 The Advantages of VisAR

The VisAR system can be considered as a hybrid of the Matrix-based and Graph-based techniques. Our technique has many advantages over the Table-based, ordinary Matrix based and Graph-based techniques including 3D visualization for mined association rules as follows.

- VisAR allows users to specify items of interest for visualizing association rules containing such items. This feature in our technique provides users to focus on specific association rules instead of viewing all association rules.
- VisAR has no limitation on the number of items in both the antecedent and the consequent to be displayed. The system can visualize both many-to-one and many-to-many association rules seamlessly.
- VisAR employs the benefits of both Matrix-based and Graph-based techniques for placing and linking items in association rules to solve the occlusion problem. The Matrix-based technique organizes the items like an array in which items are placed in rows while association rules are displayed by columns. The employed Graph-based technique links the same groups of items and the items of the same association rules so that users can easily identify the groups of items and individual association rules.
- There is no screen clutter or occlusion even when a large number of rules are displayed on the same screen.
- VisAR visually separates antecedent items and consequent items so that the users can clearly distinguish between the antecedent items and the consequent items of the association rules.
- The simplicity of VisAR helps the users to enhance their ability of interpretation. The users can identify groups of association rules which have close values of support or confidence.

5 Conclusion

Automatic association rule mining algorithms typically generate a large number of association rules which are difficult for analysts to understand and interpret. However, most of the visualization techniques display all mined association rules in a single screen. It is difficult for an analyst to interpret such large amount of information. In addition, some visualization techniques encounter problems of screen clutter and occlusion.

Our presented technique reduces the number of visualized association rules for effectively interpreting and understanding. The analysts can also choose to view specific association rules through their choice of items of interest. In addition, our visualization technique has overcome the problems of screen clutter and occlusion.

Acknowledgments. We have experimented on association rules derived from algorithms in CBA [14] as an association rule input for visualizing large number of association rules in our VisAR system.

References

1. Ong, K.H., Ong, K.L., Ng, W.K., Lim, E.P.: Crystalclear: Active visualization of association rules. In: International Workshop on Active Mining (AM-2002), in conjunction with IEEE International Conference On Data Mining. (2002)
2. Hofmann, H., Siebes, A.P.J.M., Wilhelm, A.F.X.: Visualizing association rules with interactive mosaic plots. In: Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM Press (2000) 227–235
3. Wong, P.C., Whitney, P., Thomas, J.: Visualizing association rules for text mining. In: INFOVIS. (1999) 120–123
4. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Proceedings of the 20th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. (1994) 487–499
5. Han, J., Kamber, M.: Data Mining Concepts and Techniques. Morgan Kaufmann (2001)
6. Savasere, A., Omiecinski, E., Navathe, S.B.: An efficient algorithm for mining association rules in large databases. In: Proceedings of the 21th International Conference on Very Large Data Bases, Morgan Kaufmann Publishers Inc. (1995) 432–444
7. (http://www.sas.com/technologies/analytics/datamining/miner/, S.I.I.)
8. (http://www.sgi.com/software/mineset.html, S.)
9. Zhao, K., Liu, B.: Visual analysis of the behavior of discovered rules. In: Workshop Notes in ACM SIGKDD-2001 Workshop on Visual Data Mining. (2001)
10. Techapichetvanich, K., Datta, A.: Visual mining of market basket association rules. In: Computational Science and Its Applications - ICCSA 2004, International Conference, Assisi, Italy, May 14-17, 2004, Proceedings, Part IV. Volume 3046 of Lecture Notes in Computer Science., Springer (2004) 479–488
11. Inselberg, A., Dimsdale, B.: Parallel coordinates for visualizing multidimensional geometry. Computer Graphics (Proceedings of CG International) (1987) 25–44
12. Techapichetvanich, K., Datta, A., Owens, R.: Hddv: Hierarchical dynamic dimensional visualization. In: Proc. IASTED International Conference on Databases and Applications. (2004) 157–162
13. Foley, J.D., van Dam, A., Feiner, S.K., Hughes, J.F.: Computer Graphics: Principles and Practice Second edition in C. Addison Wesley (1997)
14. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining. (1998) 80–86

An Efficient Algorithm for Mining Both Closed and Maximal Frequent Free Subtrees Using Canonical Forms*

Ping Guo, Yang Zhou, Jun Zhuang, Ting Chen, and Yan-Rong Kang

School of Computer Science, Chongqing University, 400044, Chongqing, China
guoping@cqu.edu.cn, zxhzgxzzqwxqzy@163.com

Abstract: A large number of text files, including HTML documents and XML documents, can be organized as tree structures. One objective of data mining is to discover frequent patterns in them. In this paper, first, we introduce a canonical form of free tree, which is based on the *breadth-first canonical string;* secondly, we present some properties of a closed frequent subtree and a maximal frequent subtree as well as their relationships*;* thirdly, we study a pruning technique of frequent free subtree and improvement on the mining of the nonclosed frequent free subtree; finally, we present an algorithm that mines all closed and maximal frequent free trees and prove validity of this algorithm.

1 Introduction

Many text files, such as HTML documents and XML documents, can be organized as tree structures. One objective of data mining is to discover frequent patterns in them. There are three categories of methods for mining frequent subtrees at present:

(1) The algorithms based on enumeration trees. In [9], Zaki presented an algorithm, *TreeMiner*, to discover all frequent embedded subtrees that preserve ancestor-descendant relationships. In [10] Asai et al presented an algorithm, *FREQT*, to discover frequent rooted ordered subtrees. For mining rooted unordered subtrees, Asai et al in [3] and Y.Chi et al in [4] both proposed algorithms based on enumeration tree growing. In [6], Y.Chi et al presented an algorithm, *CMTreeMiner*, to discover both closed and maximal frequent rooted unordered subtrees. (2) Apriori-like algorithms. In [1,7], Y.Chi et al studied the problem of indexing and mining free trees and developed an Apriori-like algorithm, *FreeTreeMiner*, to mine all frequent free subtrees. (3) Algorithms based on FP-tree. In [16], Xiao et al. presented such an algorithm called *PathJoin* mining maximal frequent subtrees. *PathJoin* uses a subsequent pruning that, after obtaining all frequent subtrees, prunes those frequent subtrees that are not maximal. At present, many studies focus on mining frequent subgraphs and closed frequent subgraphs [2,5,8,11,12,13].

The number of frequent subtrees usually grows exponentially with the tree size. This is the case especially when the transactions in the database are strongly correlated. The algorithm, *FreeTreeMiner*, which discovers all frequent free subtrees**,** can

* This work was supported by National Natural Science Foundation of China, NO. 50378093.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 96 – 107, 2005.

bring a mass of candidate trees. This phenomenon has two effects: first, there are too many frequent subtrees for users to manage and use; second, *FreeTreeMiner* is not able to handle frequent free subtrees with large size. To solve this problem, in this paper, based on *FreeTreeMiner*, we improve on mining technique for frequent free subtree, and generate all frequent free subtrees only by discovering closed and maximal frequent subtrees in a database of unrooted unordered trees.

This paper is organized as follows: In Section 2, we define free trees and a canonical form for them, introduce the concepts of closed frequent subtree and maximal frequent subtree, analyze their properties and relationships, and research a pruning technique of frequent free subtree. In section 3, we present an algorithm for mining closed and maximal frequent free trees, and prove validity of this algorithm. In section 4, we demonstrate experiment results with the new approach. Finally, in Section 5, we give the conclusion and future research directions.

2 Background

We define the *tree* as the acyclic connected graph. In this paper, each transaction *s* in the database is a labeled unrooted unordered tree.

2.1 The Canonical Form for Labeled Rooted Unordered Trees

Definition 1. The canonical form for labeled rooted unordered trees. For labeled rooted trees with height 0 (i.e., trees consisting of a single vertex), the canonical forms are the vertices themselves and the order among such trees is defined by the order of the vertex labels. For a labeled rooted tree with height h where h > 0, the canonical form is obtained by first normalizing all subtrees of the root then rearranging the subtrees in increasing order (from the left to the right in illustrating examples). From a rooted unordered tree we can derive many rooted ordered trees, as shown in Fig. 1. From these rooted ordered trees we want to uniquely select one as the canonical form to represent the corresponding rooted unordered tree.

First, we present two special symbols, ′$′and ′#′, which are not in the alphabet of vertex labels. In addition, we assume that (1) there exists a total ordering among vertex labels, and (2) ′#′ sorts greater than ′$′ and both sort greater than any other symbol in the alphabet of vertex labels. Second, we define the *breadth-first string encoding* for a rooted ordered tree through a *breadth-first* traversal and use ′$′ to partition the families of siblings and ′#′ to represent the end of the string encoding. Fig. 1 gives the breadth-first string encodings for each of the four trees. Finally, with the string encoding, we define the *breadth-first canonical string* of the rooted unordered tree as the minimal one among all possible breadth-first string encodings, and we define the *breadth-first canonical form* of a rooted unordered tree as the corresponding rooted ordered tree that gives the minimal BFCS. In Fig. 1, the breadth-first string encoding for *tree* (*a*) is the BFCS, and *tree* (*a*) is the BFCF for the corresponding labeled rooted unordered tree. The full construction procedure is given in [1].

Theorem 1. We can obtain the BFCF of a rooted unordered tree by sorting all the vertices of the rooted unordered tree level by level, top-down.

Proof. The full proof is given in [1].

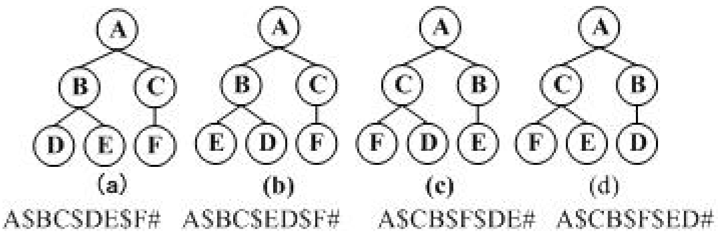


Fig. 1. Four rooted ordered trees are obtained from the same rooted unordered tree, and tree (a) is the BFCF for this rooted unordered tree. In tree (a), the rightmost path is the path "A-C-F" and the rightmost leaf is the vertex with label F. The extension points are the vertexes with label A, C, F

Theorem 2. The above BFCF construction procedure has time complexity $O(ck \log k)$, where k is the number of vertices the tree has and c is the maximal degree of the vertices in the tree.

Proof. The full proof is given in [1].

2.2 The Canonical Form for Free Tree

In the database, each Graph $G = [V, E]$ consists of a *vertex* set V, an *edge* set E. Because of its acyclicity, G has at least one vertex whose degree is 1. For each graph in the database, we label all the vertexes as following steps:

1. For each vertex in G, its label = -1.
2. For each vertex whose degree is 1, its label = 0.
3. If each *vertex* label in G is greater than or equal to zero, then finish the labeling.
4. Computing the degree of each vertex v in G: the degree of v is equal to the number of the vertexes that are v's neighbors and have the label of –1. The set of the vertexes whose degree is 0 and 1 is marked as $V = \{v_1, v_2, \dots, v_m\}$.
5. For each $v_i \in V$, labeling v_i with the maximal label of its neighbors plus one, whose label is greater than or equal to zero, then turning to 3.

This yields a labeled graph G=[V, E, L] which is a so-called *free tree* (L is the *vertex labels* set L). The node with the maximal label is called *center*. Namely, the center minimizes the maximal distance to all other nodes in the tree. As shown in Fig. 2, it is a well-known fact that every free tree has at most two *centers*.

Definition 2. The canonical form for free trees is the BFCF for the corresponding rooted unordered tree.

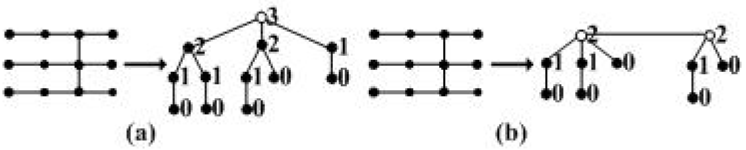


Fig. 2. Every free tree has either one (a) or two (b) centers. The centers are marked as '○'

We can build the canonical form of a free tree t according to Theorem 1 and Definition 2:

(1) We identify the canonical centers of *t*. If there is only one center, we have a unique root. If there are two centers, we remove the edge between the two centers, thus generating two subtrees. We order the two subtrees and construct the BFCF of them. The root of the smaller BFCS of the two subtrees is used as the root of the whole tree.

(2) Now that we have a rooted tree, we can construct the BFCF of this rooted tree according to Theorem 1.

Theorem 3. For an arbitrary tree, there is only one corresponding free tree.

According to Definition 2 and Theorem 3, we know the canonical form of a free tree is exclusive, so it can express tree in the database.

2.3 The Properties of Closed and Maximal Frequent Free Subtree

Each transaction *s* in the database *D* is a labeled rooted ordered tree that has been canonized. For a given tree t, we say t occurs in a transaction s if there exists at least one subtree of s that is isomorphic to t, and let $\sigma_t(s) = 1$, otherwise $\sigma_t(s) = 0$. We define the support of a tree t as

$$\sup \mathrm{p}(t) = \sum_{s \in D} \sigma_t(s)$$

A tree t is called frequent if supp(t) is greater than or equal to a minimum support (minsup). If a tree t is a subtree of another tree s, we say s is a supertree of t.

The frequent tree has the following property:

Property 1. Any subtree of a frequent tree is also frequent and any supertree of an infrequent tree is also infrequent.

We define a frequent tree *t* to be *maximal* if all of *t*'s supertrees is infrequent, and *closed* if none of *t*'s proper supertrees has the same support that *t* has. For a subtree *t*, we define the blanket *Bt* of *t* as the set of all supertrees of *t* that have one more vertex than *t*. In other words, $Bt = \{t' | \text{removing a leaf or the root from } t' \text{ can result in } t\}$.

With the blanket, we can define maximal and closed frequent subtrees in another equivalent way:

Property 2. A frequent subtree *t* is maximal iff for any $t' \in Bt$, $supp(t') < minsup$, a frequent subtree *t* is closed iff for any $t' \in Bt$, $supp(t') < supp(t)$.

Theorem 4. A maximal frequent subtree *t* is also closed.

Proof. Because *t* is a maximal frequent tree, $supp(t) \geq$ minsup, furthermore, for any $t' \in Bt$, $supp(t') < minsup$. Therefore, for any $t' \in Bt$, $supp(t') < supp(t)$, that is, *t* is also closed.

We can obtain the following property from Theorem 4:

Property 3. For a database *D* and a given *minsup*, let *F* be the set of all frequent subtrees, *C* be the set of closed frequent subtrees, and *M* be the set of maximal frequent subtrees, then $M \subseteq C \subseteq F$.

Property 4. Any frequent tree is a subtree of one (or more) maximal frequent tree(s).

Property 5. For a frequent tree t that is not closed, $supp(t) = \max\{supp(t') \mid t' \in Bt\}$.

According to the Properties 4 and 5, we can obtain all frequent subtrees from the set of maximal frequent trees; similarly, we can obtain all frequent subtrees with their supports from the set of closed frequent trees with their supports.

2.4 The Growth of Free Tree

For a free tree in its breadth-first canonical form (BFCF), we define the *rightmost leaf* as the last vertex according to the breadth-first traversal order, and *rightmost path* as the path from the root to the rightmost leaf. As is shown in Fig. 1 (a), the rightmost path is the path "*A*-*C*-*F*" and the rightmost leaf is the vertex with label *F*.

For a free tree *t,* the node in the rightmost path is marked as *extension point*. For the node *p1* which is an extension point of the free tree *t*, if the node $p2 \notin t$ *and* $p2 \in t'$ ($t' \in Bt$), we define the p-*extension* as the ordered node-node pair (*p1*:*p2*). If *t* is extended to $t'$by the extension (*p1*:*p2*), the set of graphs containing t'in the database is the so-called *support set of* (*p1*:*p2*). For an extension, the number of graphs of the support sets is the *frequency* of the extension. During the database scan, *TreeMiner* collects all extensions to all extension points *p* of a tree *t* for all occurrences of *t* in all graphs in the *extension table*. The extension table is organized as a table with two columns and each row representing one particular extension (*p1*:*p2*). It stores for each row the name of the extension (*p1*:*p2*) in the first column, and in the second column the support set of the extension (*p1*:*p2*). Thus, one can determine the frequency of t extended by (*p1*:*p2*) solely by examining the extension table. After the database scans, *TreeMiner* uses the extension table to generate reasonable extension candidates, and extends the free tree to new candidate trees by adding one new vertex.

As an example, consider the setting in Fig. 3: we are looking for frequent free subtrees with a frequency of at least two in the database containing the three graphs *g1*, *g2*, and *g3*. During the database scan we wish to determine the frequency of the free tree *t*, which has two extension points 'A' and 'C'. Obviously, the frequency of *t* is three, because it occurs in all three graphs (one occurrence in *g1, g2* and *g3*). During the scan all possible extensions of *t* in *g1, g2,* and *g3* are entered in the extension table. Thus, after the scan the extension table looks as Table 1. From this table we generate all extension candidates containing (*A*:*D*) and (*C*:*E*) and their support sets.

Theorem 5. For a frequent free subtree *t*, there exists a $t' \in Bt$. For each occurrence of *t* in a transaction of the database, there is at least one corresponding occurrence of t'. If $t' \backslash t$ is at location of case (a) ($t' \backslash t$ is the root of t'), (b) ($t' \backslash t$ is attached to a vertex of *t*

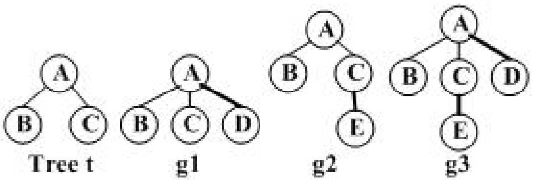


Fig. 3. A free tree t has two extension points 'A' and 'C' and two extensions containing (*A*:*D*) and (*C*:*E*)

Table 1. The extension table

Extension	Support Set
(*A*:*D*)	{*g1*, *g3*}
(*C*:*E*)	{*g2*, *g3*}

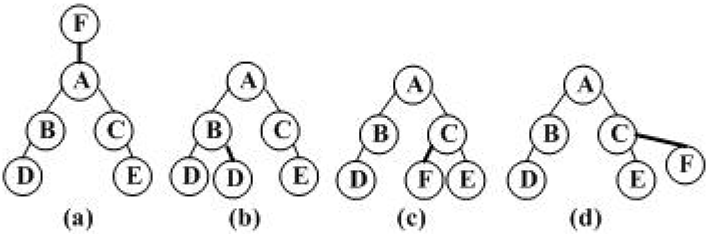


Fig. 4. For a frequent free subtree *t*, *t*′*t* is at location of case (a), (b), or (c), neither *t* nor any supertree of *t* correspond to closed or maximal frequent subtrees

that is not the extension point of *t*), or (c) (*t*′*t* is attached to a extension point of *t*, and *t*′*t* is not the new rightmost leaf of *t*′) in Fig. 4, then neither t nor any supertree of *t* correspond to closed or maximal frequent subtrees, therefore *t* together with all of its supertrees can be pruned. (*t*′*t* is the additional vertex of *t*′that is not in *t* (*t*′∈*Bt*)).

Proof. For case (a), for all occurrences of *t* in the database, the roots of the occurrences have parents with the same label-*t*′*t*. So for any *t*′that is the supertree of *t*, the roots of all the occurrences of *t*′in the database have parents with the same label-*t*′*t* also. Therefore *t*′cannot be closed because we can extend *t*′by adding the new vertex *t*′*t* to get *t*″*that* is a supertree of *t*′with the same support as *t*′. Similar idea applies to case (b) and case (c), except that we need to prove that *t*′*t* will not be used by any *t*′ that is a supertree of *t*. This is obvious, because *t*′is obtained from *t* by adding more rightmost leaves, and we know that the vertices shown in case (b) and case (c) will never be the newly added rightmost leaf.

Theorem 6. For a nonclosed frequent free subtree *t*, there exists a blanket *Bt* of *t*(*Bt*≠*Φ*). If *t*′that is a supertree of *t* with maximal support in the *Bt* is obtained by adding a new vertex *u* to the extension point *v,* then any closed frequent supertree of *t* must contain the node *u*, and the node *u* must be attached to the extension point *v*.

Proof: Applying negative approach to prove it. Because *t* is a nonclosed frequent free subtree, according to the Property 5, we can conclude that *supp*(*t*′) is equal to supp(*t*). If we assume that any closed frequent supertree *T* of *t* doesn't contain the node *u,* then we can always obtain *T*′*that is* a supertree of *T* by adding a new vertex *u* to the extension point *v* of *T,* and *supp*(*T*′) is equal to *supp*(*T*). This is contrary to the truth that *T* is closed. To sum up, the conclusion of theorem is right.

According to Theorem 5 and 6, we can obtain the method of the growing of the frequent free subtree:

Starting from a frequent node *p1* (=*t*), we need to collect all extensions (*p1*:*p2*) for all occurrences of *v* in all graphs in the extension table. (1) If there is at least one corresponding occurrence of t' for each occurrence of *t* in a transaction of the database ($t' \in Bt$) such that $t'\backslash t$ is at location of case (a), (b) or (c) in Theorem 5, then stop the growing of *p1*. (2) If *p1* is both closed and maximal, then also stop the growing of *p1*. (3) If *p1* is closed and not maximal, for each extension (*p1*:*p2*) in *ext,* we obtain a supertree of *p1* by adding a vertex *p2* to *p1* so that *p2* becomes the new rightmost leaf of the new BFCF. Then we check if the resulting new tree is in the canonical form or not. (4) If *p1* is not closed, we only select a vertex *p2* from *ext*, which makes the extensions (*p1*:*p2*) have the maximal frequency in *ext* and obtain a supertree of *p1* by adding a vertex *p2* to *p1* so that *p2* becomes the new rightmost leaf of the new BFCF. Then we check if the resulting new tree is in the canonical form or not. Finally, we prune the infrequent 2-trees and generate *frequent 2-trees* containing the vertex *p1*. Similarly, we can obtain *frequent k-trees* containing the vertex *p1*.

3 The Algorithm of Mining Closed and Maximal Frequent Free Subtrees

This algorithm is made up of three modules: (1) *DBScan* scans the database and collects all extensions for all occurrences of the free tree *t* in all graphs in the extension table. (2) *TreeGrow* makes use of the extension table and efficient pruning to collect closed and maximal frequent free subtrees containing *t*. (3) *TreeMiner* circularly calls *TreeGrow* to generate all closed and maximal frequent free trees.

3.1 The DBScan Algorithm

The purposes of this algorithm are to scan all graphs containing the free tree *t* in the database and collect all extensions for all occurrences of *t* in all graphs in the extension table. The algorithm starts with an empty table and adds a new row for each new extension that is encountered during the database scan, and then adds the graph(s) containing this extension to the support set. After the database scan, one can determine the frequency of *t* extended by (*p1*:*p2*) solely by examining the extension table.

```
Algorithm 1. DBScan(t, d)
  Input: Free tree t, database d.
  Output: extension table ext.
    ext ← empty extension table;
    for each graph g in d do
      for each occurrences o of t in graph g do
        for each extension point p1 of t do
          for each extensions p2 to p1 at o do
            if extension (p1 : p2) is not present in ext then
              Insert a row for (p1 : p2) into ext with empty support set;
              Add g to the support set in row (p1 : p2);
  return ext.
```

3.2 The TreeGrow Algorithm

The purposes of this algorithm are to scan the database and make use of efficient pruning to collect closed and maximal frequent free subtrees containing the free tree *t*. First, the algorithm calls the *DBScan* algorithm to scan the database and generates the extension table of *t*. Secondly, according to Theorem 5, the algorithm prunes all nonclosed frequent free subtrees and collects closed and maximal frequent free subtrees containing the free tree *t* in the database. Finally, according to Theorem 6, the algorithm extends *t* to *t′* that is a supertree of *t* and recursively calls the *TreeGrow* algorithm.

```
Algorithm 2. TreeGrow(t, d, CL, MX, minsup)
  Input:Free tree t, database d, closed frequent free
        subtrees set CL, maximal frequent free subtrees
        set MX, minimum support minsup.
  Output: CL and MX.
     ext ← DBScan(t, d);
     if there is at least one corresponding occurrence
        of t′ for each occurrence of t in a transaction
        of the database (t′∈Bt) such that t′\t is at lo-
        cation of case (a), (b) or (c) in Theorem 5 then
       return CL, MX;
     if each extension (p1:p2) support in ext < the sup-
        port of t then
       CL ← CL ∪ t;
     if each extension (p1:p2) support in ext < minsup
        then
       MX ← MX ∪ t;
       return CL, MX;
     else
       for each extension (p1:p2) in ext do
         t' ← t plus vertex p2, with p1 as p2's parent;
       if supp(t') ≥ minsup then
         TreeGrow(t', d, CL, MX, minsup);
       Else
         if the support of extension (p1:p2) is maximal
            in all extensions (p1:p2) support in ext then
           t' ←t plus vertex p2 , with p1 as p2's parent;
       if supp(t') ≥ minsup then
         TreeGrow(t', d, CL, MX, minsup);
     return CL, MX;
```

3.3 The TreeMiner Algorithm

The purpose of this algorithm is to mine all frequent free subtrees. First, the algorithm initializes the set of closed frequent free subtrees and the set of maximal frequent free subtrees, then searches for all frequent nodes in the database, and generates all frequent free trees with one node from this information. Second, the algorithm circularly calls the *TreeGrow* algorithm with those trees and generates all closed and maximal frequent free trees. Finally, the algorithm outputs all frequent free subtrees for a given database.

```
Algorithm 3. TreeMiner(d, minsup)
  Input: Database d, minimum support minsup;
  Output: closed frequent free subtrees set CL; maximal
          frequent free subtrees set MX;
    CL ← ∅, MX ← ∅;
    frequentLabels ← set of all node labels, which ap-
    pear in at least minsup graphs in d;
    frequent 1-trees ← set of all trees with one node
    and a label from frequentLabels;
    for each t in frequent 1-trees do
    TreeGrow(t, d, CL, MX, minsup);
  Return CL, MX;
```

From the above analysis, we can conclude that the demonstration of validity of the *TreeMiner* algorithm is made up of Theorem 5 and 6.

4 Experiments

We performed extensive experiments to evaluate the performance of the *TreeMiner* algorithm by comparing the performance of *TreeMiner* with that of *FreeTreeMiner*. All experiments were done on a 1.7GHz Intel Pentium IV PC with 256MB main memory, running Windows XP operating system. All algorithms were implemented in C++ and compiled using the Visual C++ 6.0 compiler. The datasets are the synthetic datasets.

4.1 Synthetic Datasets

To generate synthetic data that reflect properties of real applications, instead of generating datasets of trees arbitrarily, we use the universal Internet topology generator BRITE [17], developed by Medina et al at Boston University, which generates random graphs simulating Internet topologies with some specific network characteristics, such as the link bandwidth. The synthetic datasets created by BRITE has the following characteristics: |*D*| (=1000, 5000) is the number of transactions in the database and |*S*| (=5-40, 13-200) is the minimum support. The number of distinct edge and vertex labels is controlled by the parameter |*L*| (=10-15), which is both the number of distinct edge labels and the number of distinct vertex labels. |*T*| (=5-20) is the size of each transaction in the database.

4.2 Experimental Results

FreeTreeMiner is an algorithm for mining all frequent free subtrees [1]. Because the number of frequent subtrees usually grows exponentially with the tree size, *mining* all frequent free subtrees can bring a mass of candidate trees and a very expensive operation. To solve this problem, based on *FreeTreeMiner*, we improve on indexing and mining technique for frequent free subtree. In addition, by mining only closed and maximal frequent subtrees, we do not lose information.

Table 2. Minimum support, number of closed frequent free subtrees, number of maximal frequent free subtrees and runtime of *TreeMiner*

TreeMiner (|D| =1000)			
|S|	|C|	|M|	Time (s)
40	20	20	22
10	419	399	84
9	421	399	84
8	432	402	85
7	466	409	89
6	522	423	96
5	2399	1190	341

Table 2 shows the experimental results of *FreeTreeMiner* for varying minimum support thresholds. Table 3 compares the performance of *FreeTreeMiner* with that of *TreeMiner.*

As we can see from Table 2 and 3, we can observe a number of interesting points about the performance of *TreeMiner* by comparing the results of *TreeMiner* on two datasets. First, the running time increases linearly with the number of closed frequent free subtrees or the number of maximal frequent free subtrees. Second, as the number of transactions |*D*| increases, the total running time increases, because there are more automorphisms and subtree isomorphisms, and thus there are longer generating time, checking time and sorting time, while the number of closed frequent free subtrees also increases.

From Table 3, we can observe a number of interesting points about the performance of *TreeMiner a*nd *FreeTreeMiner.* First, the total number of all frequent free subtrees grows exponentially but the number of closed frequent free subtrees and

Table 3. Minimum support, number of closed frequent free subtrees, number of maximal frequent free subtrees, runtime of *TreeMiner*, number of all frequent free subtrees obtained by *FreeTreeMiner and* runtime of *FreeTreeMiner*

TreeMiner (|D| =5000)				FreeTreeMiner (|D| =5000)	
|S|	|C|	|M|	Time (s)	|A|	Time (s)
200	20	20	64	20	67
150	81	64	74	392	291
100	420	400	133	3011	1726
25	423	400	134	3019	1726
20	488	414	143	4633	3821
19	556	444	156	8832	12338
18	664	505	178	19667	40047
17	849	623	213	41189	To exhaust all memory
16	1115	810	273		
15	1574	1190	365		
14	11253	8965	6312		
13	34647	23989	49647		

maximal frequent free subtrees do not. Therefore, as shown in Table 3, the total running time of *TreeMiner* grows in polynomial fashion while that of *FreeTreeMiner* grows exponentially.

Second, the exponential explosion of *FreeTreeMiner* becomes apparent at a minimum support level of 17, because there are more automorphisms and subtree isomorphisms, and thus there are longer generating time, checking time and sorting time. However, by uniting the vertices that have the same label and father, *TreeMiner* decreases automorphisms and subtree isomorphisms, and thus there are shorter generating time, checking time and sorting time.

Finally, although *FreeTreeMiner* is very efficient for large minimum support, as minimum support decreases beyond some reasonably small value, it becomes obvious that *FreeTreeMiner* suffers from exponential explosion while *TreeMiner* does not. This is because the total number of all frequent free subtrees is much more than that of closed frequent free subtrees or maximal frequent free subtrees for small minimum support. As we can see from Table 3, for a minimum support level of 17, *FreeTreeMiner* exhausts all available memory while it took *TreeMiner* only 213 seconds!

5 Conclusion

In this paper, we have studied the issue of mining frequent free subtrees from databases of labeled unrooted unordered trees. We have presented *TreeMiner*, a new efficient algorithm that mines both closed and maximal frequent free subtrees, which make very efficient use of a canonical form for free tree (BFCF). Based on the canonical form, *TreeMiner* combines the vertices that have the same label and father, decreases generating time, checking time and sorting time, and prunes those frequent subtrees that are not closed or maximal. We have proven that the set of maximal frequent free trees or the set of closed frequent free trees is only a subset of all frequent free subtrees. But we can obtain all frequent free subtrees with their supports from them.

The experiments showed that *TreeMiner* performs in polynomial fashion instead of the exponential growth shown by *FreeTreeMiner*. Whether at large minimum support or small one, *TreeMiner* is more efficient than *FreeTreeMiner*. We plan to extend our work in several directions in the future. First, the bottleneck of *TreeMiner* is the subtree isomorphism checking, because it needs a number of comparisons of vertex label and requires large amount of memory. Our next implementation will improve on the growing technique of free tree. Second, in many applications, such as chemical compounds, there are 2D or 3D coordinates for vertices of trees. In the future, we will include this geometric information in our implementation to see if such information will improve the performance of our algorithm.

References

1. Y. Chi, Y. Yang, and R.R. Muntz. Indexing and mining free trees. In Proceedings of the 2003 IEEE Int. Conf. on Data Mining, 2003,509-512.
2. J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In Proc. of the 2003 Int. Conf. on Data Mining, 2003, 549-552.

3. T. Asai, H. Arimura, T. Uno, and S. Nakano. Discovering frequent substructures in large unordered trees. In Proc. of the 6th International Conference on Discovery Science, 2003, 47-61.
4. Y. Chi, Y. Yang, and R.R. Muntz. HybridTreeMiner: An Efficient Algorithm for Mining Frequent Rooted Trees and Free Trees Using Canonical Form. In Proceedings of the 16th International Conference on Scientific and Statistical Database Management, 2004, 11-20.
5. X. Yan and J. Han. CloseGraph: Mining closed frequent graph patterns. In Proc.of the 2003 Int. Conf. Knowledge Discovery and Data Mining, 2003, 286—295.
6. Y. Chi, Y.Yang, and R.R. Muntz. CMTreeMiner: Mining Both Closed and Maximal Frequent Subtrees.In Proceedings of 8th Pacific-Asia Conference, Advances in Knowledge Discovery and Data Mining 2004, 2004, 63-73.
7. U. Rückert and S. Kramer. Frequent Free Tree Discovery in Graph Data.In Proceedings of the 2004 ACM symposium on Applied computing, 564-570.
8. X. Yan and J. Han. gSpan: Graph-based substructure pattern mining. In Proc. of the 2002 Int. Conf. on Data Mining, 2002, 721-724.
9. M.J. Zaki. Efficiently mining frequent trees in a forest. In Proceedings of 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, 71-80.
10. T. Asai, K. Abe, S. Kawasoe, H. Arimura, H. Satamoto, and S. Arikawaient substructure discovery from large semi-structured data. In Proceedings of the 2nd SIAM Int. Conf. on Data Mining, 2002. 158-174
11. M. Kuramochi and G. Karypis. Frequent subgraph discovery. In Proc. of the 2001 IEEE Int. Conf. on Data Mining, 2001, 313-320.
12. L.D. Raedt and S. Kramer. The level-wise version space algorithm and its application to molecular fragment finding. In Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence, 2001, 853-862.
13. A. Inokuchi, T.Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. In Proceedings of the 4th European Conference on Principles and Practice of Data Mining and Knowledge Discovery, 2000, 13–23.
14. N. Pasquier, Y. Bastide, R. Taouil, and L. Lakhal. Discovering frequent closed itemsets for association rules. In Proceedings of the 7th Int'l. Conf. on Database Theory, 1999, 398-416.
15. J. Hein, T. Jiang, L. Wang, and K. Zhang. On the complexity of comparing evolutionary trees. Discrete Applied Mathematics, 1996, Vol.71,153-169.
16. Y. Xiao, J-F Yao, Z. Li, and M. Dunham. Effcient data mining for maximal frequent subtrees. In Proceedings of the 2003 IEEE Int. Conf. on Data Mining, 2003, 379-386.
17. A. Medina, A. Lakhina, I. Matta, and J. Byers, BRITE: An approach to universal topology generation, In Proceedings of the International Workshop on Modeling, Analysis and Simulation of Computer and Telecommunications Systems, 2001, 346-356.

E-CIDIM: Ensemble of CIDIM Classifiers*

Gonzalo Ramos-Jiménez, José del Campo-Ávila, and Rafael Morales-Bueno

Departamento de Lenguajes y Ciencias de la Computación,
E.T.S. Ingeniería Informática. Universidad de Málaga,
Málaga, 29071, Spain
{ramos, jcampo, morales}@lcc.uma.es

Abstract. An active research area in Machine Learning is the construction of multiple classifier systems to increase learning accuracy of simple classifiers. In this paper we present E-CIDIM, a multiple classifier system designed to improve the performance of CIDIM, an algorithm that induces small and accurate decision trees. E-CIDIM keeps a maximum number of trees and it induces new trees that may substitute the old trees in the ensemble. The substitution process finishes when none of the new trees improves the accuracy of any of the trees in the ensemble after a pre-configured number of attempts. In this way, the accuracy obtained thanks to an unique instance of CIDIM can be improved. In reference to the accuracy of the generated ensembles, E-CIDIM competes well against bagging and boosting at statistically significance confidence levels and it usually outperforms them in the accuracy and the average size of the trees in the ensemble.

1 Introduction

Classification and prediction tasks are two of the most popular activities in Machine Learning. There are many approaches that try to extract knowledge from data. These approaches are very diverse, but one of the most active research area is composed by multiple classifier systems. They have benefited from the idea of using a committee or ensemble of models to do cited tasks.

In the literature we can find many approaches to define a multiple classifier system. Thus, we have methods that mainly reduce variance, such as bagging [1] or boosting [2], and methods that reduce bias, such as stacked generalization [3]. Other multiple classifier methods, such as cascading [4], generate new attributes from the class probability estimation. Delegating [5] is another method and it works with examples in the dataset, using part of them in each classifier and delegating the rest of examples to the next classifier. In short, there are many methods for generating multiple models.

Voting is the most common way used to combine classifiers. Thus, the errors introduced by one classifier can be corrected with the good decisions made by the

* This work has been partially supported by the MOISES project, number TIC2002-04019-C03-02, of the MCyT, Spain.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 108–117, 2005.

other classifiers. Several variants of voting have been proposed. The most simple voting method is uniform voting, where every classifier has the same importance. Weighted voting is another method where each basic classifier has an associated weight that increase or decrease its importance.

Many kind of models can take part into a multiple classifier system. Decision trees are widely used by Machine Learning community (CART [6], ID3 [7], C4.5 [8], ITI [9] ...) and they have some positive characteristics. They have the ability of splitting the hyperspace into subspaces and fitting each space with different models. They also have a good feature: the understandability.

Taking this into account, we propose a multiple classifier system (called E-CIDIM) which basic classifiers are decision trees. These decision trees are induced by CIDIM (Control of Induction by sample DIvision Method) [10], an algorithm that will be explained in short.

The paper is organized as follows. In Section 2 we will briefly describe CIDIM and its utilization in a multiple classifier system. We will introduce E-CIDIM and how this method can take advantages from the design of CIDIM in section 3. Some experimental results are shown in section 4. Finally, in section 5, we summarise our conclusions and suggest future lines of research.

2 CIDIM

CIDIM (Control of Induction by sample DIvision Method) [10] was developed to induce accurate and small decision trees. It uses three ideas to reach this goal: it divides the training set into two subsets, it groups values and it defines an internal bound condition for expansion. Let us comment these characteristics with more detail:

- The top down induction of decision trees (TDIDT) algorithms [7, 8], generally, divide the set of examples into two subsets: the training subset (used to induce the tree) and the test subset (used to test the results). CIDIM makes an additional division. It divides the training subset into two new subsets with the same class distribution and similar size: the construction subset (called CNS) and the control subset (called CLS). Every node has its corresponding CNS and CLS subsets. When an expansion is made, CNS and CLS subsets of the parent node are divided into multiple CNS and CLS subsets, each one corresponding to the appropriate son node. Thus, the size of CNS and CLS decrease as the node is deeper in the tree.
- Let us consider an attribute with values $\{O_1, O_2, ..., O_n\}$. If this attribute is selected to be expanded, one branch is added to the tree for each possible value. Thus, it is necessary to know all the possible values of the attribute if it is a nominal attribute. If the attribute is continuous, a previous division into intervals must be made. CIDIM uses a greedy algorithm to find groups of consecutive values. It is based on a recursive splitting of the values in groups. Initially there is an unique group with all the values of the attribute that is being considered. In each step, CIDIM evaluates if the split will produce

an improvement. The process continues until each group has only one value or until there is no improvement since previous split. The couple (attribute, division) that produces the best improvement is selected to expand the node.

- Usually, the expansion of one tree finishes when all examples associated with a node belong to the same class, yielding too large trees. In order to avoid this overfitting, external conditions are considered by different algorithms (C5, an updated version of C4.5, demands that at least two branches have at least a pre-configurable number of examples). CIDIM uses the following as an internal condition: a node is expanded only if its expansion improves the accuracy calculated on CLS. Tree expansion supervision is local for every node and it is driven by two indexes: the absolute index I_A and the relative index I_R (see equations in (1)). For every step, a node is expanded only if one or both indexes are increased. If one index decrease, expansion is not made. The absolute and relative indexes are defined as

$$I_A = \frac{\sum_{i=1}^{N} CORRECT(e_i)}{N} \qquad \text{and} \qquad I_R = \frac{\sum_{i=1}^{N} P_{C(e_i)}(e_i)}{N} \; . \tag{1}$$

where N is the number of examples in CLS, e a single example, $C(e)$ the class of the e example, $P_m(e)$ the probability of m class for the e example, and $CORRECT(e) = 1$ if $P_{C(e_i)} = max\{P_1(e), P_2(e), ..., P_k(e)\}$ or 0 if another case.

A description of CIDIM can be seen in the Figure 1.

```
1. CNS (ConstructioN Subset) and CLS (ControL Subset) are obtained
   by a random dichotomic division of the set of examples used to induce the tree
2. for each non-leaf node do:
   2.1. Select the best splitting (considering a given disorder measure)
   2.2. if splitting does not improve prediction
             then Label node as a leaf-node
   2.3. if splitting improves prediction
             then Expand node
```

Fig. 1. CIDIM algorithm

Decision trees generated by CIDIM are usually smaller than those obtained with other TDIDT algorithms. This allows the induction of more general trees that will also be more understandable for human experts. At the same time, accuracy of the induced trees keeps similar to decision trees induced by other TDIDT algorithms.

CIDIM can be applied to any problem with a finite number of attributes. These attributes must be nominal and can be ordered or not. If the problem has continuous attributes, they can be discretized, resulting ordered nominal attributes. The class attribute must have a finite number of unordered classes.

These advantages have been used to solve real problems, such as system modelling [11] or modelling of prognosis of breast cancer relapse [12].

3 E-CIDIM

Improving the generalization of classifiers is an aim of Machine Learning. Voting methods tries to achieve this improvement. Many algorithms have been developed [13, 14, 15, 1] and numerous studies have been made about them [16, 17, 18].

We can divide these algorithms into two categories: those that change the dataset distribution depending on the previous steps of the algorithm (usually called [illegible] [13, 14, 15]) and those that do not change the cited distribution (usually called [illegible] [1]).

E-CIDIM is based on the bagging scheme. It uses CIDIM as the basic classifier, and it induces decision trees to build the ensemble. CIDIM is a "randomized" algorithm that makes a random division of the training set into two subsets (CNS and CLS) and this suits very well with the bagging scheme.

Two parameters are needed by E-CIDIM to induce the multiple classifier system: the maximum number of trees in the ensemble ($max\_number\_of\_trees$) and the number of failed attempts before stopping ($number\_of\_failed\_attempts$). For prediction process, a voting method must be selected.

```
In: $E$, $max\_number\_of\_trees$, $number\_of\_failed\_attempts$

1. $Ensemble = \varnothing$
2. $Initial\_number\_of\_trees = \lceil max\_number\_of\_trees/2 \rceil$
3. for 1 to $Initial\_number\_of\_trees$ do:
   3.1. $New\_tree \leftarrow$ Induce new tree with CIDIM using $E$
   3.2. $Ensemble = Ensemble \cup New\_tree$
4. $Failed\_attempts = 0$
5. while $Failed\_attempts < number\_of\_failed\_attempts$ do:
   5.1. $Worst\_tree = \{x \in Ensemble | success\_rate(x) < success\_rate(y)$
                    $\forall y \in Ensemble \wedge x \neq y\}$
   5.2. $New\_tree \leftarrow$ Induce new tree with CIDIM using $E$
   5.3. if $success\_rate(New\_tree) > success\_rate(Worst\_tree)$ then:
        5.3.1. $Failed\_attempts = 0$
        5.3.2. $Ensemble = Ensemble \cup New\_tree$
        5.3.3. if $|Ensemble| > max\_number\_of\_trees$ then:
               5.3.3.1. $Ensemble = Ensemble - Worst\_tree$
   5.4. else
        5.4.1. $Failed\_attempts = Failed\_attempts + 1$

Out: $Ensemble$
```

Fig. 2. E-CIDIM algorithm

Firstly, E-CIDIM initializes the ensemble ($Ensemble$) with some decision trees induced by CIDIM using the training set (E). After initialization there are half as decision trees as it is indicated by the parameter for maximum number of trees ($max\_number\_of\_trees$). New decision trees are induced by

CIDIM using the same training set (E). These decision trees are usually different because CIDIM makes a random dichotomic division and the induction is made with different subsets (CNS and CLS) for each execution. If the new induced tree (New_tree) has a success rate better than the one of the worst tree in the ensemble ($Worst\_tree$), the new induced tree is added to the ensemble. When the size of ensemble is greater than a pre-configured maximum ($max\_number\_of\_trees$) the decision tree with the worst success rate is removed from the ensemble. E-CIDIM tries iteratively to incorporate new decision trees to the ensemble. When it fails consecutively a pre-configured number of times ($number\_of\_failed\_attempts$), E-CIDIM stops. A description of E-CIDIM can be seen in the Figure 2.

When the ensemble has been induced, we can use it to predict. We have defined three kinds of voting: uniform voting, weighted by tree voting and weighted by rule voting. Uniform voting is the simplest kind of voting: every tree gives its prediction vector and every one has the same weight. If we use a weighted voting method, we must define the weights. For weighted by tree voting, every decision tree gives its prediction vector and it is weighted using the success rate of the respective decision tree, then they are combined in a new prediction vector. For weighted by rule voting, every decision tree gives its prediction vector and it is weighted using the weight of the rule in the respective tree. The class predicted by the multiple classifier system is the majority class in the prediction vector.

Now we have described E-CIDIM, we describe how the selected voting method and the parameters influence the performance of E-CIDIM.

In Table 1 we can see that uniform voting and weighted by rule voting obtain the best results. Although there are differences between them, these differences

Table 1. Comparison between voting methods. Configuration: $max\_number\_of\_trees = 10$ and $number\_of\_failed\_attempts = 10$

Dataset	Voting method	Accuracy
Balance	Uniform	77.57 ± 0.65
	Weighted by tree	77.55 ± 0.62
	Weighted by rule	**77.77** ± 0.45
Ecoli	Uniform	80.60 ± 0.56
	Weighted by tree	80.66 ± 0.54
	Weighted by rule	**80.93** ± 0.46
Ionosphere	Uniform	**90.43** ± 0.01
	Weighted by tree	90.25 ± 0.01
	Weighted by rule	90.39 ± 0.01
Pima	Uniform	73.51 ± 0.63
	Weighted by tree	73.58 ± 0.65
	Weighted by rule	**73.98** ± 0.73
Wdbc	Uniform	**95.20** ± 1.21
	Weighted by tree	95.15 ± 1.20
	Weighted by rule	95.15 ± 1.58

are insignificant. Thus, we have set uniform voting as the default value for the voting method because of its simplicity.

In Table 2 we can see that accuracy is generally better when E-CIDIM keeps more trees in the ensemble, although there are some cases in which accuracy is not the best when E-CIDIM uses 20 trees. If we watch the average number of

Table 2. Number of leaves, accuracy and execution time depending on the maximum number of decision trees in the ensemble. Configuration: uniform voting and *number_of_failed_attempts* = 10

Dataset	*max_number_of_trees*	**Leaves**	**Accuracy**	**Time (ms.)**
Balance	5	70.14 ± 0.79	75.97 ± 2.06	**425.54** ± 35.98
	10	69.03 ± 0.99	77.57 ± 1.21	673.77 ± 77.76
	20	**67.74** ± 1.16	**78.83** ± 1.41	1034.29 ±108.54
Ecoli	5	36.72 ± 0.87	80.27 ± 0.92	**913.33** ± 115.80
	10	36.20 ± 0.72	**80.60** ± 0.63	1392.29 ± 99.72
	20	**35.83** ± 0.43	80.54 ± 0.86	2295.01 ± 139.75
Ionosphere	5	20.36 ± 0.33	90.05 ± 0.88	**1745.71** ± 127.07
	10	20.26 ± 0.40	90.43 ± 0.61	2709.53 ± 248.81
	20	**19.89** ± 0.22	**90.60** ± 0.68	4266.83 ± 285.88
Pima	5	41.23 ± 2.06	73.29 ± 0.69	**608.81** ± 116.14
	10	40.02 ± 1.08	73.51 ± 0.56	960.54 ± 82.91
	20	**38.38** ± 1.30	**73.86** ± 0.43	1526.34 ± 124.11
Wdbc	5	20.62 ± 0.46	95.12 ± 0.86	**1588.23** ± 157.61
	10	20.07 ± 0.35	**95.20** ± 0.65	2345.18 ± 194.93
	20	**19.85** ± 0.15	95.12 ± 0.51	3876.90 ± 210.14

Table 3. Number of leaves, accuracy and execution time depending on the number of failed attempts to induce a better decision tree. Configuration: uniform voting and *max_number_of_trees* = 20

Dataset	*number_of_failed_attempts*	**Leaves**	**Accuracy**	**Time (ms.)**
Balance	5	**64.47** ± 0.76	**78.88** ± 1.30	**619.14** ± 48.44
	10	67.74 ± 1.16	78.83 ± 1.41	1034.29 ± 108.54
	20	70.83 ± 0.86	78.02 ± 1.22	1891.98 ± 115.39
Ecoli	5	**34.66** ± 0.36	**81.08** ± 0.66	**1359.22** ± 114.26
	10	35.83 ± 0.43	80.54 ± 0.86	2295.01 ± 139.75
	20	37.11 ± 0.49	80.39 ± 0.76	4121.79 ± 229.32
Ionosphere	5	**19.37** ± 0.28	90.40 ± 0.81	**2539.77** ± 216.45
	10	19.89 ± 0.22	**90.60** ± 0.68	4266.83 ± 285.88
	20	20.53 ± 0.24	90.34 ± 0.57	7558.41 ± 494.31
Pima	5	**34.58** ± 0.46	73.55 ± 0.52	**912.30** ± 45.60
	10	38.38 ± 1.30	**73.86** ± 0.43	1526.34 ± 142.11
	20	42.64 ± 1.09	73.68 ± 0.46	2795.18 ± 300.12
Wdbc	5	**19.15** ± 0.28	**95.29** ± 0.55	**2179.77** ± 133.89
	10	19.85 ± 0.15	95.12 ± 0.51	3876.90 ± 210.14
	20	20.52 ± 0.23	94.87 ± 0.81	6847.52 ± 309.11

leaves in the ensemble, we can note that it is smaller when E-CIDIM keeps more trees in the ensemble. Taking these questions into account, we will set the default value of *max_number_of_trees* to 20. There is a disadvantage when E-CIDIM keeps more trees in the ensemble: it takes more time to finish the induction, but in this case, we consider the performance of E-CIDIM to be positive.

We have set voting method and *max_number_of_trees* to their default values and now we study the influence of *number_of_failed_attempts* to the performance of E-CIDIM. As we can see in Table 3, the smallest trees and the most accurate trees are induced when *number_of_failed_attempts* is lower. In addition, the fastest executions are the ones with lower *number_of_failed_attempts* too. Thus, we will set the default value of *number_of_failed_attempts* to 5.

4 Experimental Results

The experiments we have done and the results we have obtained are now exposed. Before we go on to deal with the particular experiments, we must explain some questions:

- The five datasets we have used (balance-scale, ecoli, ionosphere, pima-indians-diabetes and breast-cancer-wisconsin) have been taken from the [illegible] [19] and are available online. All the used datasets have a common feature: attributes are continuous and they have been discretized. We have done it this way because CIDIM is designed for dealing with nominal variables (ordered or unordered).
- ML-CIDIM has been compared with other well-known methods: bagging [1] and boosting [2]. For the experiments, we have used the implementation of bagging and boosting given in Weka [20]. These two algorithms have been executed using J48 (implementation of C4.5) as their basic classifier. We have configured E-CIDIM with its default configuration (uniform voting, *max_number_of_trees* = 20 and *number_of_failed_attempts* = 5). Considering this, bagging and boosting have been configured to do 20 iterations.
- For each experiment, the presented values for accuracy and average size of trees have been obtained from a 10 x 10 fold cross-validation. Average and standard deviation values are given. To compare results, a statistical test must be made [21] and a t-test has been conducted using the results of the cited 10 x 10 fold cross-validation. The t-test values have been calculated using the statistical package R [22]. A difference is considered as significant if the significance level of the t-test is better than 0.05. We have selected the results obtained by E-CIDIM with default configuration as the reference values. Thus, $\oplus$ indicates that the value is significantly better than the one of E-CIDIM. $\ominus$ signifies that the value is significantly worse than the one of E-CIDIM. In addition to this comparisons, the best result for each experiment has been emphasized using numbers in boldface.

Once we have established the datasets and the configuration used for each algorithm we can continue talking about the experiments.

Having obtained the results shown in Table 4, we can reach some conclusions:

- The average size of the decision trees induced by E-CIDIM is significantly smaller than those induced by bagging or boosting for almost every experiments we have done (there is an exception in balance dataset). Here we can see one of the advantages of using CIDIM as the basic classifier: this algorithm induces small decision trees.

Table 4. Comparison between bagging, boosting and E-CIDIM. Average values and standard deviations are given. Significance tests are with respect to E-CIDIM

Dataset	Algorithm	Leaves	Accuracy
Balance	Bagging-20	**53.66** ± 0.72 ⊕	74.29 ± 1.39 ⊖
	Boosting-20	99.55 ± 1.71 ⊖	72.71 ± 1.11 ⊖
	E-CIDIM-20	64.47 ± 0.76	**78.88** ± 1.30
Ecoli	Bagging-20	65.85 ± 0.80 ⊖	78.43 ± 0.94 ⊖
	Boosting-20	89.73 ± 2.18 ⊖	75.75 ± 1.45 ⊖
	E-CIDIM-20	**34.66** ± 0.36	**81.08** ± 0.66
Ionosphere	Bagging-20	40.20 ± 0.73 ⊖	88.88 ± 0.55 ⊖
	Boosting-20	51.39 ± 0.46 ⊖	**92.33** ± 0.90 ⊕
	E-CIDIM-20	**19.37** ± 0.28	90.40 ± 0.81
Pima	Bagging-20	96.88 ± 1.28 ⊖	72.99 ± 0.68
	Boosting-20	135.93 ± 2.72 ⊖	70.71 ± 1.20 ⊖
	E-CIDIM-20	**34.58** ± 0.46	**73.55** ± 0.52
Wdbc	Bagging-20	56.51 ± 1.03 ⊖	93.95 ± 0.69 ⊖
	Boosting-20	82.35 ± 0.88 ⊖	**95.38** ± 0.37
	E-CIDIM-20	**19.15** ± 0.28	95.29 ± 0.55

Table 5. Comparison between CIDIM, E-CIDIM with *max_number_of_trees* = 10 and E-CIDIM with default configuration. Average values and standard deviations are given. Significance tests are with respect to E-CIDIM with default configuration

Dataset	Algorithm	Accuracy
Balance	CIDIM	68.48 ± 1.18 ⊖
	E-CIDIM-10	77.36 ± 1.33 ⊖
	E-CIDIM-20	**78.88** ± 1.30
Ecoli	CIDIM	77.29 ± 0.80 ⊖
	E-CIDIM-10	80.49 ± 0.89
	E-CIDIM-20	**81.08** ± 0.66
Ionosphere	CIDIM	88.71 ± 1.64 ⊖
	E-CIDIM-10	90.39 ± 0.85
	E-CIDIM-20	**90.40** ± 0.81
Pima	CIDIM	73.29 ± 0.74
	E-CIDIM-10	**73.63** ± 0.59
	E-CIDIM-20	73.55 ± 0.52
Wdbc	CIDIM	92.48 ± 1.05 ⊖
	E-CIDIM-10	95.15 ± 0.48
	E-CIDIM-20	**95.29** ± 0.55

– The accuracy reached by E-CIDIM is significantly better than the accuracy reached by bagging or boosting in many cases. To induce small decision trees without losing too much accuracy has a foundation in the way in which E-CIDIM makes good use of CIDIM's advantages. It improves the isolated performance of an unique CIDIM combining them in an ensemble. This can be seen in Table 5.

5 Conclusions

This paper introduces E-CIDIM, a multiple classifier system that uses CIDIM as its basic classifier. CIDIM is an algorithm that induces small and accurate decision trees and E-CIDIM takes advantage of their characteristics. Thus, E-CIDIM improves the isolated performance of an unique CIDIM combining them in an ensemble.

We have compared results obtained with E-CIDIM, bagging and boosting over different datasets and we can note some questions. E-CIDIM induces ensembles of trees whose sizes are very small and the accuracy that it is achieved by E-CIDIM is usually better than the accuracies achieved by bagging and boosting. Thus, we can conclude that E-CIDIM has a reasonably good performance. However, there is a disadvantage: E-CIDIM takes more time that bagging or boosting to induce the ensemble.

Out aim of improving M-CIDIM involves two issues:

– we are working to improve the CIDIM algorithm providing it the ability of working with continuous attributes. In this way, we will not have to discretize real variables to nominal ordered variables and an automatic execution of CIDIM (or E-CIDIM) will be made.
– we are also working to automatize the selection of the parameters. Thus, no previous configuration would be needed.

References

1. Breiman, L.: Bagging predictors. Machine Learning **24** (1996) 123–140
2. Freund, Y., Schapire, R.E.: Experiments with a new boosting algorithm. In: Proc. of 13th International Conference on Machine Learning. (1996) 146– 148
3. Wolpert, D.: Stacked generalization. Neural Networks **5** (1992) 241–260
4. Gama, J., Brazdil, P.: Cascade generalization. Machine Learning **41** (2000) 315–343
5. Ferri, C., Flach, P., Hernández-Orallo, J.: Delegating classifiers. In: Proceedings of the 21st International Conference on Machine Learning, Omnipress (2004)
6. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth (1984)
7. Quinlan, J.R.: Induction of decision trees. Machine Learning **1** (1986) 81– 106
8. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
9. Utgoff, P.E., Berkman, N.C., Clouse, J.A.: Decision tree induction based on efficient tree restructuring. Machine Learning **29** (1997) 5– 44

10. Ramos-Jiménez, G., Morales-Bueno, R., Villalba-Soria, A.: CIDIM. Control of induction by sample division methods. In: Proceedings of the International Conference on Artificial Intelligence (IC-AI'00), Las Vegas (2000) 1083–1087
11. Ruiz-Gómez, J., Ramos-Jiménez, G., Villalba-Soria, A.: Modelling based on rule induction learning. In: Computers and Computacional Engineering in Control. World Scientific and Engineering Society Press, Greece (1999) 158–163
12. Jerez-Aragonés, J.M., Gómez-Ruiz, J.A., Ramos-Jiménez, G., Muñoz-Pérez, J., Alba-Conejo, E.: A combined neural network and decision trees model for prognosis of breast cancer relapse. Artificial Intelligence in Medicine **27** (2003) 45–63
13. Schapire, R.E.: The strength of weak learnability. Machine Learning **5** (1990) 197–227
14. Freund, Y.: Boosting a weak learning algorithm by majority. Information and Computation **121** (1995) 256–285
15. Freund, Y., Schapire, R.E.: The strength of weak learnability. Journal of Computer and System Sciences **55** (1997) 119–139
16. Aslam, J.A., Decatur, S.E.: General bounds on statistical query learning and PAC learning with noise via hypothesis boosting. Information and Computation **141** (1998) 85–118
17. Bauer, E., Kohavi, R.: An empirical comparison of voting classification algorithms: Bagging, boosting and variants. Machine Learning **36** (1999) 105–139
18. Kearns, M.J., Vazirani, U.V.: On the boosting ability of top-down decision tree learning algorithms. Journal of Computer and System Sciences **58** (1999) 109–128
19. Blake, C., Merz, C.J.: UCI repository of machine learning databases. University of California, Department of Information and Computer Science (2000)
20. Witten, I.H., Frank, E.: Data Mining: Practical machine learning tools with Java implementations. Morgan Kaufmann, San Francisco (2000)
21. Herrera, F., Hervás, C., Otero, J., Sánchez, L.: Un estudio empírico preliminar sobre los tests estadísticos más habituales en el aprendizaje automático. In: Tendencias de la Minería de Datos en España. Red Española Minería Datos (2004)
22. R Development Core Team: R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. (2004) 3-900051-07-0. http://www.R-project.org.

Partially Supervised Classification – Based on Weighted Unlabeled Samples Support Vector Machine

Zhigang Liu[1, 2, 4], Wenzhong Shi[3], Deren Li[4], and Qianqing Qin[4]

[1] State Key Laboratory of Remote Sensing Science,
Jointly Sponsored by Beijing Normal University and
Institute of Remote Sensing Applications, CAS
[2] Research Center for Remote Sensing and GIS, School of Geography,
Beijing Key Laboratory for Remote Sensing of Environment
and Digital Cities, Beijing Normal University, Beijing, 100875, China
zhigangliu@bnu.edu.cn
[3] Advanced Research Centre for Spatial Information Technology,
Department of Land Surveying and Geo-Informatics,
The Hong Kong Polytechnic University, Hong Kong
lswzshi@polyu.edu.hk
[4] State Key Laboratory of Information Engineering in Surveying,
Mapping and Remote Sensing, Wuhan University

Abstract. This paper addresses a new classification technique: partially supervised classification (PSC), which is used to identify a specific land-cover class of interest from a remotely sensed image by using unique training samples belong to a specifically selected class. This paper also presents and discusses a novel Support Vector Machine (SVM) algorithm for PSC. Its training set includes labeled samples belong to the class of interest and unlabeled samples of all classes randomly selected from a remotely sensed image. Moreover, all unlabeled samples are assumed to be training samples of other classes and each of them is assigned a weighting factor indicating the likelihood of this assumption; hence, the algorithm is so-called 'Weighted Unlabeled Sample SVM' (WUS-SVM). Experimental results with both simulated and real data sets indicate that the proposed PSC method is more robust than 1-SVM and has comparable accuracy to a standard SVM.

1 Introduction

It is an important issue in spatial data mining to discover spatial distribution of objects based on remote sensing imagery. Classification of remote sensing imagery is one of the methods used to identify the spatial distribution of land-cover classes. There are generally two traditional classification techniques: non-supervised and supervised. Although training samples are unnecessary for non-supervised classification, the classes derived by this kind of techniques are unpredictable. In supervised classification, each class is defined by training samples which are selected by users. However sufficient and exhaustive training sample set is required for supervised classification. In many cases, a classifier that can recognize only a specific land-cover class of interest in a remotely sensed image is sufficient. Since it is expensive and time-consuming to obtain training

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 118–129, 2005.

samples, it would be very useful if such type of classifier is designed to only provide those training samples belonging to the class of interest (C_{int}), and this technique has been termed "Partially Supervised Classification (PSC)" [1].

There are two main classification schemes: relative and absolute [2]. The former scheme, for example Maximum-likelihood Classifier [3] and Support Vector Machine (SVM) [4], usually provides acceptable classification accuracy but is not suitable for PSC because it requires all the different classes' training samples being present in the corresponding remotely sensed image under analyzing process. The latter one, for example Parallelepiped classifier [3] and 1-SVM [5, 6], allows the classification task to be performed solely using training samples belonging to C_{int}; thus it is suitable for PSC. In spite of this advantage, the accuracy of this type of approach is always limited or heavily dependent on the selection of certain thresholds or parameters.

Many techniques have been developed to achieve a satisfied accuracy of PSC. Most of them try to acquire information of the classes other than C_{int} using unlabeled samples of both C_{int} and the other classes (C_{others}), and deal with PSC by a relative classifier. Liu *et al.* developed a technique called S-EM [8] based on the EM algorithm [8]. However, S-EM is not accurate because of its weak classifier. Jeon and Landgrebe proposed two methods based on probability density estimation [1, 2]. Fernádez -Prieto proposed an improved method [9], in which Radial Basis Function network and Markov Random Fields are used. Although they can improve the accuracy of PSC, all the methods are assumed that the probability density of C_{int} is known or can be estimated correctly by training samples. However, in many cases, the probability density of C_{int} is unknown and it is difficult to be estimated from training samples, especially when only limited training samples are available.

Instead of taking density estimation as an intermediate step, SVM, which forms a decision function directly, can achieve a satisfactory accuracy even with a small training sample set [4]. Because of this important fact, several SVM-based algorithms have been proposed, such as PEBL [10], Roc-SVM [11] and biased SVM [12], with a common feature of transforming PSC into a binary classification to distinguish C_{int} from C_{others}. Both PEBL and Roc-SVM are based on a two-step strategy. In the first step, a set of reliable samples belonged to C_{others} are identified, whereas in the second step, a SVM classifier is trained with training samples of both classes, C_{int} and C_{others}. These two steps together can be seen as an iterative method of increasing the number of unlabeled samples that are classified as C_{others} while maintaining the correctly classified samples of C_{int}. Unlike these two-step algorithms, biased SVM is to assume that all unlabeled samples belong to C_{others} and to try to minimize the number of those unlabeled samples classified as C_{int} whilst to constrain labeled samples of C_{int} to be correctly classified. It shows that biased SVM is superior to PEBL and Roc-SVM [12].

In this paper, a novel SVM algorithm, according to the same assumption of biased SVM that all unlabeled samples belong to C_{others}, is presented, with the special feature that each of the unlabeled samples is assigned a weighting factor indicating the likelihood of this assumption. Therefore, the novel SVM algorithm has been named "Weighted Unlabeled Samples SVM (WUS-SVM)". Besides the above-mentioned difference, penalty parameters of WUS-SVM are automatically determined by cross-validation method rather than F score taken by biased SVM [12]. In order to improve the classification speed, a PSC scheme based on WUS-SVM has also been proposed. Experimental results with both simulated and remotely sensed data verify the effectiveness of the proposed method.

2 The Proposed Technique

Given a set of input vectors (**X**)

$$\mathbf{x}_1,\hbar\ ,\mathbf{x}_m,\mathbf{x}_{m+1},\hbar\ ,\mathbf{x}_{m+n},\mathbf{x}_{m+n+1},\hbar\ ,\mathbf{x}_{m+n+t}\in R^B\,. \qquad (1)$$

where m, n, $B\in N$. Without the loss of generality, we suppose that the first m vectors, $X_L\equiv\{x_1,\ldots, x_m\}$, are the training samples labeled as $\mathrm{C}_{\mathrm{int}}$ and $\mathrm{X}_{\mathrm{U}}\equiv\{\mathrm{x}_{\mathrm{m+1}},\ldots, \mathrm{x}_{\mathrm{m+n}}\}$ are unlabeled samples randomly selected from the test set $\mathrm{X}_{\mathrm{T}}\equiv\{\mathrm{x}_{\mathrm{m+1}},\ldots, \mathrm{x}_{\mathrm{m+n+t}}\}$. The goal of PSC is to distinguish samples that belong to $\mathrm{C}_{\mathrm{int}}$ from X_{T}.

2.1 Optimal Separating Hyperplane

Firstly, the case where $\mathrm{C}_{\mathrm{int}}$ and $\mathrm{C}_{\mathrm{others}}$ are linearly separable in the input space, is considered. In other words, there is a hyperplane that separates the samples of two classes: $\mathrm{C}_{\mathrm{int}}$ and $\mathrm{C}_{\mathrm{others}}$. Because none of the training samples belong to $\mathrm{C}_{\mathrm{others}}$ is available, it is assumed that all unlabeled samples belong to $\mathrm{C}_{\mathrm{others}}$ and each of them is assigned a weight factor indicating the likelihood of this assumption. In this paper, the weight (S_i) of an unlabeled sample is defined as its minimum distance to X_{L}:

$$S_i=\min_j D(\mathbf{x}_{m+i},\mathbf{x}_j),\ \ j=1,\hbar\ ,m;\ \ i=1,\hbar\ ,n\,. \qquad (2)$$

Then, similar to SVM, a hyperplane with the margin width of $2/\|\mathbf{w}\|$ is constructed to separate as many samples as possible in X_{U} from X_{L} (Fig. 1). Then training errors of the two classes are defined as:

$$E_L\equiv\sum_{i=1}^{m}\xi_i\equiv\sum_{i=1}^{m}\left|1-(\ <\mathbf{w},\mathbf{x}_i>+b)\right|_+\,. \qquad (3)$$

$$E_U\equiv\sum_{i=m+1}^{m+n}S_i\,\xi_i\equiv\sum_{i=m+1}^{m+n}S_i\left|(<\mathbf{w},\mathbf{x}_i>+b)+1\right|_+\,. \qquad (4)$$

where $|\,\mathrm{x}\,|_+=\mathrm{x}$, if x is positive and zero otherwise.

When samples of $\mathrm{C}_{\mathrm{int}}$ are separable from those of $\mathrm{C}_{\mathrm{others}}$, there is a gap between them in the input space. Once the separating plane locates at the gap, E_{U} will be relatively small, whereas the separating margin will be relatively large. Therefore, the strategy of keeping the E_{U} as small as possible and simultaneously the margin of the separating plane (Fig. 1) as large as possible is applied to locate the separating plane properly. Obviously, the maintenances of a small E_{U} and of a large margin are two conflicting goals. A penalty parameter is introduced to determine the trade-off between them.

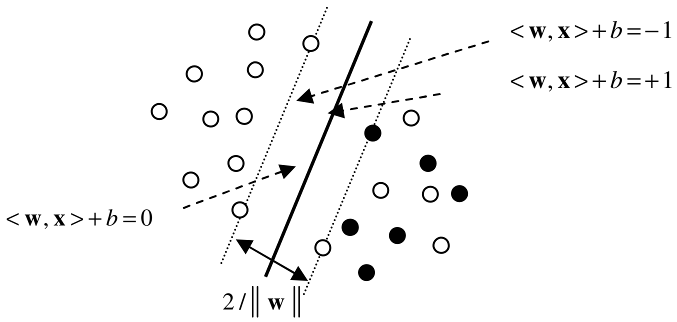

$<\mathbf{w},\mathbf{x}>+b=-1$
$<\mathbf{w},\mathbf{x}>+b=+1$
$<\mathbf{w},\mathbf{x}>+b=0$
$2/\|\mathbf{w}\|$

Fig. 1. Separating hyperplane in 2-D space (Black dots are training examples of C_{int} and white dots are training examples of C_{others}.)

Considering the three requirements, minimizing E_L, minimizing E_U and maximizing a margin's width, the optimal separating hyperplane can be constructed by solving the following optimization problem:

$$\operatorname*{Min}_{\mathbf{w}\in H,\xi\in R^{m+n}} \quad \tau(\mathbf{w},\xi)=\frac{1}{2}\|\mathbf{w}\|^2+P_{\mathrm{int}}\sum_{i=1}^{m}\xi_i \quad +P_{others}\sum_{i=m+1}^{m+n}S_i\xi_i \tag{5}$$

$$\text{s. t. } \forall_{i=1}^{m} :<\mathbf{w},\mathbf{x}_i>+b\geq 1-\xi_i$$

$$\forall_{i=m+1}^{m+n} :<\mathbf{w},\mathbf{x}_i>+b\leq -1+\xi_i$$

$$\forall_{i=1}^{m+n} : \xi_i\geq 0$$

where the penalty parameters, P_{int} and P_{others}, are positive constants to determine the trade-off between the three conflicting goals. Because (5) is a generalization of the optimization problem of SVM, we can solve (5) by a similar method.

Note that if all weight factors S_i in (5) are set to be 1, then the optimization problem is equal to that of biased SVM [12]. In such case, however, some errors may occur. Figure 2 gives an example of two hyperplanes, H_1 and H_2, with a same margin's width. Because both of them classify X_L correctly and thus E_L equals zero, the one with smaller E_U would excel the other. Therefore, H_2 would be selected. But in this example, the selection is seemed unreasonable because H_2 classifies an unlabeled sample as C_{int} far away from X_L. However, the weight factors (2) are introduced in (5), and this kind of error can be avoided.

In the above discussion, we assume that samples of C_{others} and C_{int} are linearly separable in the input space. However, in PSC, samples of C_{int} are always surrounded by those of C_{others} and thus it is difficult to separate them with a hyperplane. To generalize the above method to nonlinear cases, we can use 'kernel trick' [4]. This is done by substituting dot product/s with a kernel function. In this way, a hyperplane constructed in a high dimensional feature space can be a complex separating boundary in the input space. The classifier obtained with the above algorithm is named as *WUS-SVM.*

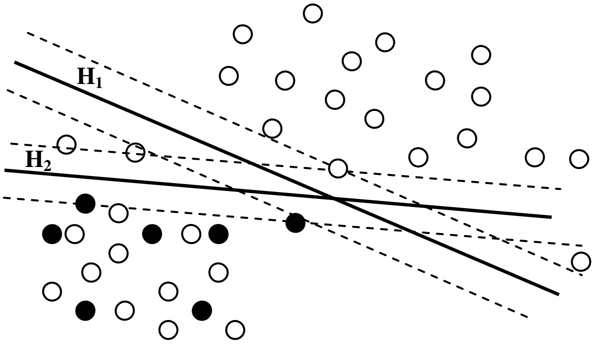


Fig. 2. Comparison of hyperplanes (Black dots are training examples of C_{int}; and white dots are training examples of C_{others}.)

2.2 Selection of Penalty Parameters

As we can always assume that most training samples X_L are labeled correctly in PSC, E_L should be as small as possible. Therefore, P_{int} is set as a large value so that a high penalty will be caused by any E_L and most of the samples in X_L will be classified correctly. In this case, the optimal hyperplane is determined by the left two terms in the right side of (5): margin width and products of E_U and P_{others}. If P_{others} is too large, the effect of E_U will dominate that of margin width. Therefore, a hyperplane closed to X_L will be constructed. However, if P_{others} is too small, the effect of E_U will be overlooked and the optimal hyperplane will probably be too far away from X_L. So it is important to set P_{others} with a reasonable value so as to achieve a satisfied classification accuracy.

In this paper, P_{others} is determined by iterative cross-validations. Firstly, X_L is randomly divided into *k* parts. For a value of P_{others}, the WUS-SVM is trained on the union of nine parts and the resulting decision function is tested on the remaining part. This is done for every part and then the average classification error is computed. If the average accuracy is greater than a predefined criterion, the test value of P_{others} will be regarded as reasonable. We start the process with a large enough value and then decrease the value gradually until finding the first reasonable value for P_{others}. Intuitively speaking, in this process, the separating boundary is too tight at first and then relaxes gradually until locating between the samples of C_{int} and C_{others}.

2.3 Partially Supervised Classification

The classification speed of all SVM-like algorithms is determined by the number of the support vectors — the samples that locate on the margin hyperplanes or within an error side of the margin hyperplanes. In a training of WUS-SVM, when a correct hyperplane is constructed, the unlabeled samples belong to C_{int} become support vectors. If there are many unlabeled samples of C_{int}, WUS-SVM will have a large number of support vectors and its classification speed will be low.

Note that once WUS-SVM is obtained, it can be used to classify the unlabeled samples. Then labeled training samples of both classes are available, to be used to train the standard SVM. The final classifier is named as Retain SVM (RT-SVM), which will have much fewer support vectors than those of WUS-SVM. Therefore, we have proposed a PSC scheme that consists of the following steps:

1) train WUS-SVM using weighted unlabeled samples and training samples of C_{int};
2) classify unlabeled samples with WUS-SVM;
3) train RT-SVM using labeled training samples belonged to both C_{int} and C_{others};
4) classify other samples with RT-SVM.

3 Experiments and Discussion

To test the performance of the proposed method, experiments were carried out with both simulated data and a remotely sensed image. For the comparison purpose, 1-SVM and SVM were also used in these experiments. The SVM was trained with training samples of all classes by the standard algorithm, C-SVM [4] with C=100. The 1-SVM was trained with only training samples belonged to C_{int} by the standard algorithm in LibSVM [13]. Because the parameter, υ, in 1-SVM was the upper bound on the fraction of outliers in training samples of C_{int} [5] and most of the training samples of C_{int} were labeled correctly in our experiments, υ was set as 0.01. In the training of WUS-SVM, P_{int} was set as the value of 100 and the value of P_{others} was automatically selected by ten-fold cross-validation. In the training of RT-SVM, the standard algorithm, C-SVM (C=100), was also used. In all classifiers, the only consideration was Gaussian RBF kernel function: $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{-\gamma\left\|\mathbf{x}_i - \mathbf{x}_j\right\|^2\right\}$. Three different values of γ (1, 10 and 100) were used to observe the sensitivities of different classifiers to γ. Classification accuracies were evaluated by the following two criteria [1]:

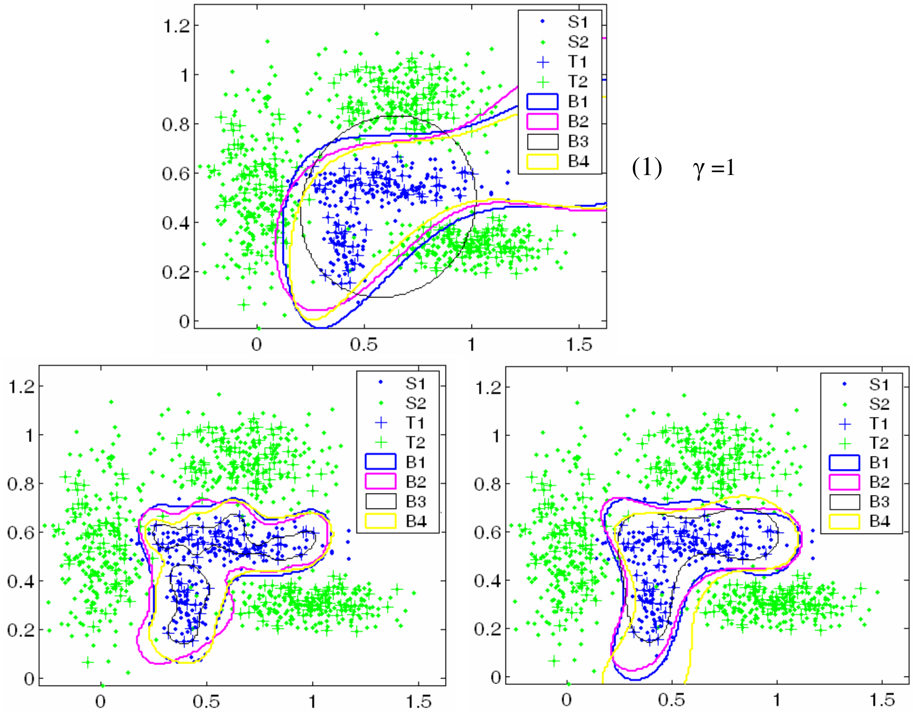
Omission error (OE): $P\{\mathbf{x}$ is decided as $C_{others} \mid \mathbf{x} \in C_{int}\}$.
Commission error (CE): $P\{\mathbf{x}$ is decided as $C_{int} \mid \mathbf{x} \in C_{others}\}$.

3.1 Experiments with Simulated Data

In these experiments, the distributions of C_{int} and C_{others} were respectively composed of two and three bivariate normal distributions. According to the distributions, 250 samples of C_{int} and 750 samples of C_{others} were generated (Fig. 3). 20% of them were selected as training samples for both classes, while 20% of all samples were randomly selected as unlabeled training samples.

The classification boundaries of all classifiers are shown in Fig. 3. They all became gradually tighter as γ increased. The boundaries of SVM, WUS-SVM and RT-SVM were closed to each other with the overall classification accuracies greater than 85%. However the performances of 1-SVM varied widely. When γ = 1, its commission error was about 20%. As γ increased, although its commission error decreased, its omission error increased rapidly. When γ = 100, the omission error was about 35%.

In 1-SVM, the numbers of support vectors were always the smallest, whereas in WUS-SVM the maximum support vectors were always true. However, after retraining, the support vector numbers of RT-SVM decreased obviously and approximated to that of SVM (Tab. 1).



(1) γ =1

(2) γ = 10

(3) γ = 100

Fig. 3. The classification results of 2-demension simulated data (S1: samples of C_{int}; S2: samples of C_{others}; T1: training samples of C_{int}; T2 training samples of C_{others}; B1: Decision boundary of WUS--SVM; B2: Decision boundary of RT-SVM; B3: Decision boundary of 1-SVM; B4: Decision boundary of SVM.)

Table 1. The Numbers of Support Vectors of the Classifiers in the simulated data experiments

Classifiers	γ = 1	γ = 10	γ = 100
SVM	30	15	86
1-SVM	3	7	23
WUS-SVM	101	137	160
RT-SVM	31	19	103

3.2 Experiments with ETM+ Remote Sensing Image

Real data experiments were carried out using a 390×350 pixel subsection of a Landsat ETM+ image acquired over the urban areas around the Victoria Harbor in Hong Kong in September, 2001 (Fig 4). With the reference of the land utilization map of Hong Kong 2001, five different information classes, including high density urban (HDU), low density urban (LDU), Vegetation (V), Water (W) and Open Space (OS), were delimited from this image. With the help of IKONOS image and fieldwork, 790 and 800 samples were selected for training and evaluation of classification accuracy. 1% of the pixels were randomly selected from the whole image as unlabeled training samples. In classification, the feature vector of each pixel was composed of the values of six spectral bands (bands 1-5, and 7).

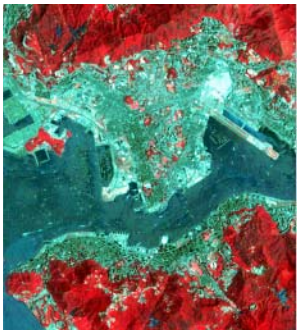


Fig. 4. Original ETM+ Image

In PSC classifications, each class was regarded as C_{int} in turn. However, it is a little complex for the classifications of SVM. Since SVM was originally developed as a binary classifier, various strategies have been proposed to adapt this method to multi-class cases. In our study, the "one–against-one" method was used, which was experimentally proved to be suitable for practice [14]. The method was constructed with a classifier for each pair of classes and the voting strategy approach was used to predict a test sample with the largest vote to a candidate class. The identical votes would be unclassifiable. Such cases occurred sparsely in our experiments. Because of the space limitation and similarity of classification results of SVM with different values of γ, only classification result with γ =10 is given in Fig. 5.

The classification results (Tab. 2) showed that, although the performances of RT-SVM classifiers with different values of γ were a bit less stable than those of SVM classifiers, RT-SVM were obviously outperformed 1-SVM classifiers. Especially in the classifications of high density urban (Fig 6) and vegetation (Fig 7), because training samples of these two classes had large reflective range, the commission errors of 1-SVM with small γ were very high. As γ increased, although commission errors decreased, omission errors increased rapidly. RT-SVM classifiers had the same tendency, but the extent of change was much less than that of 1-SVM classifiers. As to the average value of commission error and omission error, the classification accuracies of RT-SVM were only slightly lower than that of SVM and obviously higher than that of 1-SVM.

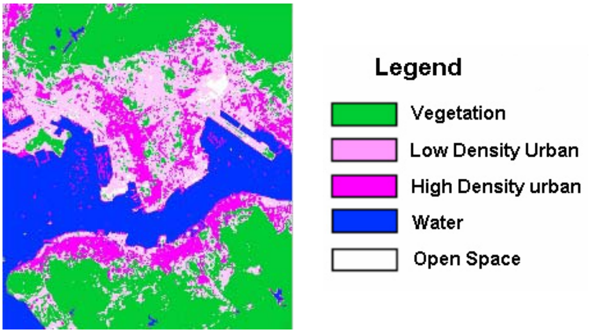
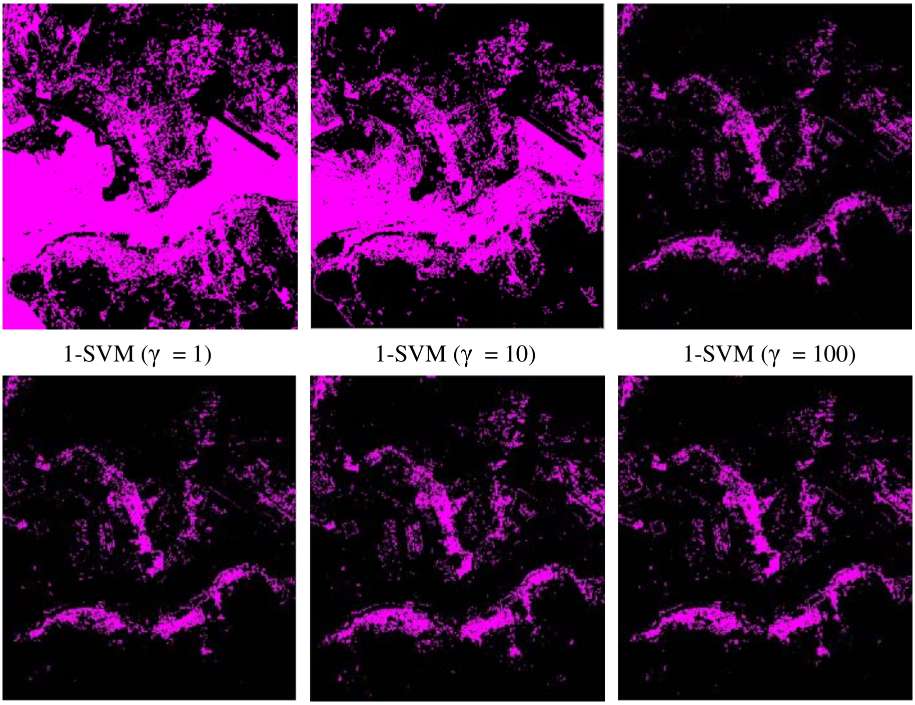


Fig. 5. Classification results of SVM (γ = 10)



1-SVM (γ = 1) 1-SVM (γ = 10) 1-SVM (γ = 100)

RT-SVM (γ = 1) RT-SVM (γ = 10) RT-SVM (γ = 100)

Fig. 6. High density urban classification results of RT-SVM and 1-SVM

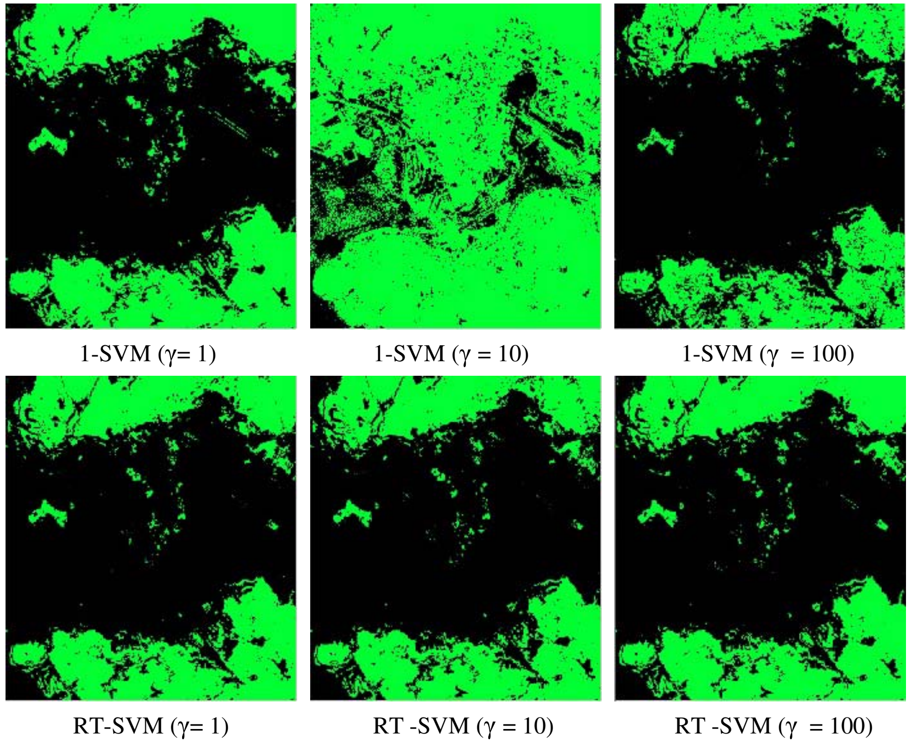

1-SVM (γ= 1)
1-SVM (γ = 10)
1-SVM (γ = 100)
RT-SVM (γ= 1)
RT -SVM (γ = 10)
RT -SVM (γ = 100)

Fig. 7. Vegetation classification results of RT-SVM

Table 2. Omission Error and Commission Error of Different Classifiers

Classifiers	γ	HDU		LDU		V		W		OS	
		OE	CE	OE	CE	OE	CE	OE	CE	OE	CE
SVM	1	7.65	7.28	5.32	5.32	1.07	1.34	5.27	0.92	2.85	0.58
1-SVM		0.13	58.64	3.83	6.81	0	87.43	2.68	5.10	3.17	2.33
RT-SVM		6.97	6.63	3.60	7.27	1.82	0.89	4.30	1.51	2.52	1.37
SVM	10	4.60	7.61	9.48	4.92	1.33	0.80	5.29	0.52	2.36	1.22
1-SVM		0.58	48.27	9.29	6.27	0.29	1.15	3.70	3.17	3.09	2.07
RT-SVM		7.34	5.86	10.56	5.21	1.38	0.91	4.47	1.36	2.97	0.95
SVM	100	4.75	7.62	9.75	4.58	1.33	0.81	5.64	0.48	2.73	0.91
1-SVM		7.67	3.68	17.43	4.65	7.86	0	5.76	0.52	20.01	0
RT-SVM		7.89	4.93	10.31	4.67	2.54	0.77	4.92	0.99	2.82	0.83

In terms of the classification speed, the 1-SVM classifiers had the fewest support vectors (Tab. 3) and thus were the fastest ones. The support vector numbers of RT-SVM classifiers were largely less than those of the corresponding WUS-SVM classifiers. Therefore, using RT-SVM classifiers could improve the classification speed obviously.

However, it is worth to point out that the whole training speed of the proposed PSC method was slower than that of 1-SVM and SVM for the following reasons: 1) In the training of WUS-SVM, extra time was spent in iterative cross validations to determine the value of P_{others}; 2) More training samples were involved in the training of WUS-SVM than those of 1-SVM and SVM; 3) It was slow to classify unlabeled samples with WUS-SVM classifier, which always have many support vectors (Tab. 3).

Table 3. The Support Vector Numbers of the Classifiers in Experiments of Remote Sensing Image

Classifiers	γ	HDU	LDU	V	W	OS
SVM	1	56				
1-SVM		2	4	4	2	3
WUS-SVM		329	462	561	349	35
RT-SVM		139	107	71	30	18
SVM	10	45				
1-SVM		2	7	5	2	5
WUS-SVM		202	373	514	336	28
RT-SVM		59	79	24	19	17
SVM	100	116				
1-SVM		11	27	26	9	15
WUS-SVM		163	334	551	331	26
RT-SVM		59	75	78	18	24

4 Conclusion

In this paper, an algorithm called Weighted Unlabeled Sample Support Vector Machines (WUS-SVM) and a new partially supervised classification method based on WUS-SVM are proposed. Experimental results of both simulated data and ETM+ image have shown that the accuracy of the proposed PSC method is just slightly lower than (sometimes comparable with) that of SVM (a fully supervised classifier) and obviously higher than that of 1-SVM. Meanwhile, because of the variance of Gaussian kernel parameter the performance of the proposed method is more robust than 1-SVM. For the speed, more time should be spent on building the classifier of the proposed PSC method. But in the view of time savings for collection of exhaustive training samples, the proposed method is convenient and effective. In future research, more experiments should be conducted to compare the proposed method with other existing PSC methods.

Acknowledgments

This research was partially supported by a grant from the Chinese Academy of Sciences (KZCX3-SW-338-1), Major State Basic Research Project of China (G20000779) and RGC CERG project (PolyU5167/03E and PolyU 5153/04E).

References

1. Jeon B., Landgrebe D. A.: Partially Supervised Classification Using Weighted Unsupervised Clustering. IEEE Transactions on Geoscience and Remote Sensing, 37(2) (1999) 1073–1079
2. Jeon B., Landgrebe D. A.: A New Supervised Absolute Classifier. Proceeding Of IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (1990) 2363–2366
3. Richards, J.: A. Remote sensing Digital Image Analysis. 2nd edn. Springer-Verlag. Berlin Heidelberg New York (1993)
4. Vapnik V.: The Nature of Statistical Learning Theory. Springer, New York (1995)
5. Schölkopf, B., Platt J. C., Shawe-Taylor J., Smola A. J., Williamson R. C.: Estimating the Support of A High-Dimensional Distribution. Neural Computation 13(7) (2001) 1443–1471
6. Tax, D.: One-class classification. PhD thesis, Delft University of Technology. (2001)
7. Bing Liu, Wee Sun Lee, Philip S Yu, Xiaoli Li: Partially Supervised Classification of Text Documents. Proceedings of the Nineteenth International Conference on Machine Learning (ICML-2002) Sydney, Australia (2002)
8. Dempster A., Laird N., Rubin D.: Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. (1977)
9. Fernádez –Prieto D: An Iterative Approach to Partially Supervised Classification Problems. International Journal of Remote Sensing. 23(18) (2002) 3887–3892
10. Yu, H., Han, J., Chang, K. C.-C.:. PEBL:Positive example based learning for web page classification using SVM. Proceeding Of Int. Conf. On Knowledge Discovery in Databases (KDD'02). (2002)
11. Li, Xiaoli, Bing Liu:. Learning to Classify Text Using Positive and Unlabeled Data. Proceedings of Eighteenth International Joint Conference on Artificial Intelligence (IJCAI-03), Acapulco, Mexico. (2003*)*
12. Liu, Bing, Yang Dai, Xiaoli Li, Wee Sun Lee, Philip Yu: Building Text Classifiers Using Positive and Unlabeled Examples. Proceedings of the Third IEEE International Conference on Data Mining (ICDM'03), Melbourne, Florida, (2003)
13. Chang. C.-C., Lin, C.-J.: LIBSVM: a libray for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm/ (2001)
14. Hsu C. W., Lin C. J.:. A comparison of methods for multi-class support vector machines. IEEE Transactions on Neural Networks.13 (2002) 415–425

Mining Correlated Rules for Associative Classification[*]

Jian Chen, Jian Yin, and Jin Huang

Department of Computer Science, Zhongshan University, Guangzhou, China
ellachen@gmail.com

Abstract. Associative classification is a well-known technique which uses association rules to predict the class label for new data object. This model has been recently reported to achieve higher accuracy than traditional classification approaches. There are various strategies for good associative classification in its three main phases: rules generation, rules pruning and classification. Based on a systematic study of these strategies, we propose a new framework named MCRAC, i.e., *Mining Correlated Rules for Associative Classification*. MCRAC integrates the advantages of the previously proposed effective strategies as well as the new strategies presented in this paper. An extensive performance study reveals that the advantages of the strategies and the improvement of MCRAC outperform other associative classification approaches on accuracy.

1 Introduction

Association rules describe the co-occurrence relationships among data item in a large transaction database. They have been extensively studied in the literature for their usefulness in many real world areas such as market baskets analysis, expert system, stocks trend prediction, even public health surveillance, etc. So association rule mining have become one of the most important fields in knowledge discovery. There have been many efficient algorithms and their variants of association rules discovery, such as Apriori or FPgrowth.

Classification is a supervised machine learning method, which aims to build a classifier and make prediction for new data object whose class label is unknown. Classification has multiple applications and has been applied in many fields such as text categorization, speech recognition, drug discovery and development, etc.

In recent years, a new classification technique, called associative classification, is proposed to combine the advantages of association rule mining and classification. In general, this model extracts class association rules from the training

[*] This work is supported by the National Natural Science Foundation of China (60205007), Natural Science Foundation of Guangdong Province (031558, 04300462), Research Foundation of National Science and Technology Plan Project (2004BA721A02), Research Foundation of Science and Technology Plan Project in Guangdong Province (2003C50118) and Research Foundation of Science and Technology Plan Project in Guangzhou City (2002Z3-E0017).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 130–140, 2005.

set and build a classifier based on these rules to make prediction for new data object. The recent studies show that this classification model achieves higher accuracy than traditional classification approaches such as C4.5 [1].

Different algorithms have been developed for associative classification such as CBA [2], CMAR [3], CPAR [4], MMAC [5], etc. Various search strategies have been developed, such as Foil Gain vs. Support, single rule prediction vs. multiple rules prediction, independent rule vs. strong correlated rule, etc. However, two critical things are missing: (1) there is no systematic study on comparing the strategies and evaluate their pros and cons objectively; and (2) there is no thorough discussion on how to integrate the winning strategies and achieve an even better algorithm. With the research proceeded so far, it is the right time to ask "what is a best strategy in every phase in special case?" and "how can we pick and integrate the best strategies to achieve higher performance in general cases?" In this study, we answer the above questions by a systematic study on the search strategies and develop a winning algorithm MCRAC. MCRAC integrates the advantages of the previously proposed effective strategies as well as the new strategies presented in this paper. A thorough performance study has shown the advantages of the strategies and the improvement of MCRAC over existing mining algorithms on accuracy, including CBA, CMAR and CPAR.

The remainder of the paper is organized as follows: Section 2 gives a basic concept and problem statement on associative classification. In Section 3, we present an overview of the principal search strategies developed before and analyze their pros and cons. We introduce our algorithm MCRAC in details in section 4. Experimental results are described in Section 5 along with the performance of our algorithm in compared with other previous classification based on association rules. Finally, we summarize our research work and draw conclusions in Section 6.

2 Basic Concepts and Terminology

2.1 Association Rule

Let $A = \{A_1, A_2, \ldots, A_k\}$ is a set of k attributes. $V[A] = \{v_1, v_2, \ldots, v_j\}$ is the domain of attribute A. Each continuous and nominal attribute in it has been discretized into a categorical attribute. Let $C = \{c_1, c_2, \ldots, c_m\}$ is a set of possible class label for class attributes. Let $T = \{t_1, t_2, \ldots, t_n\}$ is called a dataset, where each t in T follows the scheme $\{v_1, v_2, \ldots, v_k\} (v_i \in V[A], 1 \leq v_i \leq n)$.

Definition 1. ***(Literal)*** [illegible] ***literal*** l [illegible] $A_i = v$, $A_i \in A, v \in V[A]$ [illegible] $t = \{v_1, v_2, \ldots, v_k\}$ [illegible] l [illegible] $v_i = v$, [illegible] v_i [illegible] i^{th} [illegible] t

Definition 2. ***(Association Rule)*** [illegible] ***association rule*** [illegible] $a_r \Rightarrow c_r$ [illegible] a_r [illegible] $l_1 \wedge l_2 \wedge \ldots \wedge l_i$ [illegible] ***antecedent*** [illegible] c_r [illegible] $l'_1 \wedge l'_2 \wedge \ldots \wedge l'_j$ [illegible] ***consequent*** [illegible] $a_r \cap c_r = \emptyset$

[illegible] a_r [illegible] c_r [illegible]

Definition 3. ***(Support and Confidence)*** [illegible] $a_r \Rightarrow c_r$, $sup(r) = |\{t|t \in T, t\ satisfies\ r\}|$ [illegible] ***support*** [illegible]

$$conf(r) = \frac{sup(r)}{|\{t|t \in T, t\ satisfies\ a_r\}|}$$

[illegible] ***confidence*** [illegible]

2.2 Class Association Rule

The main task of classification is to discover a set of rules from the training set with the attributes in the rules' antecedents and the class label in the rules' consequent, and use them to build a classifier that is used later in the classification process. So only association rules relating the rule antecedent to a certain class label are of interest. In the literature on associative classification the term class association rule has been introduced to distinguish such rules from regular association rules that may consist of an arbitrary conjunction of literals.

Definition 4. ***(Class Association Rule)*** ***class association rule*** [illegible] $a_r \Rightarrow c_r$ [illegible] a_r [illegible] $l_1 \wedge l_2 \wedge \ldots \wedge l_i$ [illegible] ***antecedent*** [illegible] c_r [illegible] c_i [illegible] ***consequent*** [illegible] $c_i \in C$, [illegible] C [illegible]

Typical classification based on association rules has three main phases when a training set is given:

1. `Rules generation.` Generating all the class association rules (CARs) satisfying certain user-specified threshold as candidate rules by association rule mining algorithm. These discovered rules have the form of ($attributes \Rightarrow class\_label$).
2. `Rules pruning.` Evaluating the qualities of all CARs discovered in the previous phase and pruning the redundant and low effective rules. The challenging task in this phase is how to select a good criterion to evaluate the qualities of the rules. Just "useful" rules in training set with high qualities are selected to form a classifier. To improve prediction accuracy, most methods have taken different strategies to prune redundant or negative correlated rules.
3. `Classification.` Assigning a class label for a new data object. When a new data object without a class label comes, the classifier ranks the fitness of these rules and select some or all suitable rules to make a prediction.

3 Strategies for Associative Classification

Various strategies for the above three main phases of associative classification are proposed in the previous studies. In this section, we present a systematic overview of these strategies, and analyze their pros and cons.

3.1 Foil Gain vs. Support

In the first phase of associative classification there are several criterions for rule extraction. Most previous works on associative classification like CBA or CMAR usually uses support based threshold and set its value to 1%. This threshold allows rule extraction to be tractable and on the average yields a good accuracy. But low support threshold often causes numerous rules, in part of which are useless and redundant. However, it is also bad if the support is assigned too high. Underfitting will happen and some highly predictive rules with low support but high confidence will probably be missed. It is not easy to decide on a good value of support. And some works [6] already pointed out this support threshold based method may not be enough accurate in some case. In [7] an extensive compact form to encode a complete rules set is developed to resolve this problem. This proposed compact form is based on the concept of essential rule and taking the notion of closed itemset for rules set compression, which is significantly smaller than the complete rules set. This compact form also provides a wider selection of rules obtained by allowing very low support threshold.

A recent novel study CPAR[4] uses Foil Gain [8] to evaluate the information gain of the current rule r. This gain values are calculated by using the total weights in the positive and negative instance sets instead of simply counting up the number of records in the training sets. The positive instances are those not only satisfying a_r but also c_r, while the negative instances just satisfy a_r but does not consist with c_r. Given a literal p and the current rule $r : a_r \Rightarrow c_r$, $p \notin a_r$, the Foil Gain of p is defined as:

$$FoilGain(p) = |P^*|(\log \frac{|P^*|}{|P^*| + |N^*|} - \log \frac{|P|}{|P| + |N|}) \tag{1}$$

where there are $|P|$ positive instances and $|N|$ negative instances satisfying a_r. And after appending p to r, there will be $|P^*|$ positive and $|N^*|$ negative instances satisfying the antecedent $a_{r'} = a_r \wedge p$ of the new rule r'. In fact, $FoilGain(p)$ is the number of bits saved in representing all the positive instances by adding p to r. By using this gain measure, CPAR performs a depth-first-search rule generation process directly from the training set with a much smaller set of high quality and low redundancy CARs, avoiding repeated calculation in rule generation.

3.2 Single Rule Prediction vs. Multiple Rules Prediction

Once the classifier has been established in the form of a list of rules, regardless of the methodology used to generate it, there are a number of strategies for using the resulting classifier to classify unseen data object as follows.

Best rule. Choose the single best rule which matches the data object and has the highest ranks to make a prediction. There are several schemes to score the rules. (1) Combination effect of confidence, support and size of antecedent, with confidence being the most significant factor, like CBA. (2) Interestingness measures which is the weighted relative accuracy reflecting associations among attributes in a rule [9].

All Rules. Collect all rules in the classifier satisfying the given unseen data and make a prediction by the "combined effect" of different class association rules. This rule selection strategy yields three situations: (1) If there are no rule's antecedent satisfying the new object, a default class is predicted; (2) If all the rules matching the new object have the same class label, this label will be assigned to the new object directly; and (3) The most complex situation is when there are multiple rules which are not consistent in class label satisfying the new object. Then all rules will be separated into different group according to their class labels and the "combined effect" of each group will be estimated. The strongest one wins. We will discuss the method adopted in CMAR to measure the "combined effect" in next sub-section.

***k*-Best rules.** Selects k-best rules for each class and evaluates the average accuracy of each class. CPAR use Laplace expected error estimate to evaluate the goodness of a rule. The expected accuracy of a given rule r is defined as:

$$LaplaceAccuracy(r) = \frac{sup(r) + 1}{sup(a_r) + k} \qquad (2)$$

where $sup(r)$ is the total support for r, $sup(a_r)$ is the support for r's antecedent, k is the number of classes. CPAR compares the average expected accuracy of each class and choose the class with the highest expected accuracy as the predicted class.

And the experimental results have shown that the multiple rules selection strategies adopted in CMAR and CPAR achieves higher prediction accuracy than just relying on a single rule to classify data [3].

3.3 Independent Rule vs. Correlated Rule

In most associative classification approaches, all attributes in class association rule are deemed to be independent of each other. But in real world, it is not true. It has been recognized that associations are not appropriate for all situations and many researchers tried to take some strategies to overcome this bias. Brin et. al mentioned for the first time in [10] the notion of correlation and studied the problem of efficiently finding strong correlated rules set of data objects from large databases. They defined the correlation on the χ^2 metric, which is widely used by statisticians for testing independence. The idea is that a set is said to be correlated with probability α provided its χ^2 metric exceeds the expected χ^2 value corresponding to the probability α.

In the rule pruning phase, for each rule r, CMAR uses a χ^2 testing to judge whether r is positively correlated or negative correlated. Only positive correlated rules can be used to build the classifier, and all the other rules are pruned. Furthermore, in classification phase, when there are more than one rules with different class labels satisfying the new data object, the authors observed that simply choosing the rule with highest χ^2 value may be favorable to minority classes. So CMAR adopts a weighted χ^2 value as equation (3) to measure the ” [illegible] ” of each rules group in which rules share the same class class label. For each rule r in the classifier:

$$max\chi^2(r) = (\min\{sup(a_r), sup(c_r)\} - \frac{sup(a_r)sup(c_r)}{|T|})^2|T|e \tag{3}$$

Please take the literature [3] as reference to get more details about the parameter e. Then the weighted χ^2 measure of the group (with n rules) is defined as:

$$weighted\chi^2 = \sum_{i=1}^{n} \frac{\chi^2(r_i)\chi^2(r_i)}{max\chi^2(r_i)} \tag{4}$$

4 Mining Correlated Rules for Associative Classification

Some studies above have showed that there are cases when many uninteresting rules may be produced even when they satisfy user-specified constrains. This is because the class distribution in the dataset is not taken into account. In this paper we consider adding a measure based on correlation analysis but differs from χ^2 value.

4.1 Correlation Coefficient

For a given class association rule $r : a_r \Rightarrow c_r$, $c_r \in C$, where C is the set of class label. $p(a_r)$ is the probability that a_r occurs in the dataset and $p(\bar{a_r}) = 1 - p(a_r)$ the probability that a_r does not occur in the dataset. Likewise, $p(a_r c_r)$ is that the probability that a_r occurs in the instances and these instances belong to class c_r, while $p(\bar{a_r} c_r)$ is that the probability that a_r does not occur in the instances but these instances belong to class c_r.

Table 1 summarizes the information about a_r and c_r variables in a dataset in a 2×2 contingency table. The cells of this table represent the possible combinations of a_r and c_r and give the probability associated with each combination.

Definition 5. ***(Independent Rule and Correlated Rule)*** [illegible] $r : a_r \Rightarrow c_r$ [illegible] ***Independent Rule*** [illegible] $p(a_r c_r) = p(a_r)p(c_r)$ [illegible] r [illegible] ***Correlated Rule***

Table 1. 2x2 contingency table for antecedent a_r and consequent c_r

	c_r	$\bar{c_r}$	$\sum_{row}$
a_r	$p(a_r c_r)$	$p(a_r \bar{c_r})$	$p(a_r)$
$\bar{a_r}$	$p(\bar{a_r} c_r)$	$p(\bar{a_r} \bar{c_r})$	$p(\bar{a_r})$
$\sum_{col}$	$p(c_r)$	$p(\bar{c_r})$	1

The correlation coefficient is a measure of the strength of the linear relationship between a pair of two variables. It is discussed in the context of association patterns in [11]. Given the values in the contingency table for binary variables, We can give the ϕ correlation coefficient as the equation (5):

$$\phi = \frac{p(a_r c_r) - p(a_r)p(c_r)}{\sqrt{p(a_r)p(c_r)p(1 - p(a_r))p(1 - p(c_r))}} \tag{5}$$

The power of correlation coefficient has been discussed in [12]. They presented that a correlation greater than 0.5 is large, 0.5-0.3 is moderate, 0.3-0.1 is small, and anything smaller than 0.1 is insubstantial, trivial, or otherwise not worth worrying.

The correlated rules used in CMAR are those referred to positive association rules like $(a_r \Rightarrow c_r)$. In fact, the negative association such as $(\bar{a}_r \Rightarrow c_r)$, $(a_r \Rightarrow \bar{c}_r)$ and $(\bar{a}_r \Rightarrow \bar{c}_r)$ can also provide valuable information as same as positive association rules. MCRAC takes the [illegible] of attributes into consideration in our algorithm as a basis for generating rules.

4.2 More General Rule

To make the classification more accurate and effective, MCRAC prunes rules whose information can be expressed by other simpler but more essential rules.

Definition 6. ***(More General Rule)*** [illegible] $r_1 : a_{r1} \Rightarrow c_{r1}$ [illegible] $r_2 : a_{r2} \Rightarrow c_{r2}$, [illegible] r_1 [illegible] ***more general than*** r_2 [illegible] ( ) $LaplaceAccuracy(r_1) \geq LaplaceAccuracy(r_2)$ ( ) $a_{r1} \subset a_{r2}$ [illegible] ( ) $c_{r1} = c_{r2}$

If r_1 is more general than r_2, that means r_1 does more contribution to classification but occupies smaller memory space. MCRAC just keeps these rules have higher rank and fewer attributes in its antecedent and other more specific rules with low rank should be pruned.

4.3 Our Algorithm

MCRAC inherits the basic idea of CPAR in rule generation and attributes' contributions estimation. It uses an exhaustive and greedy algorithm based FOIL to extract CARs directly from the training set. At each step, every possible literal (or its negative form) is evaluated and the best one is appended to the current rule. All literals in a rule must be a single form (all positive literals or all negative literals). Moreover, when selecting literals during the rule building process, there are usually many rules with similar gain based on the remaining dataset. Instead of selecting only the best one, MCRAC keeps all close-to-the-best literals in rules generation process so that it will not miss the some important rule. Moreover, MCRAC combines the first two phases of associative classification by pruning the weak correlated rules directly in the process of rules generation. The advantage is that this is not only more efficient (no post-pruning is necessary) but also more elegant in that it is a direct approach. Algorithm 1 gives the detailed pseudo-code for our algorithm.

Algorithm 1: Mining Correlated Rules for Associative Classification

```
Input: Training set $T$, Global attributes set $A$, Class labels set $C$
Output: Correlated Rules Set $R$
begin
    generate positive instance arrays $P$;
    $R \leftarrow \emptyset$;
    while (TotalWeight(P) > MIN_TOTAL_WEIGHT) do
        $A' \leftarrow A, T' \leftarrow T$, $a_r \leftarrow \emptyset$;
        while (1) do
            foreach literal $l_i \in A'$ do CalculateFoilGain($l_i$) in $T'$;
            $l$ = AttributeOfBestGain();
            if $l.gain \leq MIN\_BEST\_GAIN$ then break;
            $a_r \leftarrow l$;
            gainThreshold = bestGain*GAIN_SIMILARITY_RATIO;
            foreach $l' \in A'$ do
                if $l'.gain \leq gainThreshold$ then $a_r \leftarrow l'$
            end
            remove $a_r$ from $A'$;
            remove each $t \in T'$ that do not satisfy $a_r$;
        end
        $\phi = Correlation(a_r, c)$;
        if $\phi \geq \phi_{min}$ then
            if $NoMoreGeneralRule(a_r \Rightarrow c)$ then $R \leftarrow R \cup (a_r \Rightarrow c)$;
            if $NoMoreGeneralRule(\bar{a}_r \Rightarrow \bar{c})$ then $R \leftarrow R \cup (\bar{a}_r \Rightarrow \bar{c})$;
        if $\phi \leq -\phi_{min}$ then
            if $NoMoreGeneralRule(\bar{a}_r \Rightarrow c)$ then $R \leftarrow R \cup (\bar{a}_r \Rightarrow c)$;
            if $NoMoreGeneralRule(a_r \Rightarrow \bar{c})$ then $R \leftarrow R \cup (a_r \Rightarrow \bar{c})$;
        foreach $t \in T$ satisfy $a_r$ do
            reduce $t.weight$ by a decay factor
        end
    end
end
```

Finally, in the phase of classification, MCRAC uses a set of best rules to give good prediction for new data object so that important rules will no be missed. But instead of taking all rules satisfying it into consideration, MCRAC just selects a small set of strong classification rules to make prediction by the following procedure: (1) For rules satisfying the new data in antecedent, selects the best k rules for each class according to rules' consequents; (2) calculate the rank of each group by summing up the ϕ correlation coefficient of relevant rules. We think the average expected accuracy is not advisable because many trivial rules with low accuracy will weaken the effect of the whole group; and (3) the class label of new data will be assigned by that of the highest rank group. The experimental results will show that a small number of classification rules is very desirable. And the classification phase becomes faster with a small rules set, which can be important for some applications. Another advantage is that a small set becomes human-readable.

5 Experimental Results and Performance Study

To evaluate the accuracy and efficiency of MCRAC, we have performed an extensive performance study on some datasets from UCI Machine learning Repository [13]. It would have been desirable to use the same datasets as those used by CBA, CMAR and CPAR; however it was discovered that many of these datasets were no longer available in the UCI repository. A 10-fold cross validation was performed on each dataset and the results are given as average of the accuracies obtained for each fold. In addition, to have a fair comparison with the other algorithms that we wanted to compare, we used the same discretization method for continuous attributes as in [2]. The parameters of MCRAC are set as the following. In the rule generation algorithm, _ _ is set to 0.05, _ _ to 0.7, _ _ to 0.99, $\phi_{\min}$ to 0.2 and decay factor to 2/3. The best 3 rules are used in prediction. All the experiments are performed on a 2.4GHz Pentium-4 PC with 512MB main memory. The experimental results for C4.5, CBA, CMAR and CPAR are taken from [3] and [4].

The table 2 gives the average predictive accuracy for each algorithm respectively. **Bold** values denote the best accuracy for the respective dataset. The first four columns give the name, the number of attributes, classes and records of each dataset. And the **Column 9** shows the the accuracy of a MCRAC-like algorithm called MRAC without using correlated rules. **Column 10** gives the accuracy of MCRAC algorithm which uses the correlation measure.

Table 2. Comparison of Accuracies on C4.5, CBA, CMAR, CPAR and MCRAC

Datasets	♯attr	♯class	♯record	C4.5	CBA	CMAR	CPAR	MRAC	MCRAC
breast	10	2	699	95	96.3	**96.4**	96	95.68	94.83
hepatic	19	2	155	80.6	**81.8**	80.5	79.4	80.15	81.5
horse	22	2	368	82.6	82.1	82.6	**84.2**	82.3	83.59
iono	34	2	351	90	92.3	91.5	92.6	92.87	**93.44**
iris	4	3	150	95.3	94.7	94	94.7	94.5	**95.7**
led7	7	10	3200	73.5	71.9	72.5	73.6	71.06	**74.47**
pima	8	2	768	75.5	72.9	75.1	73.8	75.65	**77.1**
waveform	21	3	5000	78.1	80	**83.2**	80.9	77.12	78.38
wine	13	3	178	92.7	95	95	**95.5**	92.54	93.13
zoo	16	7	101	92.2	96.8	**97.1**	95.1	96	95.3
Average	-	-	-	85.55	86.38	86.79	86.58	85.787	**86.744**

As can be seen, MCRAC on almost all occasions achieves best accuracy or comes very close. Moreover, the classification accuracy increases when the correlated rules are taken into consideration. This is most noticeable for the Ionosphere, the Iris Plant and the Pima Indians diabetes datasets on which the correlated rules can give better supervision to the classification.

We also conducted another experiment to compare different rules selection strategies in associative classification. Table 3 shows the accuracies and runtime of MCRAC on waveform dataset by selecting different k best rules for classification. Based on these results, we observe that a small rules set with high quality and strong correlation is a preferable for classification, instead of relying on single one or all of them.

Table 3. Accuracies and Runtime of Different k Best Rules Selection on waveform

k	1	2	3	4	5	6	7	8	9	all
Accuracy	76.34	78.24	**78.8**	78.64	78.38	78.04	77.84	77.78	77.7	77.62
Runtime	**67.02**	67.2	67.44	67.63	67.7	67.81	68.04	68.23	68.35	68.48

6 Conclusions

Associative classification has been studied extensively in data mining and machine learning research. In this study, we have re-examined some previously proposed methodologies, and mainly focused on the new techniques developed for MCRAC, a new algorithm to discover correlated rules for associative classification, with high accuracy and more efficient. The thorough performance evaluation in this study reveals that: (1) For class association rules generation, Foil Gain is more economical and precise. It should be a preference over complex support setting when selecting contributing attributes to form a rule; (2) There is a popular truth that using a set of best rules to make prediction is better than just selecting the best one; and (3) The class distribution in the dataset and the correlated relationship among attributes should be taken into account because it is conforming to the usual or ordinary course of nature in the real world. In the future, we will explore more applications, including text categorization and Web user classification and in large databases.

References

1. Quinlan, J.R.: C4.5: Programs for Machine Learning. Morgan Kaufmann (1993)
2. Liu, B. Hsu, W. and Ma, Y.: Integrating Classification and Association Rule Mining. Proceedings KDD-98, New York, 27-31 August. AAAI. (1998) 80-86.
3. Li W., Han, J. and Pei, J.: CMAR: Accurate and Efficient Classification Based on Multiple Class-Association Rules. Proceedings of the 2001 IEEE International Conference on Data Mining, San José, California, USA, IEEE Computer Society (2001)
4. X. Yin and J. Han: CPAR: Classification based on Predictive Association Rules, Proc. 2003 SIAM Int.Conf. on Data Mining (SDM'03), San Fransisco, CA, May 2003.
5. Thabtah F., Cowling P.I, and Peng Y. MMAC: A New Multi-Class, Multi-Label Associative Classification Approach. Proceedings of the Fourth IEEE International Conference on Data Mining, Brighton, UK, pp. 217-224, Nov. 2004.
6. Bing Liu, Yiming Ma, Ching Kian Wong: Improving an Association Rule Based Classifier. Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD-2000) Lyon, France, LNAI 1910, (2000) 504-509
7. Elena Baralis, Silvia Chiusano, Paolo Garza: On support thresholds in associative classification. Proceedings of the 2004 ACM symposium on Applied computing (2004) 553-558.

8. J. R. Quinlan and R. M. Cameron-Jones: FOIL: A midterm report. In Proceedings of European Conference. Machine Learning, Vienna, Austria 1993, (1993) 3-20
9. Lavrač, N., Flach, P. and Zupan, B. (1999) Rule Evaluation Measures: AUnifying View. Proc. 9th Int. Workshop on Inductive Logic Programming(ILP'99), Sringer-Verlag, 174-185.
10. Sergey Brin, Rajeev Motwani, and Craig Silverstein: Beyond market baskets: Generalizing association rules to correlations. SIGMOD Record (ACM Special Interest Group on Management of Data), (1997) 265-276.
11. Tan, P., Kumar, V.: Interestingness measures for association patterns: A perspective. In Proceedings. of Workshop on Postprocessing in Machine Learning and Data Mining. (2000)
12. W. Hopkins. A new view of statistics. http://www.sportsci.org/resource/stats/, 2002.
13. Blake, C.L. and Merz, C.J. (1998). UCI Repository of machine learning databases. http://www.ics.uci.edu/ mlearn/MLRepository.html, Irvine, CA: University of California, Department of Information and Computer Science.

A Comprehensively Sized Decision Tree Generation Method for Interactive Data Mining of Very Large Databases

Hyontai Sug

Division of Internet Engineering, Dongeo University,
Busan, 617-716, South Korea
sht@dongseo.ac.kr

Abstract. For interactive data mining of very large databases a method working with relatively small training data that can be extracted from the target databases by sampling is proposed, because it takes very long time to generate decision trees for the data mining of very large databases that contain many continues data values, and size of decision trees has the tendency of dependency on the size of training data. The method proposes to use samples of confidence in proper size as the training data to generate comprehensible trees as well as to save time. For medium or small databases direct use of original data with some harsh pruning may be used, because the pruning generates trees of similar size with smaller error rates.

1 Introduction

Decision trees have been used for good classification tasks, so finding trees with the smallest error rates for a given data set has been a major task for their success. But, since KDD (Knowledge Discovery in Databases) problems often contain a tremendous amount of data, the generated trees may be too large for a user to understand, even though the trees achieve reasonable error rates. Moreover, because the target domain of KDD is real world, it is well known that there are many meaningless/useless branches in the generated trees [1].

As a way to generate meaningful decision trees, interactive tree generation process is widely used, and as a way of interactive tree generation process, focusing on interesting nodes only attracted researchers' attention [2], [3]. But the approach is prone to be myopic, because the interaction process is performed node by node, and since the problem of finding the smallest decision tree is a NP-complete problem, there is a tremendous number of possible choices for the root attribute of each subtree. Another method to solve this problem may be to show the user some smaller decision trees from samples to make them more understandable. But sampling has an innate problem—sampling errors. So, this paper presents a novel idea to use samples without being much affected by sampling errors so that it allows the user to refine some of the interesting branches further as a part of interactive tree generation process in knowledge discovery.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 141 – 148, 2005.

Conventional methods to generate smaller decision trees rely mostly on pruning, with overpruning performed to obtain trees smaller than a specified size [4]. According to our experiment using C4.5, the overpruned trees are bigger and take far more computing time than the trees generated by our method when both trees have similar error rates.

The remainder of this paper presents related work in section 2; our suggesting method in section 3; some experiments with the method in section 4; and, finally, the conclusion in section 5.

2 Related Work

Decision tree algorithms are based on greedy method. So, generated decision trees are not optimum and some improvement may be possible. There have been a lot of efforts to build better decision trees with respect to error rates. For example, one of standard decision tree algorithm C4.5 [4] uses entropy-based measure, and CART [5] uses purity-based measure, and CHAID [3] uses chi-square test-based measure for split.

There have been also scalability related efforts to generate decision trees for large databases such as SLIQ [6], SPRINT [7], PUBLIC [8], and RainForest [9]. SLIQ saves computing time especially for continuous attributes by using a pre-sorting technique in tree-growth phase, and SPRINT is an improved version of SLIQ to solve the scalability problem by building trees in parallel. PUBLIC tries to save some computing time by integrating pruning and generating branches. The authors of PUBLIC claimed that their algorithm is more efficient than SPRINT. RainForest saves more computing time than SPRINT when certain minimum amount of main memory is available. According to literature [8], for very large data sets computing time becomes exponential for SPRINT and polynomial for PUBLIC despite of their efforts for scalability. Moreover, these methods may generate very large decision trees for very large data sets so that comprehensibility problem can occur.

Generating right-sized decision trees requires a universal application of pruning [1], [4], [5], [10] so that overpruning was a natural consequence to generate comprehensively sized decision trees. In his Ph.D. dissertation, ‘mega induction’ for very large databases [1], J. Catlett relied on overpruning to obtain comprehensible trees. As a result of this overpruning, the generated tree may not have sufficient accuracy compared to near optimal, similar sized trees.

Another simple method to use, when no explicit post processing for pruning is applied, is to stop the tree generation if the tree size becomes larger than some maximum allowable size. But, this method has similar problem with that of overpruning.

There has been also a lot of research for feature subset selection and dimensionality reduction problem [11]. One of major problem in scientific data is that the data are often high dimensional so that it takes very long computing time for pattern recognition algorithms. To solve this problem dimensionality reduction algorithms have been invented to select the most important features so that further processing like pattern recognition algorithms can be simplified without comprising the quality of final results. On the other hand, feature subset selection algorithms try to eliminate irrelevant features from attribute set [12], [13], [14]. Irrelevant features are features that are dependent on other features so that they have bad effects on their size and error rate of generated trees. If there are n features, we can have 2^n possible feature subsets, so feature subset selection is a hard problem. There are two directions to deal with the computation problem—wrapper approach and filter approach. The wrapper approach adds features to the set of good features incrementally based on test results of underlying algorithms. It needs $O(n^2)$ runs of the algorithm to test. The filter approach uses some heuristic to select good subset of features, so it's faster than the wrapper approach. Most dimensionality reduction algorithms are kinds of filter approach.

There are also efforts to find a best rule set by applying association rule finding algorithms [15], [16], [17]. ART [18] tries to ensure good scalability by building decision list efficiently. But, because association rule finding algorithms are exhaustive algorithms so that its applicability is limited. The largest data set used for experiment to test its performance has size of only 12,960. Moreover, association rule finding algorithms can deal with interval or nominal values only, so discretization is necessary as preprocessing. Even though some papers showed good performance of discretization in experiment [19], it is still an open problem. Some researchers prefer decision trees to association rules because of decision tree algorithms' capability to deal with continuous values [20].

Decision trees are one of the mostly used data mining methods, because they have many good characteristics, especially the easy-to-understand structure. So, when there are a lot of training data, we want to appreciate the structure of decision trees expressed in smaller size, as well as to provide more economical way of prediction by providing shorter building time of decision trees. The difference of our approach from feature subset selection or dimensionality reduction algorithms is that our method can be applied to feature-selected decision trees also so that the utility of the algorithms can be strengthened.

3 Generating Comprehensively Sized Decision Trees

If a target database has n attributes, a hypothesis that can be extracted from the target database has up to n attributes. Because each attribute of the hypothesis can have different number of values, let the first attribute have a_1 number of values, the second attribute have a_2 number of values, and so on, then the size of hypothesis space becomes

$$1 + \{a_1 + a_2 + \cdots + a_n\} + \{(a_1 \cdot a_2 + a_1 \cdot a_3 + \cdots + a_1 \cdot a_n) + (a_2 \cdot a_3 + a_2 \cdot a_4 + \cdots + a_2 \cdot a_n) + \cdots + (a_{n-1} \cdot a_n)\} + \cdots + \{a_1 \cdot a_2 \cdots a_n\}$$
$$= 1 + \sum_{i = 1 \sim n} a_i + \sum_{i1 = 1 \sim n-1} \sum_{i2 = i1+1 \sim n} a_{i1} \cdot a_{i2} + \cdots + \sum_{i1 = 1 \sim 1} \cdots \sum_{in = in+1 \sim n} a_{i1} \cdots a_{in}.$$

The first term in the above equation is for null hypothesis, and the second group of equation, $\{a_1 + a_2 + \cdots + a_n\}$, is for hypotheses that have one attribute, and the third group of equation, $\{(a_1 \cdot a_2 + a_1 \cdot a_3 + \cdots + a_1 \cdot a_n) + (a_2 \cdot a_3 + a_2 \cdot a_4 + \cdots + a_2 \cdot a_n) + \cdots + (a_{n-1} \cdot a_n)\}$, is for hypotheses that have two attributes, and so on. The final term is the same as $a_1 \cdot a_2 \cdots a_n$. Therefore, the above equation becomes

$$1 + \sum_{i = 1 \sim n} a_i + \sum_{i1 = 1 \sim n-1} \sum_{i2 = i1+1 \sim n} a_{i1} \cdot a_{i2} + \cdots + a_1 \cdot a_2 \cdots a_n. \tag{1}$$

In PAC (Probably Approximately Correct)-learning theory [21] the sample size m can be represented by

$$m \geq (1/\varepsilon) \cdot (ln(1/\delta) + ln\,|\mathbf{H}|) \tag{2}$$

where ε and δ are small constants, and **H** is the set of possible hypotheses, and m is the number of training examples in the training set. In other words, hypotheses that are consistent with at least m training examples have error at most ε with probability at least 1 - δ. So, based on |**H**| which can be calculated by equation (1), we may set m as a sample size with some small values of ε and δ for a reliable decision tree. ε and δ are used as reference values for the reliable sample size. In addition, because we are dealing with large databases, which means we have a lot of training examples, it may not be a problem to set large m for reliability.

The following is a brief description of the procedure of the method:

```
INPUT: a database in table form.
OUTPUT: a smaller decision tree of user's interest.
1. Determine a sample size m based on the PAC-learning theory for input.
2. Do random sampling.
3. Generate a decision tree and determine a focusing area.
4. Select data set for the focusing area.
5. If the size of data set > predefined size limit Then
       Let the selected data set be new input and go to step 1. /* loop */
       Else generate a decision tree.
   End if
```

By selecting focusing area not by selecting node by node, we may avoid the myopia phenomenon in the interactive decision tree generation process.

At step 4 of the above procedure data selection for a focusing area can be done by selecting data set that has matching values in the database table with interesting branch of the generated decision tree. For example, if user selects a subtree that has root of C=c and the subtree has ancestor nodes of A=a and B=b, we select data records that have attribute A's value is 'a' and attribute B's value is 'b' only so that next time we focus on interesting part of data only. Thus, we'll have smaller data set for data mining. At step 5 of the above procedure the treatable data set size is dependent upon the available computing resources.

4 Experimentation

An experiment was run using a database in UCI machine learning repository [22] called 'census-income' to see the effect of the method. The number of instances for training is 199,523 in size of 99MB data file, and the number of instances for testing is 99,762 in size of 49.5MB. Class probabilities for label -50000 and 50000+ are 93.8% and 6.2% respectively.

The database was selected because it is relatively very large and contains lots of manifest facts and continuous values. The total number of attributes is 41. Among them eight attributes are continuous attributes. According to equation (2), $|\mathbf{H}|$ is 4.03×10^{47}. All continuous attribute values are counted, because each value constitutes potential hypothesis.

We used C4.5 to generate decision trees from various sample sizes. Other decision tree algorithms like SPRINT, PUBLIC, or RainForest were not considered for experiment, because C4.5 is widely accepted to become a de facto standard and freely available. In addition, even though the scalable algorithms may save some computing time, they also have similar comprehensibility problem like C4.5, so we didn't use the algorithms. The following Table 1 shows tree sizes and error rates depending on sample size. The used computer is Sun Blade1000 workstation. We did not run cross validation in the experiment, because the test data is big enough.

It took more than 8 hours to generate a tree for the 1/3 sample, while it took only 17 seconds for the 1/50 sample, so it may be unpractical to use very large data. The confidence (CF) for pruning is a default value of 25%.

Table 1. Decision Trees and Error Rates by C4.5 with Various Sample Sizes for 'census-income' Database

Sample size	ε & δ	Tree size (before pruning)	Tree size (after pruning)	Error Rate(%)	Computing Time
1/200	0.112	260	1	6.2	2 sec.
1/100	0.0564	530	3	5.8	4 sec.
1/50	0.0284	1,252	10	5.7	17 sec.
1/25	0.0143	2,392	114	5.4	46 sec.
1/12	0.0072	5,171	214	5.3	195 sec.
1/6	0.0036	10,570	326	5.1	43 min. 42 sec.
1/3	0.0018	19,999	1,106	4.9	8 hours 15 min.

Table 2. The Summary of Decision Trees based on Different Levels of Pruning Confidence(CF) by C4.5 with the 1/12 size sample for 'census-income' Database

CF(%)	Tree size (before pruning)	Tree size (after pruning)	Error Rate(%)
25	5,171	214	5.3
5	5,171	11	5.5
1	5,171	11	5.5
0.2	5,171	3	5.7
0.05	5,171	3	5.7

Table 3. The Summary of Decision Trees based on Different Levels of Pruning Confidence(CF) by C4.5 with the 1/3 size sample for 'census-income' Database

CF(%)	Tree size (before pruning)	Tree size (after pruning)	Error Rate(%)
25	19,999	1,106	4.9
5	19,999	87	5.4
1	19,999	11	5.7
0.2	19,999	11	5.7
0.05	19,999	11	5.7

For comparison with the pruning based simple tree generation method, we generated decision trees for several levels of pruning confidence with a sample of 1/12 and 1/3 size of original training data. Note that the lower the pruning confidence is, the severer the pruning is. The low limit of the confidence is 0.001% in C4.5. The results are shown in Table 2 and Table 3 respectively.

As shown in table 2 and Table 3, we found that smaller CF values or harsher pruning did not improve the decision trees much:

1. If you compare Table 2 and Table 3, we can find that pruning confidences below 5% do not show better results, even though we use larger samples.
2. If we compare Table 1 and Table 3, the tree size of 1/3 sample with pruning CF=1% is similar to the tree size of 1/50 sample. But the computing time for 1/3 sample was 8 hours, while that of 1/50 sample was 17 seconds with the Sun Blade1000 workstation. So, interactive data mining task may be almost impossible for large databases.

Therefore, we have the following provisional conclusions from the result of our experiments:

1. When the size of training examples are relatively not very large compared to available computing resources, for example, below 10MB or so, when used computer is Sun Blade 1000 workstation or similar one and used decision tree algorithms is C4.5 or similar one, it is better to use the original training examples and do severe pruning like CF=5% than to use samples to generate user understandable decision trees.
2. When the size of training examples are very large, it is better to use samples, because tree size is dependent on sample size and computing time may be prohibitive.

The provisional conclusion in database size may be slightly different for each used computer and decision tree software, because response time is also dependent on available hardware and software.

5 Conclusion

In general, the target databases of KDD may have many manifest facts and may be very large. As a result, a direct application of the decision tree generation methods may generate very large trees with numerous meaningless subtrees or branches, resulting in trees that are not as useful as expected. Moreover, generating decision trees for very large database with many continuous data values takes very long computing time even for high speed computers, so that data mining task becomes unrealistic. If the size of database is very large, sampling is an alternative choice, because samples of smaller size generate smaller decision trees in reasonable computing time.

By showing decision trees that are made small by using samples with reliability, a user can grasp a more general view of the original data set, hence the user's understandability over the generated trees will be enhanced. So it will be possible for the user to be able to get a better decision tree of his interest by further interactive process. In addition, we had better do some severe pruning on decision trees from original data set than sampling for the case of moderately sized databases, because samples may contain sampling errors.

References

1. Catlett, J., Megainduction: Machine Learning on Very Large Databases. PhD thesis, University of Sydney, Australia (1991)
2. SPSS: Clementine 8.0 User's Guide Package. SPSS inc. (2004)
3. StatSoft, Inc.: Electronic Statistics Textbook. Tulsa, OK, StatSoft. WEB: http://www.statsoft.com/textbook/stathome.html (2004)
4. Quinlan, J.R.: C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, Inc. (1993)
5. Breiman, L., Friedman, J., Olshen, R. and Stone, C.: Classification and Regression Trees. Wadsworth International Group (1984)
6. Mehta, M., Agrawal, R., and Rissanen, J.: SLIQ : A Fast Scalable Classifier for Data Mining. (EDBT'96), Avignon, France (1996)
7. Shafer, J., Agrawal, R., and Mehta., M.: SPRINT : A Scalable Parallel Classifier for Data Mining. Proc. 1996 Int. Conf. Very Large Data Bases, Bombay, India, Sept. 1996. 544-555.
8. Rastogi, R., Shim, K.: PUBLIC : A Decision Tree Classifier that Integrates Building and Pruning. Data Mining and Knowledge Discovery, Vol. 4, no. 4. Kluwer International (2002) 315-344
9. Gehrke, J., Ramakrishnan, R., and Ganti, V.: Rainforest: A Framework for Fast Decision Tree Construction of Large Datasets. Proc. 1998 Int. Conf. Very Large Data Bases, New York, NY, August 1998. 416-427
10. SAS, Decision Tree Modeling Course Notes. SAS Publishing (2002)
11. Jolliffe, I.T.: Principal Component Analysis. Springer Verlag, 2nd ed. (2002)
12. Almuallim, H., Dietterich, T.G.: Efficient Algorithms for Identifying Relevant Features. Proc. of the 9th Canadian Conference on Artificial Intelligence (1992) 38-45
13. Kononenko, I., et. al.: Overcoming the Myopia of Inductive Learning Algorithms with RELIEF. Applied Intelligence, Vol.7, no. 1 (1997) 39-55
14. Liu, H., Motoda, H.: Feature Extraction, Construction and Selection: A Data Mining Perspective. Kluwer International (1998)
15. Liu, B., Hsu, W., Ma, Y.: Integrating Classification and Association Rule Mining. Proc. of the 4th International Conference on Knowledge Discovery and Data Mining (KDD-98), New York, New York, (1998) 80-86

16. Liu, B., Hu, M., Hsu, W., Multi-level Organization and Summarization of the Discovered Rule. Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA (2000) 208-217
17. Wang, K., Zhou, S., He, Y.: Growing Decision Trees on Support-less Association Rules. Proc. of the 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA (2000) 265-269
18. Berzal, F., Cubero, J., Sanchez, D., Serrano, J.M.: ART: A Hybrid Classification Model. Machine Learning, Vol. 54 (2004) 67-92
19. Liu, H., Hussain, F., Tan, C.L., Dash, M.: Discretization: An Enabling Techniques. Data Mining and Knowledge Discovery, Vol. 6, no. 4 (2002) 393-423
20. Witten, I.V., Frank, E.: Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations. Morgan Kaufmann Publishers (2000)
21. Russel, S., Norvig, P.: Artificial Intelligence: A Modern Approach, 2nd ed. Prentice Hall, Inc. (2002)
22. Hettich, S., Bay, S.D.: The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science (1999)

Using Latent Class Models for Neighbors Selection in Collaborative Filtering

Xiaohua Sun, Fansheng Kong, Xiaobing Yang, and Song Ye

Institute of Artificial Intelligence, Zhejiang University,
Hangzhou, Zhejiang, China, 310027
{sunxh, kfs, yesong}@zju.edu.cn
konglab@cise.zju.edu.cn

Abstract. Collaborative filtering is becoming a popular technique for reducing information overload. However, most of current collaborative filtering algorithms have three major limitations: accuracy, data sparsity and scalability. In this paper, we propose a new collaborative filtering algorithm to solve the problem of data sparsity and improve the prediction accuracy. If the rated items amount of a user is less than some threshold, the algorithm utilizes the output of latent class models for neighbors selection, then uses the neighborhood-based method to produce the prediction of unrated items, otherwise it predicts the rating using the STIN1 method. Our experimental results show that our algorithm outperforms the conventional neighborhood-based method and the STIN1 method.

1 Introduction

Collaborative filtering is becoming a popular technique for reducing information overload; it is an approach to make recommendations by finding correlations among users of a recommender system. The problem of collaborative filtering is to predict how well a user will like an item that he has not rated given that users ratings for other items and a set of historical ratings for a community of users[8].

The most prevalent algorithms used in collaborative filtering are what we call the neighborhood-based methods[8]. In the neighborhood-based methods, a subset of appropriate users are chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user. Other algorithmic methods that have been used are Bayesian networks[2], dimension reduction methods, such as singular value decomposition[14], inductive rule learning[1], a Bayesian mixed effect model[5], and a combination of neighborhood-based algorithms with weighted majority weighting[6].

Most of current algorithms have three major limitations: accuracy, data sparsity and scalability[14]. Usually nearest neighbor algorithms rely upon exact matches that cause the algorithms sacrifice coverage and accuracy. In particular, since the correlation coefficient is only defined between users who have rated at least two items in common, many pairs of users have no correlation at all, it causes sparsity of recommender system also. Nearest neighbor algorithms require computation that grows with both the number of users and the number of items. With millions of users

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 149 – 156, 2005.

and items, a typical web-based recommender system running existed algorithms will suffer serious scalability problem.

In this paper, we propose a new collaborative filtering algorithm to solve the problem of data sparsity and improve the prediction accuracy. We utilize the output of latent class model for neighbor selecting, then use the neighborhood-based method to produce prediction of unrated items. In order to eliminate the effect of the average or the norm of original ratings on neighbors selecting, we use a map to subtract from their means and divide by their norm.

The rest of the paper is organized as follows. The next section describes the background and notation associated with our algorithm. In Section 3, we give a short review of latent class models, neighborhood-based algorithms and STIN1 algorithm. In Section 4, we propose our algorithm. Then, we evaluate our algorithm on a dataset of user ratings for movies in Section 5, and show that our approach outperforms the conventional neighborhood-based method and the STIN1 method. Finally, we conclude the paper and discuss further developments in the last section.

2 Background and Notation

2.1 Description of the Recommender System

The problem of collaborative filtering is to predict the like of a user to an item that he has not rated given that users ratings for other items and a set of historical ratings for a community of users. Assuming a recommender system with m users and n items, the system can be represented by an $m \times n$ matrix. We define the rating of user i on item j is $R_{i,j}$. So collaborative filtering can be regarded as the problem of predicting missing values in the user-item matrix[8]. We denote a user as an active user if the recommender system is predicting his rating on an item.

2.2 Lebesgue Norms

For the purpose of this paper, we define Lebesgue norms[11] of a vector X for $p = 1, 2, \hbar$ as

$$\| \mathrm{X} \|_{l_p} = \sqrt[p]{\sum_{i=1}^{N} \frac{|\mathrm{X}_i|^p}{N}} \qquad (1)$$

Where the sum is over all indexes of X.

3 Related Work

3.1 Latent Class Models

Latent Class Model (LCM) is a statistical method for finding subtypes of related cases (latent classes) from multivariate categorical data. The Expectation Maximization (EM) algorithm[7] is typically used to train LCMs. The EM algorithm is very useful for computing maximum likelihood estimation of parameters when data is partially observed. Each iteration of the EM algorithm involves two steps: the Expectation (E) step and the Maximization (M) step. In the E step, the posterior probabilities are computed based on the current estimates of the parameters. In the M step, the parameters are determined using the posterior probabilities computed in the previous E step.

3.2 Related Collaborative Filtering Approaches

3.2.1 Neighborhood-Based Approach

In neighborhood-based methods, a subset of appropriate users are chosen based on their similarity to the active user, and a weighted aggregate of their ratings is used to generate predictions for the active user. The GroupLens[13] system first introduced a collaborative filtering system using a neighborhood-based algorithm. GroupLens provides personalized predictions for Usenet news articles. The GroupLens system uses Pearson correlations to weight user similarity and all available correlated neighbors to compute a final prediction by performing a weighted average of deviations from the neighbor's mean.

3.2.2 STIN1 Approach

Lemire[11] introduced a Scale and Translation Invariant (STI) approach. He defines the first-order STI Non personalized (STIN1) scheme with

$$v_i^{(1)} = \frac{1}{card(S_i(\chi))} \sum_{u \in S_i(\chi)} m_{\|\cdot\|}(u)_i \tag{2}$$

Where $v_i^{(1)}$ is the i^{th} component of $v^{(1)}$ and $card(S_i(\chi))$ is a short-hand for the number of evaluations u meeting the conditions that item $i \in S(u)$. $m_{\|\cdot\|}$ is a map from all incomplete vectors to R_n denoted by

$$m_{\|\cdot\|}(u)_i = \begin{cases} \dfrac{u_i - \bar{u}}{\| (u_k - \bar{u})_{k \in S(u)} \|} & i \in S(u) \\ 0 & i \notin S(u) \end{cases} \tag{3}$$

Where $\| \cdot \|$ indicates Lebesgue norm.
Defining

$$v_u^{(1)} = v^{(1)} - \overline{v_{|S(u)}^{(1)}} \tag{4}$$

Then we can get the prediction of evaluation u

$$P_{STIN1}(u) = \bar{u} + \frac{< u, v_u^{(1)} >}{< v_u^{(1)}, v_u^{(1)} >_{S(u)}} v_u^{(1)} \tag{5}$$

The scale and translation invariant algorithm outperforms other learning-free constant time schemes as well as expensive memory-based schemes. Please refer to [11] for details of the STIN1 algorithm.

4 Our Algorithm

LCM defines latent classes by the criterion of "conditional independence". This means that, within each latent class, each variable is statistically independent of every other variable[4]. We use LCM to capture the latent relationships between users and items that allow us to compute the predicted likeliness of a certain item by a user.

From [11], we know that a STI scheme usually outperforms or at least matches the performance of the corresponding non-STI scheme. But we found that the accuracy of the STIN1 approach was not as well as the Pearson method if the rated items number of per user is small. So we want to combine the LCM and the conventional neighborhood-based method to predict the unknown ratings. We use neighborhood-based method to predict the ratings of unrated items for an active user when the rated items amount of the user is less than an appropriate threshold t, otherwise we predict the rating using the STIN1 algorithm[11]. We call our algorithm LCM_STI.

The procedure of LCM_STI is described as follows:

1. Preprocess: Set a threshold and map ratings to normal form.
2. Select the neighborhood: Using the Tempered Expectation Maximization (TEM) algorithm proposed by Hoffmann[9] to train the latent class model, then select the neighborhood based on the output of the latent class model.
3. Make predictions: Make predictions for unrated item by performing Pearson algorithm or STIN1 algorithm.

4.1 Preprocess

First we set the average amount of rated items of all users to threshold t, based on which we decide to use Pearson or STIN1 algorithm. Then we map the original rating matrix R to a new matrix R' using equation (3).

4.2 Neighbor Selecting

We use the Tempered Expectation Maximization (TEM) algorithm proposed by Hoffmann[9] to train the latent class model and get $P(y \mid x)$, where x is a user and y is an item. The input of the latent class model is the processed matrix R'. Each iteration of the algorithm consists of two steps: E step, where posterior probabilities are computed for the latent variable z, based on the current estimates of the parameters, and M step, where parameters are determined using the posterior probabilities computed in the previous E step.

E step:

$$P(z \mid x, y) = \frac{P(z)P(x \mid z)p(y \mid z)}{\sum_{z'} P(z')P(x \mid z')p(y \mid z')} \tag{6}$$

M step:

$$P(z) = \frac{\sum_{x',y'} R_{x,y} P(z \mid x', y')}{\sum_{x',y',z'} R_{x,y} P(z' \mid x', y')} \tag{7}$$

$$P(y \mid z) = \frac{\sum_{x'} R_{x',y} P(z \mid x', y)}{\sum_{x',y'} R_{x',y'} P(z \mid x', y')} \tag{8}$$

$$P(x \mid z) = \frac{\sum_{y'} R_{x,y'} P(z \mid x, y')}{\sum_{x',y'} R_{x',y'} P(z \mid x', y')} \tag{9}$$

Then the probability that a user x buys an item y can be computed as

$$P(y \mid x) = \sum_{z' \in Z} P(z' \mid x) P(y \mid z') \tag{10}$$

Where

$$P(z \mid x) = \frac{P(x \mid z)P(z)}{\sum_{z' \in Z} P(x \mid z')P(z')} \tag{11}$$

We can order the probabilities $P(y \mid x)$ from the biggest to the smallest. After that, we select n users x associated with the largest $P(y \mid x)$ as neighbors.

4.3 Make Predictions

If the rated items amount of the active user is less than threshold t, we predict the rating of item y of the user using the neighborhood-based Pearson method. In the Pearson algorithm, the predicted rating of the active user on item j, $P_{a,j}$, is a weighted sum of the ratings of the other users[2]:

$$P_{a,j} = \overline{R_a} + \kappa \sum_{u=1}^{n} w_{a,u} (R_{u,j} - \overline{R_u}) \tag{12}$$

Where, κ is a normalizing factor such that the absolute values of the weights sum to unity. $R_{a,i}$ represents the rating of the active user for item i and $\overline{R_a}$ is the average rating of the active user over all items he had rated. $w_{a,u}$ denotes the Pearson correlation coefficient between the active user and neighbor u , we substitute correlation $w_{a,u}$ with $P(y \mid x)$ in our case. n is the number of neighbors.

If the rated items number of the active user is greater or equal to threshold t, we predict the rating of item y of the user using STIN1 method with equations (3)-(5).

Our algorithm overcomes the sparsity problem by utilizing the latent relationships between users and items; it utilizes the advantages of the STI scheme also.

5 Experiments

5.1 Dataset

In order to evaluate the above approach for collaborative filtering, we use the data from EachMovie[12]. The EachMovie database is a public available database that has 72916 users entered a total of 2811983 numeric ratings for 1628 different movies. User ratings were collected on a numeric 6-point scale between 0 and 1, where 0 is the worst. To make the results comprehensible, we rescaled ratings to integers. Our task is to provide an estimated rating for a previously unseen movie. We select a sub dataset from the EachMovie dataset, including those rating whose user number is less than 1000. The ratio of ratings numbers in the training set and the test set is about 0.8:0.2.

5.2 Evaluation Metrics

The evaluation of a collaborative filtering algorithm usually focuses on its accuracy. We discuss two types of metrics for evaluating the prediction result. First, we use a popular statistical accuracy metric, Mean Absolute Error (MAE) [14] to evaluate the accuracy of our algorithm. The MAE E of a recommender system is evaluated by the equation:

$$E = \frac{1}{N}\sum_{i=1}^{N} | P_i - T_i | \tag{13}$$

Where N is the total number of unrated user-item entries of a recommender system. P_i is the value predicted by the system for an unrated user-item entry; and T_i is the target value for unrated user-item entry. The lower the MAE, the more accurately the recommendation engine predicts user rating.

Then we use Receiver Operating Characteristic (ROC) curve [10] to evaluate how effectively predictions help a user to select high-quality items from the item set. ROC is the only available measure that is uninfluenced by decision biases and probabilities. The more the area under the ROC curve, the more accuracy the recommender system is. To use it as a metric, we must determine which items are “good” and which are “bad” [8]. In our experiments, we consider rating of above 3 indicates “good” signal else it is noise.

5.3 Result and Discussion

The optimal number of latent classes is crucial for controlling the model complexity of LCM so that they should be flexible enough to capture the true patterns but at the same time strict enough to avoid spurious patterns due to noise in the data[3]. All the experiments assume the number of latent classes K is set to 3, threshold t is set to

20 in LCM_STI algorithm which is the average amount of rated items of all users. The neighbors' number varies from 1 to 100 in Pearson and LCM_STI algorithms. We set the stop inverse computation temperature[9] to 0.60.

Figure 1 shows the results of experiments of the dataset. It can be observed from the results that the MAE of LCM_STI method is greatly less than the STIN1 method if the number of neighbors is above 10; meanwhile the area under the ROC curve of LCM_STI algorithm is greater than that of the STIN1 method if the number of neighbors is above 13. LCM_STI algorithm also performs better than the Pearson algorithm.

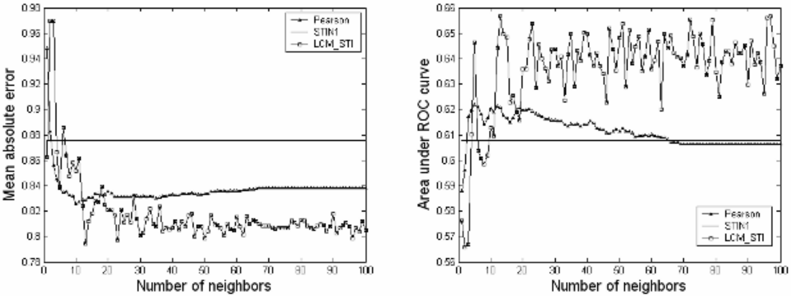


Fig. 1. The prediction results of the EachMovie dataset: MAE (*left*) and ROC (*right*)

From the experimental results, we can show that LCM_STI has potential to provide better performance than the Pearson algorithm, it also have more accurate prediction than the STIN1 algorithm.

6 Conclusion

In this paper, we used the result of latent class model for neighbors selecting, then we use the neighborhood-based approach to produce prediction of unrated items. Experimental results show that the proposed approach outperforms the Pearson method and STIN1 method. Our further work will be to study how to improve the performance with more data and how to integrate with content-based algorithms to enhance the prediction accuracy of a recommender system.

Reference

1. Basu, C., Hirsh, H., and Cohen, W.: Recommendation As Classification: Using Social and Content-based Information in Recommendation. In Proceedings of the 1998 Workshop on Recommender Systems, AAAI Press (1998) 11-15
2. Breese, J. S., Heckerman, D., and Kadie, C.: Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (1998) 43-52

3. Cheung, K.-W., Tsui, K.-C., and Liu, J.: Extended Latent Class Models for Collaborative Recommendation. Systems, Man and Cybernetics, Part A, IEEE Transactions on System, Man, and Cybernetics. Vol.34 (2004)
4. Compuserve: LCA Frequently Asked Questions. http://ourworld.compuserve.com/homepages/jsuebersax/faq.htm (2004)
5. Condliff, M. K. and Lewis, D. D.: Bayesian Mixed-effects Models for Recommender Systems. In Proceedings of the SIGIR-99 Workshop on Recommender Systems: Algorithms and Evaluation, Berkeley, CA (1999)
6. Delgado, J. and Ishii, N.: Memory-based Weighted-majority Prediction for Recommender Systems. In Proceedings of the ACM SIGIR-99, Recommender Systems Workshop, UC Berkeley (1999) 1-5
7. Dempster, A., Laird, N., and Rubin, D: Maximum Likelihood from Incomplete Data via the EM algorithm. Journal of the Royal Statistical Society, Series B, 39 (1977) 1-38
8. Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J.: An Algorithmic Framework for Performing Collaborative Filtering. In Proceedings of ACM SIGIR'99, ACM press (1999)
9. Hofmann, T.: Probabilistic Latent Semantic Analysis. In Proceedings of the 15th Conference on Uncertainty in AI (1999)
10. John A.Swets: Measuring the Accuracy of Diagnostic System. Science, Vol.240, (1988) 1285-1289
11. Lemire, D.: Scale and Translation Invariant Collaborative Filtering Systems. Information Retrieval, Vol.7, (2004) 1-22
12. McJones, P.: EachMovie Collaborative Filtering Data Set. DEC Systems Research Center (1997)
13. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., and Riedl, J.: GroupLens: An Open Architecture for Collaborative Filtering of Netnews. In Proceedings of CSCW '94, Chapel Hill, NC (1994)
14. Sarwar, B. M., Karypis, G., Konstan, J. A., and Riedl, J.: Application of Dimensionality Reduction in Recommender Systems-A Case Study. In ACM WebKDD 2000 Web Mining for E-Commerce Workshop (2000)

A Polynomial Smooth Support Vector Machine for Classification*

YuBo Yuan and TingZhu Huang

School of Applied Mathematics, University of Electronic Science and
Technology of China, Chengdu, Sichuan, 610054, P. R. China
ybyuan@uestc.edu.cn

Abstract. A new polynomial smooth method for solving the support vector machine (SVM) is presented in this paper. It is called the polynomial smooth support vector machine (PSSVM). BFGS method and Newton-Armijo method are applied to solve the PSSVM. Numerical experiments confirm that PSSVM is more effective than SVM.

Keywords: data mining; classification; support vector machine; smooth methods; combinational optimal; BFGS method; Newton-Armijo method.

1 Introduction

Support vector machine (SVM) is one of important statistical learning technology ([1]-[3]). In statistical learning theory, the problem of consistency of learning procedure in machine learning is the one where the empirical risk converges uniformly to the actual risk. To obtain a small actual risk, i.e., a good generalization performance, the statistical learning theory shows that it is necessary to have a right balance between the empirical risk and the capacity of a learning machine. SVM can obtain a good generalization performance. SVM also has other attractive properties, for example, SVM has a unique global optimal solution and avoid the curse of dimensionality.

In recent years, many famous researchers were involved in the development of the SVM, such as A. Gammerman, V. Vapnik, Y. Le Cun, N. Bozanic, L. Bottou, C. Saunders, B. Schlköpf, A. Smola, M. O. Stitson, V. Vovk, C. Watkins, J. A. E. Weston, O. L. Mangarasian and so on. The SVM is applied to pattern recognition, regression estimation and other problems in information science.

It has been made some progress in classification ([2], [3]). The SVM can be formulated into a non-smooth unconstrained optimization problem (See in [2](Yuh-Jye Lee and O. L. Mangarasian (2001))) but the objective function is non-differentiable at zero. Yuh-Jye Lee and O. L. Mangarasian employed smooth method to solve the resulting optimization problem. They used the integral $p(x,k)$ of the sigmoid function $\frac{1}{1+e^{-kx}}$ to smooth the plus function x_+. It is a

* Supported by the Youth Key Foundations of Univ. of Electronic Science and Technology of China (Jx04042) and NCET of China.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 157–164, 2005.

very important and significative result to SVM since many famous algorithms can be used to solve it.

In this paper, polynomial functions with parameter k are introduced to smooth the converted objective function. For a given $k \in Z^{+}$, $q(x,k)$((2)), $h(x,k)$ ((3)) can approximate x_{+} infinitely with $k \to \infty$. Comparing the square difference between the smooth function and x_{+}, we have the following results:

i) The integral of sigmoid function ([2], Lemma 2.1):

$$p(x,k)^2 - x_{+}^2 \leq ((log2)^2 + 2log2)\frac{1}{k^2} \approx 0.6927\frac{1}{k^2};$$

ii) The quadratic polynomial smooth function((2),(4)):

$$q(x,k)^2 - x_{+}^2 \leq (\frac{1}{11})\frac{1}{k^2} \approx 0.0909\frac{1}{k^2};$$

iii) The forth polynomial smooth function((3),(5)):

$$h(x,k)^2 - x_{+}^2 \leq (\frac{1}{19})\frac{1}{k^2} \approx 0.0526\frac{1}{k^2}.$$

So the proposed polynomial smooth functions are more effective than their one, the smooth precision is higher one order.

The paper is organized as follows. In Sec. 2, we derive the polynomial smooth support vector machine. BFGS and Newton-Armijo algorithms are given in Sec. 3. Numerical tests and comparisons are given in Sec. 4. In Sec. 5, we make a conclusion of this paper.

2 The Polynomial Smooth Support Vector Machine (PSSVM)

The support vector machine is the unconstrained optimization problem

$$\min_{(\omega,\gamma)\in R^{n+1}} \frac{\nu}{2}\|(e-(D(A\omega - e\gamma)))_{+}\|_2^2 + \frac{1}{2}(\|\omega\|_2^2 + \gamma^2). \tag{1}$$

This is a strongly convex minimization problem without any constraints and exists a unique solution. However, the objective function in (1) is not differentiable at zero. We introduce two polynomial functions with parameter k to smooth the objective function.

The approximation functions have the following formulation:

$$q(x,k) = \begin{cases} x, & \text{if } x > \frac{1}{k}, \\ \frac{k}{4}x^2 + \frac{1}{2}x + \frac{1}{4k}, & \text{if } -\frac{1}{k} \leq x \leq \frac{1}{k}, \\ 0, & \text{if } x < -\frac{1}{k}. \end{cases} \tag{2}$$

$$h(x,k) = \begin{cases} x, & \text{if } x > \frac{1}{k}, \\ -\frac{k^3}{16}(x+\frac{1}{k})^3(x-\frac{3}{k}), & \text{if } -\frac{1}{k} \leq x \leq \frac{1}{k}, \\ 0, & \text{if } x < -\frac{1}{k}. \end{cases} \tag{3}$$

where $k \in Z^{+}$ is a positive integer.

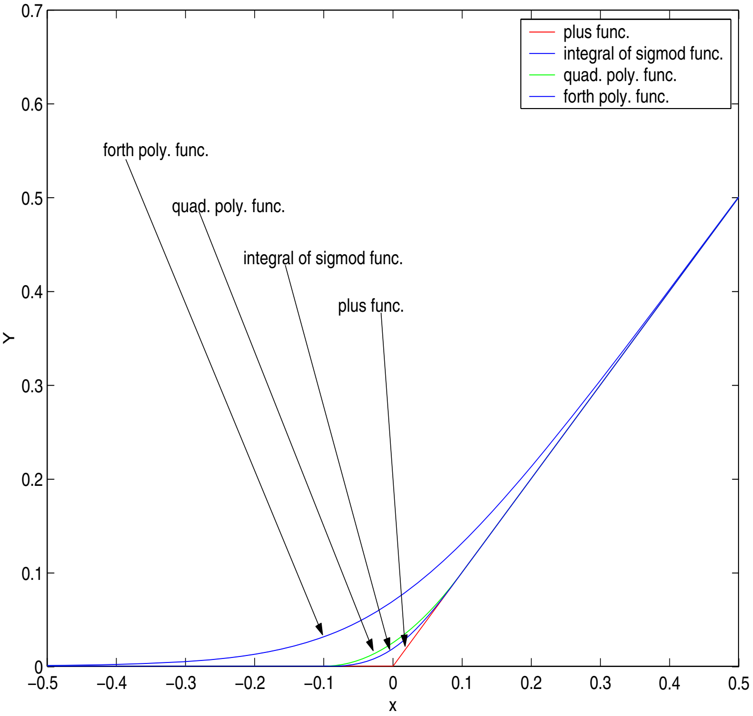


Fig. 1. Smooth performance compare with smoothing parameter k=10

The smooth performance of the polynomial functions can be seen in the figure 1.

Theorem 2.1. If the smooth functions have the formulation as (2) and (3), we have the following results:

i) $q(x, k)$ and $h(x, k)$ are continuous for any given $k \in Z^+$;

ii) $q(x, k)$ is first-order differentiable and $h(x, k)$ is twice-order differentiable for any given $k \in Z^+$.

Theorem 2.2. For $x \in R$, $a(x, k) \geq x_+$. For the quadratic polynomial smooth function,

$$q(x, k)^2 - x_+^2 \leq \frac{1}{11k^2}; \tag{4}$$

For the forth polynomial smooth function,

$$h(x, k)^2 - x_+^2 \leq \frac{1}{19k^2}. \tag{5}$$

When the polynomial functions $q(x, k)$ and $h(x, k)$ are applied to the object function in problem (1), we get the polynomial smooth support vector machine (PSSVM):

$$\min_{(\omega,\gamma)\in R^{n+1}} \varphi(\omega,\gamma) = \frac{\nu}{2}\|q(e-(D(A\omega - e\gamma)),k)\|_2^2 + \frac{1}{2}(\|\omega\|_2^2+\gamma^2). \tag{6}$$

$$\min_{(\omega,\gamma)\in R^{n+1}} \varphi(\omega,\gamma) = \frac{\nu}{2}\|h(e-(D(A\omega - e\gamma)),k)\|_2^2 + \frac{1}{2}(\|\omega\|_2^2+\gamma^2). \tag{7}$$

3 The BFGS and Newton-Armijo Methods

BFGS methods are suitable for unconstrained optimization with function and gradient value evaluation available, and it is well known that the BFGS method is the most widely used one among various quasi-Newton methods.

The BFGS method employed in the solution of problem PSSVM has the following form.

Algorithm 3.1. (The BFGS algorithm for PSSVM).

Step 1: Given the control factor of algorithm accuracy ϵ and the initial smooth parameter k_0. According (4)(5), computing the correspond lowest bound k^* and let $K = int[\frac{ln(k^*)}{ln(k_0)}]$(int[] is the integral function).

Step 2: Given $H^0 = I, (\omega^0, \gamma^0) = p^0 \in R^{n+1}$, $\varepsilon = 10^{-8}$ and set $i := 0, j := 0$;

Step 3: If $j \leq K$, set $k_j = (k_0)^j, \varepsilon_j = \frac{\varepsilon}{10^{K-j}}$;

Step 4: Evaluate $\varphi^i = \varphi(p^i, k_j)$ and $g^i = \nabla\varphi(p^i, k_j)$;

Step 5: If $\|g^i\|_2^2 \leq \varepsilon_j$, then stop, and accept $p^i = (\omega^i, \gamma^i)$ as the optimal solution of PSSVM, else calculate $d^i = -H^i g^i$;

Step 6: Line search along direction d^i to get a step length $\alpha^i > 0$; Let

$$p^{i+1} = p^i + \alpha^i d^i,\ s^i = p^{i+1} - p^i = -\alpha^i H^i g^i,$$

and evaluate $\varphi^{i+1}(k_j) = \varphi(p^{i+1}, k_j)$, $g^{i+1} = \nabla\varphi(p^{i+1}, k_j)$ and $y^i = g^{i+1} - g^i$;

Step 7: Update H^i to get H^{i+1}:

$$H^{i+1} = (I - \frac{s^i(y^i)^T}{(s^i)^T y^i})H^i(I - \frac{y^i(s^i)^T}{(s^i)^T y^i}) + \frac{s^i(s^i)^T}{(s^i)^T y^i}; \tag{8}$$

Step 8: Set $i := i + 1$, go to step 5;

Step 9: Set $j := j + 1$, go to step 3.

The Newton-Armijo method is a fast solution method for optimal problems. We will use it in the forth polynomial smooth model.

The Newton-Armijo method employed in the solution of problem (1) has the following form.

Algorithm 3.2. (The Newton-Armijo algorithm for PSSVM).

Step 1: Given the control factor of algorithm accuracy ϵ and the initial smooth parameter k_0. According (4)(5), computing the correspond lowest bound k^* and let $K = int[\frac{ln(k^*)}{ln(k_0)}]$(int[] is the integral function).

Step 2: Given $(\omega^0, \gamma^0) = p^0 \in R^{n+1}$, $\varepsilon = 10^{-8}$ and set $i := 0, j := 0$;

Step 3: If $j \leq K$, set $k_j = (k_0)^j, \varepsilon_j = \frac{\varepsilon}{10^{K-j}}$;

Step 4: Evaluate $\varphi^i(k_j) = \varphi(p^i, k_j)$ and $g^i = \nabla\varphi(p^i, k_j)$;

Step 5: If $\|g^i\|_2^2 \leq \varepsilon_j$, then stop, and accept $p^i = (\omega^i, \gamma^i)$ as the optimal solution of PSSVM, else calculate Newton directiond^i from the system of equations

$$\bigtriangledown^2\varphi(p^i, k_j)d^i = -g^i;$$

Step 6: Line search along direction d^i with Armijo step to get a step length $\alpha^i > 0$; Let

$$p^{i+1} = p^i + \alpha^i d^i;$$

Step 7: Set $i := i + 1$, go to step 5;

Step 8: Set $i := i + 1$, go to step 3.

Now, we analyze the convergence of Algorithm 3.1 and Algorithm 3.2.

1) If k_j is given (inter-cycle), Algorithm 3.1 and 3.2 are commonly BFGS and Newton-armijo algorithms. The convergent theorems of them can be seen in ([4]-[10]).

2) Let $\{k_0, k_1, k_2, \cdots, k_K\}$ be the increasing sequence of smooth parameter k (outer-cycle), $\{p^*(k_0), p^*(k_1), p^*(k_2), \cdots, p^*(k_K)\}$ is the sequence of the optimal solutions generated with the increasing k. Since $\lim\limits_{k\to\infty} x(k)^* = x^*$, the sequence $\{x^*(k_0), x^*(k_1), x^*(k_2), \cdots, x^*(k_K)\}$ is convergent to the optimal solution x^*.

4 Experiment Results and Analysis

This section presents the numerical experiment results.

To evaluate the performance of the proposed the BFGS algorithm and Newton-Armijo algorithm, we perform extensive simulation experiments and study the margin, testing correctness (%), objective function minimize values of PSSVM and the iteration numbers to get the super separating plane $P = \{x|x \in R^n, x^T\omega = \gamma\}$.

The experiments are implemented on 6 randomly generated databases with normal distribution and on a PC with 1.8G MHz Pentium IV and 256 MB SDRAM using MATLAB 6.1.

In the following table, m is the number of sample points for classification, tn is the number of test points applied for SVM, n is the number of features including in database or the dimension of data points. In experiments, select $k_0 = 5, \nu = 1, \epsilon = 10^{-5}, \varepsilon = 10^{-8}$. QPSSVM is the quadratic polynomial smooth support vector machine. FPSSVM is the forth polynomial smooth support vector machine. SSVM is the smooth support vector machine in [2].

Table 1. Experimental results

Datasetsize			Margin Test Correctness(%) Iteration number The minimize value of objective function				
m	n	tn	**BFGS method**			**Newton-Armijo method**	
			QPSSVM	FPSSVM	SSVM	FPSSVM	SSVM
100	20	1000	25.6777	26.0973	21.5929	26.0979	21.5932
			92.50	92.50	92.30	92.30	92.30
			233	262	222	22	15
			0.0031	0.0030	0.0046	0.0030	0.0046
500	20	1000	6.0142	6.0802	5.4927	6.0801	5.4927
			98.70	98.70	98.90	98.70	98.90
			187	206	211	21	16
			0.0582	0.0567	0.0739	0.0567	0.0739
1000	20	1000	2.6004	2.5990	2.3579	2.5990	2.4662
			99.20	99.20	99.00	99.10	99.00
			230	227	242	26	17
			0.3383	0.3361	0.3825	0.3361	0.4142
100	100	1000	91.0093	92.1096	68.4472	92.1508	68.4543
			73.80	74.30	74.30	74.30	74.20
			510	701	381	24	18
			$2.4256e^{-4}$	$2.3744e^{-4}$	$4.5679e^{-4}$	$2.3722e^{-4}$	$4.5656e^{-4}$
500	100	1000	24.0068	25.1347	19.3556	24.3772	19.3565
			91.90	92.40	92.10	91.70	92.10
			1056	1027	732	49	17
			0.0035	0.0034	0.0057	0.0034	0.0057
1000	100	1000	12.7751	12.9832	10.7127	12.9833	10.7128
			94.90	94.90	95	94.90	95
			1014	919	744	44	17
			0.0124	0.0121	0.0188	0.0121	0.0188

Experiment results are presented in Table 1.

From observing the results in Table 1, the margins obtained by the BFGS method with QPSSVM and FPSSVM are larger than SSVM. This situation can not change with increasing of data scale. The following figure 2 can show it. In another hand, the minimize values of objective function obtained by the BFGS method with QPSSVM and FPSSVM are less than SSVM. This situation also can not change with increasing of data scale. The following figure 3 can show it. The situation of Newton-Armijo method is similar with this.

The iteration numbers under Newton-Armijo algorithm with same smooth function are less than BFGS method very much. There is little difference under same method. Test Correctness is similar under different smooth function with same database.

In [2], Yuh-Jye Lee and O. L. Mangarasian has compared the results of SSVM with RLP directly and SOR [11], SMO [11] and SVM^{light} and presented

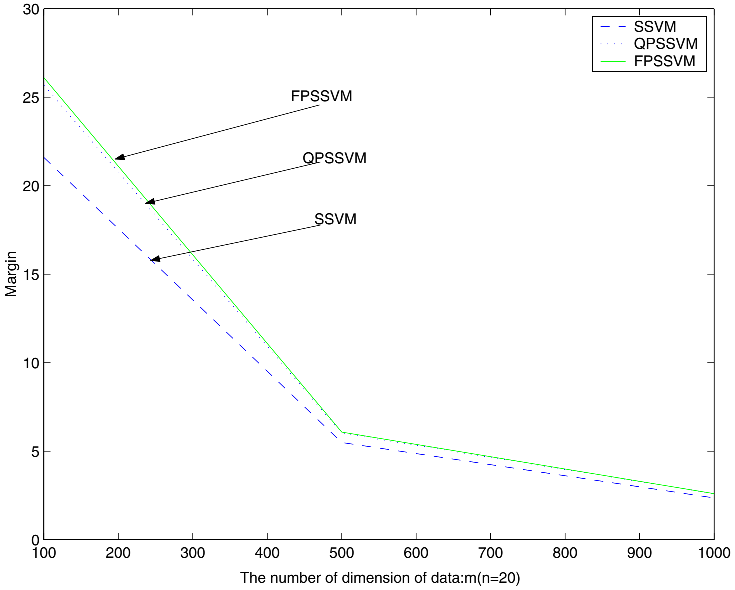


Fig. 2. The magins situation with increasing of data scale under the BFGS method

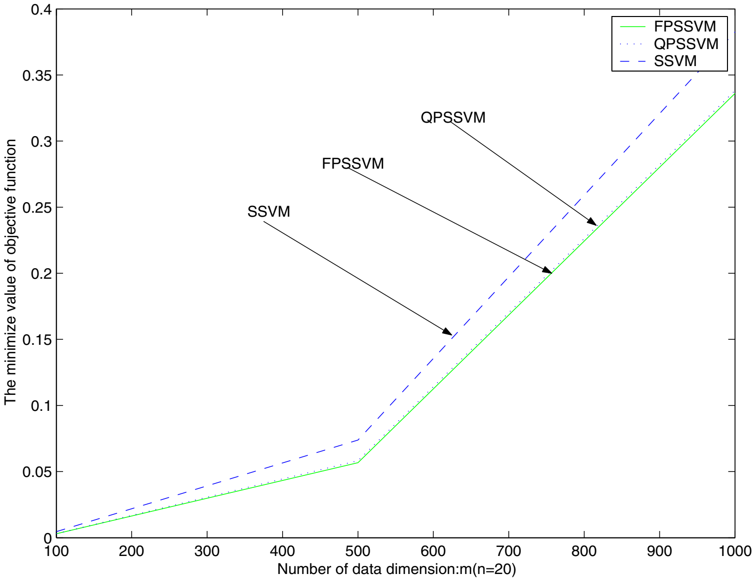


Fig. 3. The minimize value of objective function with different data dimension

that SSVM is more effective than RLP directly and SOR [11], SMO [11] and SVM^{light}.

In practice, the FPSSVM and Newton-Armijo method are the first selection for application of support vector machine for classification.

5 Conclusion

This paper presents a new polynomial smooth for the problem of support vector machine. The smooth function is no difference with x_+ at intervals $(-\infty, -\frac{1}{k}]$ and $[\frac{1}{k}, \infty)$. The BFGS and Newton-Armijo methods are given in order to solve the PSSVM and get better classifier from PSSVM than SSVM. It is more effective than the prevenient methods for solving the SVM. How to get a higher order polynomial smooth function is a difficult problem. The authors would believe that there must be a higher order polynomial that can be used to get a good smooth support vector machine.

Acknowledgment

The authors would like to thank the editor Xue Li for his carefully reading the manuscript which improved the English presentation and also thank the referees for the valuable comments.

References

1. V. Vapnik, The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995.
2. Yuh-Jye Lee and O.L. Mangarasian, SSVM: A Smooth Support Vector Machine for Classification. *Computational Optimization and Applications*, 22(2001):5-21.
3. C.J.C.Burges, A tutorial on support vector machines for pattern recognition, *Data Mining and Knowledge Discovery*, 2(1998): 121-167.
4. C. X. Xu and J. Z. Zhang, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, Technical Report, Department of Mathematics, City University of Hong Kong, 1999.
5. C.X. Xu and J.Z. Zhang, A syuvey of quasi-Newton equations and quasi-Newton methods for optimization, *Annals of Operations Research*, 103(2001): 213-234.
6. J. Z. Zhang and C. X. Xu, Properties and numerical performance of quasi-Newton methods with modified quasi-Newton equations, *J. Comp. Appl. Math.*, 137(2001):269-278.
7. Y. Yuan and R. Byrd, Non-quasi-Newton updates for unconstrained optimization, *J. Comp. Math.*, 13(1995):95-107.
8. Y. Yuan and W. Y. Sun, *Optimal Method and Technology*, Chinese Science Publisher, 2000.
9. Y. Yuan, A modified BFGS algorithm for unconstrained optimization, *IMA J. Numer. Anal*, 11 (1991):325-332.
10. J.Z. Zhang, N.Y. Deng and L.H. Chen, New quasi-Newton equation and related methods for unconstrained optimization, *J. Optim. Theory Appl*, 102 (1999):147-167.
11. O. L. Mangasarian and David R. Musicant, Successive overrelaxation for support vector machines, *IEEE Transactions on Neural Networks*, 10(1999):1032-1037. ftp://ftp.cs.wisc.edu/math-prog/techreports/98-18.ps.

Reducts in Incomplete Decision Tables

Renpu Li[1, 2] and Dao Huang[2]

[1] College of Computer Science and Technology, Yantai Normal University,
Yantai 264025, China
lip0109@sohu.com

[2] College of Information, East China University of Science and Technology,
Shanghai 200237, China
dhuang@ecust.edu.cn

Abstract. Knowledge reduction is an important issue in data mining. This paper focuses on the problem of knowledge reduction in incomplete decision tables. Based on a concept of incomplete conditional entropy, a new reduct definition is presented for incomplete decision tables and its properties are analyzed. Compared with several existing reduct definitions, the new definition has a better explanation for knowledge uncertainty and is more convenient for application of the idea of approximate reduct in incomplete decision tables.

1 Introduction

Rough set theory [1, 2], developed by professor Pawlak, has been conceived as a valid mathematical theory to deal with inexact, uncertain or vague knowledge in many applicants such as data mining, machine learning and decision support.

One of the main tasks of the rough set theory is knowledge reduction, in which reduct is a core concept. A reduct is a minimal subset of the attributes that provides the same information for classification purposes as the whole set of attributes. The idea of reduct has been proved to be very effective in knowledge reduction and has received a great deal of research [3, 4]. But the research about reduct for incomplete information systems, especially for incomplete decision tables, is very scarce.

Using the concept of generalized decision, Kryszkiewicz firstly presents a reduct definition for the knowledge reduction in incomplete decision tables [5]. Several other definitions including distribution reduct, maximum distribution reduct, assignment reduct and assignment order reduction are proposed in [6], and some relationships among them are also discussed.

In this paper, we present a more informative reduct definition for incomplete decision tables by combining rough set theory and entropy. This new definition is based on the concept of incomplete conditional entropy, which is an extension of conditional entropy in incomplete decision tables. Compared with previous definitions, the new definition provides a mathematical quantitative measure of knowledge uncertainty in incomplete decision tables and is more convenient for application of the idea of approximate reduct [4] in incomplete decision tables.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 165 – 174, 2005.

2 Basic Concepts

Definition 1. An *information system* is a pair $S=(U,A)$ where U is a non-empty finite set of *objects* and A is a non-empty finite set of *attributes*, such that *a*: $U \to V_a$ for any $a \in A$, where V_a is called the *value set* of *a*.

If some attribute values of objects in an information system are missing, these values are called missing values (or null values), which will be denoted by symbol "*" in this paper. If an information system contains at least one missing value, it is called an *incomplete information system*, otherwise it is *complete*.

Definition 2. *A decision table is a special information system* $T=(U, C \cup \{d\})$*, where* d*,* $d \notin C$ *and* $* \notin V_d$*, is a distinguished attribute called decision, and the elements of* C *are called condition attributes.*

Definition 3. Let $S=(U,A)$ be a complete information system. Each subset of attributes $P \subseteq A$ determines a binary *indiscernibility relation* $IND(P)$ on U :

$$IND(P)=\{(x,y)\in U\times U \mid \forall a\in P, f_a(x)=f_a(y)\}.$$

The relation $IND(P)$ is an equivalence relation on U so it constructs a partition of U , which will be denoted by $U/IND(P)$. Let $U/IND(P)=\{X_1, X_2,..., X_n\}$, X_i is called an *equivalence class*. Let $I_P(x)=\{y\in U \mid (x,y)\in IND(P)\}$. $I_P(x)$ is the maximal set of objects which are indiscernible by P with x .

Theorem 1. Let $S=(U,A)$ be a complete information system, $P \subseteq A$ and $U/IND(P)=\{X_1, X_2,..., X_n\}$. If $x \in X_i$ ($i=1,2,...,n$), then $I_P(x)=X_i$.

Definition 4. *Let* $S=(U,A)$ *be an incomplete information system. Each subset of attributes* $P \subseteq A$ *determines a binary similarity relation* $SIM(P)$ *on* U *:*

$$SIM(P)=\{(x,y)\in U\times U \mid \forall a\in P, f_a(x)=f_a(y) \quad or \quad f_a(x)=* \quad or \quad f_a(y)=*\}.$$

Let $S_P(x)=\{y\in U \mid (x,y)\in SIM(P)\}$. $S_P(x)$ is the maximal set of objects which are possibly indiscernible by P with x . The relation $SIM(P)$ is a tolerance relation on U . Let $U/SIM(P)=\{S_P(x) \mid x\in U\}$. Any element from the family set $U/SIM(P)$ is called a *tolerance class*. Tolerance classes in $U/SIM(P)$ in general do not constitute a partition of U . In fact they constitute a covering of U .

Theorem 2. Let $S=(U,A)$ be an information system, $P \subseteq A$. If S is complete, then $I_P(x)=S_P(x)$ for $\forall x \in U$.

Theorem 3. Let $S=(U,A)$ be an information system. If $P \subseteq Q \subseteq A$, then $S_Q(x) \subseteq S_P(x)$ for $\forall x \in U$.

Definition 5. *Let* $S=(U,A)$ *be a complete information system,* $P\subseteq A$ *and* $U/IND(P)=\{X_1,X_2,...,X_n\}$. *The entropy of knowledge* P *is defined by*

$$E(P)=-\sum_{i=1}^{n}\frac{|X_i|}{|U|}\log\frac{|X_i|}{|U|},$$

where |*X*| is the cardinality of set *X* and $\log x$ denotes $\log_2 x$.

Definition 6. Let $T=(U,C\cup\{d\})$ be a complete decision table, $P\subseteq C$, $U/IND(P)=\{X_1,X_2,...,X_n\}$ and $U/IND(\{d\})=\{Y_1,Y_2,...,Y_m\}$. The *conditional entropy* of decision d with respect to knowledge P is defined by

$$E(d\,|\,P)=-\sum_{i=1}^{n}\frac{|X_i|}{|U|}\sum_{j=1}^{m}\frac{|X_i\cap Y_j|}{|X_i|}\log\frac{|X_i\cap Y_j|}{|X_i|}.$$

3 Reduct Based on Incomplete Conditional Entropy

Entropy is a useful metric for quantifying information content. The research of combining rough set theory and entropy for knowledge representation and knowledge reduction of information systems has received a great deal of attentions and produced some important results [7, 8, 9]. In this section, the concept of conditional entropy is firstly extended in incomplete decision tables and then based on the extended concept a new reduct definition is presented. Relative properties are also examined.

In a complete system knowledge is regarded as a partition of the universe produced by the indiscernibility relation and an elementary knowledge unit is an equivalence class of the partition, while in an incomplete system knowledge corresponds to a covering of the universe based on the similarity relation and accordingly an elementary knowledge unit is an tolerance class of the covering. Based on this thought we present for incomplete decision tables a new concept - *incomplete conditional entropy*, which is proved to be an extension of the concept of conditional entropy in incomplete decision tables.

Definition 7. Let $S=(U,C\cup\{d\})$ be an incomplete decision table, $P\subseteq C$, $U=\{u_1,u_2,...,u_{|U|}\}$, and $U/IND(\{d\})=\{Y_1,Y_2,...,Y_m\}$.

We define the *incomplete conditional entropy* of u_i by

$$IE_{u_i}(d\,|\,P)=-\frac{1}{|U|}\sum_{j=1}^{m}\frac{|S_P(u_i)\cap Y_j|}{|S_P(u_i)|}\log\frac{|S_P(u_i)\cap Y_j|}{|S_P(u_i)|}.$$

The *incomplete conditional entropy* of decision d with respect to knowledge P is defined as follows:

$$IE(d\,|\,P)=\sum_{i=1}^{|U|}IE_{u_i}(d\,|\,P).$$

$IE_{u_i}(d\,|\,P)$ can be interpreted as the amount of uncertainty of u_i concerning *d* under the information about P based on similarity relation and $IE(d\,|\,P)$ is the sum of uncertainty amount of all objects.

Theorem 4. Let $T=(U,C\cup\{d\})$ be a decision table, $P\subseteq C$. If T is complete, then $IE(d\mid P)=E(d\mid P)$.

Proof. Let $U=\{u_1,u_2,...,u_{|U|}\}$. Since T is complete, assume that $U/IND(P)=\{X_1,X_2,...,X_n\}$ and $U/IND(\{d\})=\{Y_1,Y_2,...,Y_m\}$. From Theorem 2, we have that $I_P(u_k)=S_P(u_k)$ for $k=1,2,...,|u|$. From Theorem 1, it follows that if $u_k\in X_i$, then $I_P(u_k)=X_i$ for $k=1,2,...,|u|, i=1,2,...,n$. Therefore

$$\begin{aligned}
IE(d\mid P) &= -\sum_{k=1}^{|U|}\frac{1}{|U|}\sum_{j=1}^{m}\frac{|S_P(u_k)\cap Y_j|}{|S_P(u_k)|}\log\frac{|S_P(u_k)\cap Y_j|}{|S_P(u_k)|}\\
&= -\sum_{k=1}^{|U|}\frac{1}{|U|}\sum_{j=1}^{m}\frac{|I_P(u_k)\cap Y_j|}{|I_P(u_k)|}\log\frac{|I_P(u_k)\cap Y_j|}{|I_P(u_k)|}\\
&= -\sum_{u_k\in X_1}\frac{1}{|U|}\sum_{j=1}^{m}\frac{|X_1\cap Y_j|}{|X_1|}\log\frac{|X_1\cap Y_j|}{|X_1|}-\sum_{u_k\in X_2}\frac{1}{|U|}\sum_{j=1}^{m}\frac{|X_2\cap Y_j|}{|X_2|}\log\frac{|X_2\cap Y_j|}{|X_2|}\\
&\quad -...-\sum_{u_k\in X_n}\frac{1}{|U|}\sum_{j=1}^{m}\frac{|X_n\cap Y_j|}{|X_n|}\log\frac{|X_n\cap Y_j|}{|X_n|}\\
&= -\sum_{i=1}^{n}\frac{|X_i|}{|U|}\sum_{j=1}^{m}\frac{|X_i\cap Y_j|}{|X_i|}\log\frac{|X_i\cap Y_j|}{|X_i|}=E(d\mid P).
\end{aligned}$$

It can be concluded from Theorem 4 that incomplete conditional entropy is an extension of conditional entropy in incomplete decision tables.

Property 1. Let $T=(U,C\cup\{d\})$ be a decision table. The relation "if $P\subseteq Q\subseteq A$, then $IE(d\mid P)\geq IE(d\mid Q)$" does not always hold when T is incomplete (the relation holds when T is complete, which is proved in [9, p.119]).
The following example is an illustration of Property 1.

Example 1. An incomplete decision table IDT1 is described in Table 1, where $U=\{1,2,3,4,5\}$, $C=\{a,b,c\}$ and d is the decision.

Table 1. IDT1

Objects	*a*	*b*	*c*	*d*
1	1	1	0	1
2	1	1	0	1
3	1	*	0	0
4	*	2	2	1
5	2	2	*	0

It can be obtained easily that $U/IND(\{d\})=\{Y_1,Y_2\}$, $Y_1=\{1,2,4\}$, $Y_2=\{3,5\}$, and $U/SIM(C)=\{S_C(1),S_C(2),S_C(3),S_C(4),S_C(5)\}$, $S_C(1)=S_C(2)=S_C(3)=\{1,2,3\}$ and $S_C(4)$ $=S_C(5)=\{4,5\}$.

Let $P=\{a,b\}$, we have $U/SIM(P)=\{S_P(1),S_P(2),S_P(3),S_P(4),S_P(5)\}$, where $S_P(1)$ $=S_P(2)=\{1,2,3\}$, $S_P(3)=\{1,2,3,4\}$, $S_P(4)=\{3,4,5\}$, $S_P(5)=\{4,5\}$. Therefore,

$$\begin{aligned}IE(d\mid C)&=-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})\\&-\frac{1}{5}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})-\frac{1}{5}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})=0.9510,\\IE(d\mid P)&=-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})-\frac{1}{5}(\frac{3}{4}\log\frac{3}{4}+\frac{1}{4}\log\frac{1}{4})\\&-\frac{1}{5}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})-\frac{1}{5}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})=0.9132.\end{aligned}$$

Hence we have $P\subseteq C$ and $IE(d\mid P)<IE(d\mid C)$

Property 1 shows that in incomplete decision tables incomplete conditional entropy of knowledge does not decrease monotonously with the increase of attributes in knowledge, although it does in complete decision tables.

Definition 8. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $U=\{u_1,u_2,...,u_{|U|}\}$. A set $P\subseteq C$ is called an incomplete entropy reduct of T iff $IE_{u_i}(d\mid P)\le IE_{u_i}(d\mid C)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $IE_{u_j}(d\mid P')>IE_{u_j}(d\mid C)$.

Theorem 5. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $P\subseteq C$. If P is an incomplete entropy reduct of T, then $IE(d\mid P)\le IE(d\mid C)$.

Proof. Suppose $U=\{u_1,u_2,...,u_{|U|}\}$. Since $P\subseteq C$ is a reduct of T, we have $IE_{u_i}(d\mid P)\le IE_{u_i}(d\mid C)$ for $\forall u_i\in U$. According to the Definition 7, we have

$$IE(d\mid P)=\sum_{i=1}^{|U|}IE_{u_i}(d\mid P)\le\sum_{i=1}^{|U|}IE_{u_i}(d\mid C)=IE(d\mid C).$$

4 Comparison with Other Reducts

4.1 Other Reduct Definitions

Definition 9 [5]. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $P\subseteq C$ and $U=\{u_1,u_2,...,u_{|U|}\}$. Let $\partial_P(u_i)=\{t\mid t=d(y)\ \ and\ \ y\in S_P(u_i)\}$ is called a *generalized decision* of u_i. P is called a *generalized decision reduct* of T iff $\partial_P(u_i)=\partial_C(u_i)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $\partial_{P'}(u_j)\ne\partial_C(u_j)$.

Definition 10 [6]. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $P\subseteq C$, $U=\{u_1,u_2,...,u_{|U|}\}$ and $U/IND(\{d\})=\{Y_1,Y_2,...,Y_m\}$.

For $\forall u_i\in U$, Let $\mu_P(u_i)=(Y_1^P(u_i),Y_2^P(u_i),...,Y_m^P(u_i))$, where $Y_j^P(u_i)=\dfrac{|Y_j\cap S_P(u_i)|}{|S_P(u_i)|}$, $j=1,2,...,m$.

Let

$$\gamma_P(u_i)=\{Y_h : Y_h^P(u_i)=\max_{1\le j\le m} Y_j^P(u_i)\},$$

$$\delta_P(u_i)=\{Y_j : Y_j \cap S_P(u_i)\ne \Phi\},$$

$$\rho_P(u_i)=\{Y_{j_1}\ge Y_{j_2}\ge ...\ge Y_{j_k} : \sum_{l=1}^{k} Y_{j_l}(u_i)=1, Y_{j_l}(u_i)>0\}, \text{ then}$$

(1) P is called a distribution set of T if $\mu_P(u_i)=\mu_C(u_i)$ for $i=1,2,...,|U|$. P is called a distribution reduct of T iff $\mu_P(u_i)=\mu_C(u_i)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $\mu_P(u_j)\ne\mu_C(u_j)$.
(2) P is called a maximum distribution set of T if $\gamma_P(u_i)=\gamma_C(u_i)$ for $i=1,2,...,|U|$. P is called a maximum distribution reduct of T iff $\gamma_P(u_i)=\gamma_C(u_i)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $\gamma_P(u_j)\ne\gamma_C(u_j)$.
(3) P is called an assignment set of T if $\delta_P(u_i)=\delta_C(u_i)$ for $i=1,2,...,|U|$. P is called an assignment reduct of T iff $\delta_P(u_i)=\delta_C(u_i)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $\delta_P(u_j)\ne\delta_C(u_j)$.
(4) P is called an assignment order set of T iff $\rho_P(u_i)=\rho_C(u_i)$ for $i=1,2,...,|U|$. P is called an assignment order reduct of T iff $\rho_P(u_i)=\rho_C(u_i)$ for $i=1,2,...,|U|$ and for $\forall P'\subset P$, exist $u_j\in U$ such that $\rho_P(u_j)\ne\rho_C(u_j)$.

Theorem 6. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $P\subseteq C$. P is a generalized decision reduct of T iff P is an assignment reduct of T.

The proof is obvious from the definitions of the two kinds of reducts. So the assignment reduct definition is equivalent to the generalized decision reduct definition. For simplification, assignment reduct is only used in the following content.

4.2 Relationships Among These Reducts

According to the classification in [8], the reduct definition based on incomplete conditional entropy is presented from the information view of rough set theory, and the other definitions in [6] can be categorized to the algebra view of rough set theory. The relationships among these reducts are discussed in this subsection respectively in consistent incomplete decision tables and in inconsistent incomplete decision tables.

Definition 11. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $U=\{u_1,u_2,...,u_{|U|}\}$. If $|\partial_C(u_i)|=1$ for $i=1,2,...,|U|$, T is consistent, otherwise it is inconsistent.

Theorem 7. Let $T=(U,C\cup\{d\})$ be an incomplete decision table, $U=\{u_1,u_2,...,u_{|U|}\}$ and $P\subseteq C$. If T is consistent, then P is an incomplete entropy reduct of T iff P is a distribution reduct of T.

Proof. Assume $U/IND(\{d\})=\{Y_1,Y_2,...,Y_m\}$. Since T is consistent, then $|\partial_C(u_i)|=1$ for $i=1,2,...,|U|$. That is to say, for each $u_i\in U$, all objects in $S_C(u_i)$ have the same decision value, that is $d(u_i)$. Let $u_i\in Y_j$, this means that Y_j contains all objects of U whose decision value is $d(u_i)$. Therefore, we have $S_C(u_i)\subseteq Y_j$ and for $1\le k\le m$

$$\begin{cases} S_C(u_i) \cap Y_k = S_C(u_i) & k = j, \\ S_C(u_i) \cap Y_k = \Phi & k \neq j. \end{cases}$$

Hence

$$IE_{u_i}(d \mid C) = -\frac{1}{|U|}\left(\frac{|S_c(u_i) \cap Y_j|}{|S_c(u_i)|}\log\frac{|S_c(u_i) \cap Y_j|}{|S_c(u_i)|}\right) = -\frac{1}{|U|}\left(\frac{|S_c(u_i)|}{|S_c(u_i)|}\log\frac{|S_c(u_i)|}{|S_c(u_i)|}\right) = 0\,.$$

$\mu_C(u_i) = (Y_1^C(u_i), Y_2^C(u_i), ..., Y_m^C(u_i))$, and for $1 \leq k \leq m$

$$\begin{cases} Y_k^C(u_i) = 1 & k = j, \\ Y_k^C(u_i) = 0 & k \neq j. \end{cases}$$

It can be seen that in a consistent incomplete decision table, for each $u_i \in U$, its incomplete decision entropy equals 0 while only one element of $\mu_C(u_i)$ equals 1 and the other elements equal 0.

Firstly, suppose that P is an incomplete entropy reduct, we prove that P is also a distribution reduct. Since P is an incomplete entropy reduct, we have that $IE_{u_i}(d \mid P) \leq IE_{u_i}(d \mid C)$ for each $u_i \in U$. It is known that $IE_{u_i}(d \mid C) = 0$ and according to Definition 7 $IE_{u_i}(d \mid P) \geq 0$. So we have $IE_{u_i}(d \mid P) = 0$ for each $u_i \in U$. This means that all objects in $S_P(u_i)$ have the same decision value, that is $d(u_i)$. Let $u_i \in Y_j$, we also have $S_P(u_i) \subseteq Y_j$ and for $1 \leq k \leq m$

$$\begin{cases} S_P(u_i) \cap Y_k = S_C(u_i) & k = j, \\ S_P(u_i) \cap Y_k = \Phi & k \neq j. \end{cases}$$

It follows $\mu_P(u_i) = (Y_1^P(u_i), Y_2^P(u_i), ..., Y_m^P(u_i))$, where for $1 \leq k \leq m$

$$\begin{cases} Y_k^P(u_i) = 1 & k = j, \\ Y_k^P(u_i) = 0 & k \neq j. \end{cases}$$

So it is obtained that $\mu_P(u_i) = \mu_C(u_i)$ for $i = 1,2,...,|U|$. In addition, since P is an incomplete entropy reduct, for $\forall P' \subset P$, we can find at least one object $u_i \in U$ such that $IE_{u_i}(d \mid P') > IE_{u_i}(d \mid C)$, hence $IE_{u_i}(d \mid P') > 0$. This means that the objects in $S_{P'}(u_i)$ have at least two different decision values. From the definition of $\mu_{P'}(u_i)$ it is easily obtained that at least two elements of $\mu_{P'}(u_i)$ are larger than 0, it follows $\mu_{P'}(u_i) \neq \mu_C(u_i)$. According to the discussion above we have that P is a distribution reduct of T .

Secondly, suppose that P is a distribution reduct, we prove that P is also an incomplete entropy reduct. Since P is a distribution reduct, we have that $\mu_P(u_i) = \mu_C(u_i)$ for each $u_i \in U$. It follows that there is only one element of $\mu_P(u_i)$ equals 1 and the other elements equal 0. This means that all objects in $S_P(u_i)$ have the same decision value, that is $d(u_i)$. From the Definition 7 we have $IE_{u_i}(d \mid P) = 0$ for each $u_i \in U$. So it is obtained that $IE_{u_i}(d \mid P) = IE_{u_i}(d \mid C) = 0$ for each $u_i \in U$. In addition, since P is a distribution reduct, for $\forall P' \subset P$, we can find at least one object u_i such that $\mu_{P'}(u_i) \neq \mu_C(u_i)$. From Theorem 3, we know $S_{P'}(u_i) \subseteq S_C(u_i)$, so there is only one case that the objects in $S_{P'}(u_i)$ have at least two different decision values, it is easily induced that $IE_{u_i}(d \mid P') > 0$. It follows that $IE_{u_i}(d \mid P') > IE_{u_i}(d \mid C)$. According to the discussion above we have that P is an incomplete entropy reduct of T .

The proof is completed.

It can be concluded from Theorem 7 that the incomplete entropy reduct definition and the distribution reduct definition are equivalent in consistent incomplete decision tables.

In fact in a similar way we can prove that the incomplete entropy reduct definition is also equivalent to the definitions of assignment reduct and assignment order reduct.

However, in inconsistent decision tables incomplete entropy reduct definition is not equivalent to any one of those definitions presented in [6]. This inequivalence between incomplete entropy reduct definition and assignment reduct definition is illustrated with the following examples.

Example 2. An incomplete decision table IDT2 is described in Table 2, where $U=\{1,2,...,12\}$, $C=\{a,b,c\}$ and d is the decision.

Table 2. IDT2

objects	*a*	*b*	*c*	*d*
1	3	*	1	0
2	3	3	1	1
3	*	2	2	1
4	2	2	*	2
5	1	1	0	2
6-12	1	*	0	1

It can be obtained easily that $U/IND(d)=\{Y_1,Y_2,Y_3\}$, where $Y_1=\{1\}$, $Y_2=\{2,3,6-12\}$, $Y_3=\{4,5\}$.

Also we have $U/SIM(C)=\{S_C(1),S_C(2),...,S_C(12)\}$, where $S_C(1)=S_C(2)=\{1,2\}$, $S_C(3)=S_C(4)=\{3,4\}$, and $S_C(5)=S_C(6)=...=S_C(12)=\{5,6-12\}$. Therefore

$$IE_1(d\mid C)=IE_2(d\mid C)=IE_3(d\mid C)=IE_4(d\mid C)=-\frac{1}{12}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})=0.0833,$$

$$IE_5(d\mid C)=IE_6(d\mid C)=...=IE_{12}(d\mid C)=-\frac{1}{12}(\frac{7}{8}\log\frac{7}{8}+\frac{1}{8}\log\frac{1}{8})=0.0453,$$

$\delta_C(1)=\delta_C(2)=\{Y_1,Y_2\}$, and $\delta_C(3)=\delta_C(4)=...=\delta_C(12)=\{Y_2,Y_3\}$.

Let $P=\{a,b\}$, so $U/SIM(P)=\{S_P(1),S_P(2),...,S_P(12)\}$, where $S_P(1)=\{1,2,3\}$, $S_P(2)=\{1,2\}$, $S_P(3)=\{1,3,4,6-12\}$, $S_P(4)=\{3,4\}$, $S_P(5)=\{5,6-12\}$, $S_P(6)=S_P(7)=...$ $=S_P(12)=\{3,5,6-12\}$. It follows

$$IE_1(d\mid P)=-\frac{1}{12}(\frac{2}{3}\log\frac{2}{3}+\frac{1}{3}\log\frac{1}{3})=0.0765,$$

$$IE_2(d\mid P)=-\frac{1}{12}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})=0.0833,$$

$$IE_3(d\mid P)=-\frac{1}{12}(\frac{8}{10}\log\frac{8}{10}+\frac{1}{10}\log\frac{1}{10}+\frac{1}{10}\log\frac{1}{10})=0.0768,$$

$$IE_4(d\mid P)=-\frac{1}{12}(\frac{1}{2}\log\frac{1}{2}+\frac{1}{2}\log\frac{1}{2})=0.0833,$$

$$IE_5(d\mid P)=-\frac{1}{12}(\frac{7}{8}\log\frac{7}{8}+\frac{1}{8}\log\frac{1}{8})=0.0453,$$

$$IE_6(d\mid P)=IE_7(d\mid P)=...=IE_{12}(d\mid P)=-\frac{1}{12}(\frac{8}{9}\log\frac{8}{9}+\frac{1}{9}\log\frac{1}{9})=0.0419.$$

So we have $IE_i(d \mid P) \le IE_i(d \mid C)$ for $i = 1,2,...,12$. And since

$$IE_3(d \mid \{a\}) = -\frac{1}{12}(\frac{9}{12}\log\frac{9}{12} + \frac{2}{12}\log\frac{2}{12} + \frac{1}{12}\log\frac{1}{12}) = 0.0867 > IE_3(d \mid C) ,$$

$$IE_1(d \mid \{b\}) = -\frac{1}{12}(\frac{9}{12}\log\frac{9}{12} + \frac{2}{12}\log\frac{2}{12} + \frac{1}{12}\log\frac{1}{12}) = 0.0867 > IE_1(d \mid C) .$$

According to Definition 8 $P = \{a,b\}$ is an incomplete entropy reduct of IDT2. However, since $\sigma_P(3) = \{Y_1, Y_2, Y_3\} \ne \sigma_C(3)$, $P = \{a,b\}$ is not an assignment reduct of IDT2 according to Definition 10.

Example 3. For the incomplete decision table IDT3 described in Table 3, in a similar way, we can also prove that $P = \{a,b\}$ is an assignment reduct of IDT3 according to Definition 10, but it is not an incomplete entropy reduct of IDT3 according to Definition 8.

Table 3. IDT3

objects	*a*	*b*	*c*	*d*
1	1	1	0	1
2	1	1	0	0
3	1	*	0	0
4	*	2	2	1
5	2	2	*	0

Compared with those definitions presented from the algebra view of rough set theory in [6], the incomplete entropy reduct definition gives a mathematical quantitative measure for estimating the knowledge uncertainty of different attribute sets, hence it has a better explanation for the knowledge uncertainty and can be more easily used in knowledge reduction of incomplete decision tables.

4.3 Application of the Idea of Approximate Reduct

Conditions of preserving the degree of the model (in)consistency while reducing attributes turn out to be too rigorous with respect to possible noise and fluctuations in data [4]. A feasible solution for handling this problem is the idea of approximate reduct, in which the conditions of reduct definition should be weaken under the constraint of preserving the useful information at a reasonable high level.

The new reduct definition based on incomplete conditional entropy is more convenient than those reduct definitions proposed in [6] for the application of the idea of rough approximate reduct. For example, we can easily give two following approximate reduct definitions based on the incomplete entropy reduct.

Definition 12. Let $T = (U, C \cup \{d\})$ be an incomplete decision table, $U = \{u_1, u_2, ..., u_{|U|}\}$ and $\varepsilon \in [0,1]$. A set $P \subseteq C$ is called a local $\varepsilon - approximate$ incomplete entropy reduct of T iff $IE_{u_i}(d \mid P) \le (1+\varepsilon) IE_{u_i}(d \mid C)$ for $i = 1,2,...,|U|$ and for $\forall P' \subset P$, exist $u_j \in U$ such that $IE_{u_j}(d \mid P') > (1+\varepsilon) IE_{u_j}(d \mid C)$.

Definition 13. Let $T = (U, C \cup \{d\})$ be an incomplete decision table, $U = \{u_1, u_2, ..., u_{|U|}\}$ and $\varepsilon \in [0,1]$. A set $P \subseteq C$ is called a global $\varepsilon - approximate$ incomplete entropy reduct of T iff $IE(d \mid P) \le (1+\varepsilon)IE(d \mid C)$ and for $\forall P' \subset P$, $IE(d \mid P') > (1+\varepsilon)IE(d \mid C)$.

5 Conclusions

In this paper a new reduct definition called incomplete entropy reduct is proposed for knowledge reduction of incomplete decision tables. This new definition and other reduct definitions presented from the algebra view of rough set theory are compared and analyzed respectively in consistent and inconsistent decision tables, some equivalence relations and different properties are obtained. It is also showed that incomplete entropy reduct is more convenient for application of the idea of approximate reduct in incomplete decision tables. How to construct efficient algorithms for knowledge reduction in incomplete decision tables based on incomplete entropy reduct is our future work.

References

1. Pawlak, Z.: Rough Set. Int. J. of Computer and Information Sciences 11(1982) 341-356
2. Pawlak, Z., Grzymala-Busse, J.W., Slowinski, R., Ziarko, W.: Rough Sets. Communications on the ACM 38(1995) 89-95
3. Susmaga, R.: Reducts and Constructs in Attribute Reduction. Fundamenta Informaticae 61(2004) 159-181
4. Slezak, D.: Approximate Entropy Reducts. Fundamenta Informaticae 53(2002) 365-390
5. Kryszkiewicz, M.: Rough Set Approach to Incomplete Information Systems. Information Sciences 112(1998) 39-49
6. Zhou, X.Z., Huang, B.: Rough Set-based Attribute Reduction under Incomplete Information Systems. Journal of Nanjing University of Science and Technology 27(2003) 630-635
7. Li, R.P., Wang, Z.O.: An Entropy-based Discretization Method for Classification Rules with Inconsistency Checking. Proceedings of 2002 ICMLC. IEEE Press, Beijing (2002) 243-246
8. Wang, G.Y., Yu, H., Yang, D.C.: Decision Table Reduction Based on Conditional Information Entropy. Chinese Journal of Computers 25(2002) 759-766
9. Duntsch, I., Gediga, G.: Uncertainty Measures of Rough Set Prediction. Artificial Intelligence 106(1998) 109-137

Learning k-Nearest Neighbor Naive Bayes for Ranking*

Liangxiao Jiang[1], Harry Zhang[2], and Jiang Su[2]

[1] Faculty of Computer Science, China University of Geosciences,
Wuhan,430074, P.R.China

[2] Faculty of Computer Science, University of New Brunswick,
P.O. Box 4400, Fredericton, NB,E3B 5A3, Canada

Abstract. Accurate probability-based ranking of instances is crucial in many real-world data mining applications. KNN (k-nearest neighbor) [1] has been intensively studied as an effective classification model in decades. However, its performance in ranking is unknown. In this paper, we conduct a systematic study on the ranking performance of KNN. At first, we compare KNN and KNNDW (KNN with distance weighted) to decision trees and naive Bayes in ranking, measured by AUC (the area under the Receiver Operating Characteristics curve). Then, we propose to improve the ranking performance of KNN by combining KNN with naive Bayes. The idea is that a naive Bayes is learned using the k nearest neighbors of the test instance as the training data and used to classify the test instance. A critical problem in combining KNN with naive Bayes is the lack of training data when k is small. We propose to deal with it using sampling to expand the training data. That is, each of the k nearest neighbors is "cloned" and the clones are added to the training data. We call our new model instance cloning local naive Bayes (simply ICLNB). We conduct extensive empirical comparison for the related algorithms in two groups in terms of AUC, using the 36 UCI datasets recommended by Weka[2]. In the first group, we compare ICLNB with other types of algorithms C4.4[3], naive Bayes and NBTree[4]. In the second group, we compare ICLNB with KNN, KNNDW and LWNB[5]. Our experimental results show that ICLNB outperforms all those algorithms significantly. From our study, we have two conclusions. First, KNN-relates algorithms performs well in ranking. Second, our new algorithm ICLNB performs best among the algorithms compared in this paper, and could be used in the applications in which an accurate ranking is desired.

1 Introduction

Classification is one of the most important tasks in data mining. In classification, a classifier is learned from a set of training instances with class labels, and an in-

* This work was supported by Excellent Youth Foundation of China University of Geosciences(No.CUGQNL0505) and Natural Science Foundation of Hubei of China(No.2001ABB006 and No.2003ABA043).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 175–185, 2005.

stance x is often represented by a tuple of attibutes $< a_1(x), a_2(x), \ldots, a_n(x) >$, where $a_i(x)$ denotes the value of the th attribute A_i of x. The performance of a classifier is typically measured by its classification accuracy. Many classifiers can also produce the class probability estimates $p(c|x)$ that is the probability of an instance x in the class c, as a by-product. Thus, a ranking of instances based on the class probabilities would be generated. Indeed, in many data mining applications, such a ranking is useful. For example, in direct marketing, we often need to deploy different promotion strategies to customers with different likelihoods of buying some products, in which a ranking of customers in terms of their likelihoods of buying is desired.

KNN has been widely used for decades as an effective classification model. KNN is based on a distance function that measure the difference or similarity between instances. Given a test instance x, its k closest neighbors $y_1, \cdots, y_k$, are found and a vote are conducted to assign the most common class to x. That is, the class of x, denoted by $c(x)$, is determined by the following equation.

$$c(x) = \arg\max_{c \in C} \sum_{i=1}^{k} \delta(c, c(y_i)), \tag{1}$$

where $c(y_i)$ is the class of y_i, and δ is a function that $\delta(u, v) = 1$ if $u = v$.

KNN also produces an estimate $\tilde{p}(c|x)$ of the class probability $p(c|x)$, using common voting. That is, $\tilde{p}(c|x)$ is the fraction of instances of class c in the k nearest neighbors, shown in the following equation.

$$\tilde{p}(c|x) = \frac{\sum_{i=1}^{k} \delta(c, c(y_i))}{k}. \tag{2}$$

Essentially, KNN can be also viewed as a probability-based classifier, shown in Equation 3. Thus, improving the probability estimates of KNN will also lead to an improvement in its classification performance.

$$c(x) = \arg\max_{c \in C} \tilde{p}(c|x). \tag{3}$$

From Equation 2, intuitively, the probability estimates yielded by KNN should be poor, since they are estimated from only the k nearest instances, instead of the whole training data. Thus, its ranking performance should be poor too. We will see later that, in fact, this intuition is sort of wrong.

When we study the ranking performance of a classifier, how to evaluate it is a question. In most scenarios in data mining, the underlying true ranking of training instances is unknown, and only a set of instances with class labels is given. Fortunately, The area under Receiver Operating Characteristics curve, or simple AUC, could be used for this purpose[6].

In recent years, AUC has attracted considerable attention in machine learning and data mining community. Hand and Till[7] show that, for binary classification, AUC is equivalent to the probability that a randomly chosen instance of class $-$ will have a smaller estimated probability of belonging to class $+$ than a randomly chosen instance of class +. They present a simple approach to calculating the AUC of a classifier G below.

$$\hat{A} = \frac{S_0 - n_0(n_0+1)/2}{n_0 n_1}, \tag{4}$$

where n_0 and n_1 are the numbers of negative and positive instances respectively, and $S_0 = \sum r_i$, where r_i is the rank of i_{th} positive instance in the ranking.

From Equation 4, it is clear that AUC is essentially a measure of the quality of a ranking. For example, the AUC of a ranking is 1 (the maximum value of AUC) if there is no positive instance preceding a negative instance.

In recent years, reseachers have started to study the ranking performance of traditional classification models, such as decision trees, naive Bayes and SVM (support vector machine) [3, 8, 9]. To our knowledge, there has been no work on studying the ranking performance of KNN-related algorithms (KNN and its variants). The motivation of this paper is to study systematically the performance of KNN-related algorithms in ranking, measured by AUC.

The rest of the paper is organized as follows. In Section 2, we introduce the related work on learning classifiers with accurate ranking. In Section 3, we empirically compare KNN-related algorithms with decision trees and naive Bayes in AUC, and then propose a novel algorithm for learning k-nearest neighbor naive Bayes for ranking. We also present the experimental results of an extensive empirical comparions on various algorithms in Section 3. In Section 4, we make a conclusion and outline our main directions for future research.

2 Related Work

In recent years, researchers have paid considerable attention to exploring learning algorithms for yielding accurate ranking, since classification is not enough in many applications, such as data mining. A substantial amount of research work has been focussed on decision trees. It has been observed that decision trees produce poor probability estimates. Provost and Domingos[3] point out that the decision tree representation can approximate any probability distribution as accurately as possible, but modern decision tree algorithms are biased against building a tree with accurate probability estimates. They propose using Laplace correction and turning off the reduced-error pruning in C4.5[10] to improve the probability estimates. The resulting algorithm is called C4.4. They compared C4.4 to C4.5 by empirical experiments, and showed that C4.4 is a significant improvement over C4.5 with regard to AUC.

Ling and Yan also propose a method to improve the AUC of a decision tree[8]. They present a novel probability estimation algorithm, in which the class probability of an instance is an average of the probability estimates from all leaves of the tree, instead of only using the leaf into which it falls. In other word, each leaf contributes to the class probability estimate of an instance.

Kohavi[4] presents a model NBTree to combine a decision tree with naive Bayes. In an NBTree, a local naive Bayes is deployed on each leaf of a traditional decision tree, and an instance is classified using the local naive Bayes on the leaf into which it falls. The experiments showed that NBTree outperforms naive Bayes significantly in terms of classification.

Some other traditional learning algorithms, such as decision trees[3, 8], naive Bayes[11] and SVM[9], have been studied in terms of ranking. To our knowledge, there has been no systematic study on the performance of KNN-related algorithms in producing accurate probability estimates or probability-based rankings. Instead, a substantial amount of work has been done on improving the classificaiton accuracy of KNN. Recently, researchers have observed that an significant improvement can be achieved by combining KNN with naive Bayes[5, 12, 13]. That is, a naive Bayes is deployed in the neighborhood of the test instance, consisting of its k nearest neighbors. Indeed, naive Bayes is a simple but effective classifier, which is based on the assumption that all the attributes are independent given the class (the conditional independence assumption). In addition, it performs well when the size of training data is small[4]. Thus, naive Bayes is a suitable local model within KNN. The idea for combining KNN with naive Bayes is quite straightforward. Whenever a test instance is classified, a local naive Bayes is trained using the k nearest neighbors of the test instance, with which the test instance is classified. The classification of the local naive Bayes is based on the following equation.

$$c(x) = \arg\max_{c \in C} p(c) \prod_{i=1}^{k} p(a_i(x)|c), \tag{5}$$

where x is the test instance. The parameters of the local naive Bayes are the probabilities $p(c)$ and $p(a_i(x)|c)$ in Equation 5 that are estimated from the local training data (the k nearest neighbors of x) based on frequency.

Frank et al.[5] present an model to combine KNN with naive Bayes, called locally weighted naive Bayes(LWNB). In LWNB, each of nearest neighbors is weighted in terms of its distance to the test instance. Then a local naive Bayes is built from the weighted training instances. Their experiments show that LWNB outperforms naive Bayes significantly.

Most of the existing research works on combining KNN with naive Bayes are motivated by improving naive Bayes through relaxing the conditional independence assumption using lazy learning. It is expected that there are no strong dependences within the k nearest neighbors of the test instance, although the attribute dependences might be strong in the whole data.

3 Probability-Base Ranking of KNN

3.1 Experiment Methodology

The study in this paper is mostly based on experiments. Thus, we first introduce the setup of our experiments. We run our experiments on the 36 UCI data sets recommended by Weka[2]. All the preprocessing stages of data sets are carried out by the Weka[14]. They mainly include the following three processes:

1. We use the filter of ReplaceMissingValues in Weka to replace the missing values of attributes.
2. We use the filter of Discretize in Weka to discretize numeric attributes. Thus, all the attributes are treated as nominal.
3. It is well-known that, if the number of values of an attribute is almost equal to the number of instances in a data set, this attribute does not contribute any information to classification. So we use the filter of Remove in Weka to delete these attributes.

In our experiments, we use the Laplace estimation to avoid the zero-frequency problem. Assume that there are p instances of the class c, N total instances, and C total classes in the training data. The frequency-based estimation calculates the estimated probability $p(c) = \frac{p}{N}$. The Laplace estimation calculates the estimated probability $p(c) = \frac{p+1}{N+C}$. In the Laplace estimation, $p(a_i(x)|c) = \frac{1+N_{ic}}{N_i+N_c}$, where N_{ic} is the number of instances in class c and with $A_i = a_i(x)$, N_c is the number of instances in class c, and N_i is the number of values for attribute A_i.

All algorithms are implemented within the Weka[14]. Multi-class AUC is calculated by the M-measure[7]. The AUC of a classifier on a data set is obtained by averaging the result from a ten-fold cross validation. Runs with the various algorithms are carried out on the same training sets and evaluated on the same test sets. Finally, we conduct two-tailed t-test with significantly different probability of 0.95 to compare each pair of algorithms. That is, we speak of two results for a data set as being "significantly different" only if the difference is statistically significant at the 0.05 level according to the corrected two-tailed t-test.

3.2 The Ranking Performance of KNN

We have studied the ranking performance of KNN, measured by AUC, by experimentally comparing KNN with naive Bayes and C4.4. Table 1 shows the detailed experimental results at $k = 10$, and a summary of t-test results at $k = 5, 10, 30$ is shown in Table 2. This paper, we only present the detailed experimental results at $k = 10$ for KNN-related algorithms, due to the space limit. But in the summary of t-test, we present comparison results at $k = 5, 10, 30$. From Table 1 and 2, we have a few observations on KNN as follows:

1. KNN performs worse than naive Bayes in ranking, when k is small. The AUC scores of KNN are lower than naive Bayes' in 10 data sets, and higher than naive Bayes' in 5 data sets at $k = 5$; and the cresponding numbers are 6 and 4, respectively, at $k = 10$.
2. KNN outperforms C4.4 in ranking. The AUC scores of KNN are higher than C4.4's at all the k values in Table 4. In addition, KNN outperforms C4.4 in larger margin at larger k values.
3. The ranking performance of KNN improves as k increases.

Generally, the ranking performance of the traditional KNN is poor when k is small. In real applications of KNN, a small k value is preferred, since the classification performance of KNN typically degrades as the increase of k. In addition, small k conforms closer to the data.

It is a natural extension to KNN that weights the instances in the neighborhood in terms of their distance to the test instance. The resuling model is called k-nearest neighbor with distance weighted (KNNDW). The probability estimate $\tilde{p}(c|x)$ yielded by KNNDW is shown in Equation 6.

$$\tilde{p}(c|x) = \frac{\sum_{i=1}^{k} w_i \delta(c, c(y_i))}{\sum_{j=1}^{k} w_i}, \tag{6}$$

where w_i is the weight of y_i, which is a function of the distance $d(x, y_i)$. In our experiments, $w_i = \frac{1}{d^2(x,y_i)}$.

Our experiments show that KNNDW outperforms KNN in ranking. From Table 4, you can see that the number of data sets on which KNNDW has higher AUC scores is significantly greater than the converse.

3.3 *K*-Nearest Neighbor Naive Bayes for Ranking

As we showed in Section 1, the probability estimate of KNN is based on a voting within the neighborhood of the test instance. It is believed that a more sophisticated local model within the neighborhood, instead of voting, would improve probability estimates. It is natural to learn a local naive Bayes for a test instance using only the k nearest neighbors. Although the conditional independence assumption of naive is always violated on the whole training data, it is expected that the dependences within the neighborhood of the test instance are not strong and thus naive Bayes performs better. However, when the local naive Bayes is learned from only the k nearest neighbors, the training data tends to be insufficient, espeically when k is small. Thus, the parameters of naive Bayes cannot be accurately estimated. Then, the performance of a local naive Bayes would be poor. In NBTree[4], a threshold on the size of the training data on a decision node is set to avoid this problem. Each node should have at least 30 training instances. In LWNB[5], Laplace estimation has been used to smooth probability estimates, and a relatively large k, such as $k = 50$, is chosen.

We propose to an approach to handling the issue of lack of training data by expanding the neighborhood. We “clone” each neighbor in terms of its distance to the test instance and add the clones to the training data. Thus, the parameters in naive Bayes can be estimated more accurately and reliably, and the resulting local naive Bayes performs better.

Our sampling (cloning) is based on an explicit function, defined in Equation 7, which measures the similarity between two instances with nominal attributes. Let x and y are two instances, their similarity, denoted by $s(x, y)$, is defined as:

$$s(x, y) = \sum_{i=1}^{n} \delta(a_i(x), a_i(y)). \tag{7}$$

Given a test instance x, for each instance y in its neighborhood, $s(x, y)$ clones of y are added to the training data. Then, a local naive Bayes is learned from the expanded training data with which x is classified. We call our method [illegible], or simply ICLNB. Its algorithm is depicted below.

```
**Algorithm** ICLNB($\mathbf{T}$, $k$, $x$)
**Input** : a set $\mathbf{T}$ of training instances, integer $k$, and a test instance $x$.
**Output** : the class of $x$
    1. Find $x$'s $k$ nearest neighbors $y_1, \cdots, y_k$, from $\mathbf{T}$.
    2. Local training set $\mathbf{L} = \{y_1, \cdots, y_k\}$
    3. For each neighbor $y_i$ of $x$
        – Compute $s(x, y_i)$ using the similarity function in Equation 7.
        – Add $s(x, y_i)$ clones of $y_i$ to $\mathbf{L}$.
    4. Create a local naive Bayes NB using $\mathbf{L}$ as the training data.
    5. Use NB to produce the class $c(x)$ and the probability estimate $\tilde{p}(c|x)$.
```

ICLNB is based on instance sampling, different from the instance weighting in LWNB[5]. ICLNB replicates instances in order to improve the parameter estimates of naive Bayes, and thus leads to more accurate probability estimates from the local naive Bayes. On the other hand, the instance weighting of LWNB aims to differentiate the contributions of instances to classification, and is not necessarily helpful to the probability estimates of the local naive Bayes, which will be shown by the experimental results in Section 3.4.

3.4 Experimental Results for ICLNB

We conducted two group of comparisons. In the first group, we compared ICLNB with KNN, naive Bayes, NBTree, and C4.4. Table 1 and 2 show the experimental results at $k = 10$ and a summary of t-test results at $k = 5, 10, 30$ respectively. In the second group, we compared ICLNB with the KNN-related algorithms, including KNN, KNNDW, LWNB, and the experimental results at $k = 10$ and a summary of t-test results at $k = 5, 10, 30$ are shown in Table 3 and 4 respectively.

From Table 1 and 2, we can see that ICLNB generally outperforms all the other types of algorithms compared in this paper in AUC. We summarize the highlights as follows:

1. ICLNB outperforms naive Bayes significantly. The $w/t/l$ values between ICLNB and NB respectively is 9/25/5, 9/24/3, and 9/22/5 at $k = 5, 10, 30$.
2. ICLNB outperforms C4.4 significantly. The $w/t/l$ values between ICLNB and C4.4 are 11/23/2, 13/22/1, and 14/21/1 at $k = 5, 10, 30$, respectively. Notice

Table 1. Experimental results on AUC and standard deviation. ICLNB: instance cloning local naive Bayes; KNN: k-nearest neighbor; NB: naive Bayes; NBTree: naive Bayes tree; C4.4: C4.5 with laplace correction and without tree pruning. The value of **K** in each related algorithm is **10**

Datasets	ICLNB	KNN	NB	NBTree	C4.4
anneal	95.46±3.77	94.52±3.94	95.9±1.3	96.45±0.28	93.78±2.9
anneal.ORIG	94.77±3.74	92.19±7.17	94.49±3.67	94.71±3.74	92.69±3.15
audiology	71.11±0.7	70.93±0.74	70.96±0.73	71.14±0.71	70.58±0.63
autos	94.13±2.69	89.55±2.95	89.18±4.93	93.93±2.68	90.73±4.52
balance-scale	72.2±2.91	65.86±2.94	84.46±4.1	84.46±4.1	63.06±6.18
breast-cancer	61.03±12.37	62.92±12.49	69.71±15.21	68.95±11.27	59.3±12.03
breast-w	99.41±0.74	98.37±1.59	99.19±0.87	99.21±0.73	97.85±1.86
colic	82.19±4.82	86.74±5.7	83.71±5.5	85.92±6.3	85.02±7.03
colic.ORIG	76.39±6.37	76.95±6.5	80.67±6.98	80.06±8.69	80.56±8.94
credit-a	90.09±3.33	91.59±3.55	92.09±3.43	91.48±3.52	89.42±3.1
credit-g	75.11±5.64	75.97±5.23	79.27±4.74	77.75±5.97	69.62±5
diabetes	79.28±6.08	78.35±5.68	82.31±5.17	82.31±5.17	75.5±5.76
glass	84.43±6.05	82.53±4	80.5±6.65	82.53±8.46	82.36±4.38
heart-c	83.57±1.05	83.8±0.77	84.1±0.54	83.96±0.51	83.1±1.19
heart-h	83.51±0.86	83.6±0.79	83.8±0.7	83.78±0.62	83.04±0.85
heart-statlog	88.27±3.51	90.6±4.82	91.3±4.19	89.66±3.42	81.36±9.15
hepatitis	85.2±14.49	86.97±9.3	88.99±8.99	88.03±8.29	82.03±14.04
hypothyroid	84.99±10.95	82.25±10.83	87.37±8.52	87.01±9.1	81.58±8.8
ionosphere	98.14±1.27	95.11±4.24	93.61±3.36	96.84±2.16	93.1±3.76
iris	98.75±2.12	97.83±3.12	98.58±2.67	98.08±2.67	97.33±2.63
kr-vs-kp	99.54±0.36	99.07±0.46	95.17±1.29	99.17±0.68	99.95±0.06
labor	95±11.25	95.83±10.58	98.33±5.27	100±0	74.17±31.04
letter	99.75±0.05	99.25±0.15	96.86±0.24	98.47±0.15	95.39±0.39
lymph	89.61±2.02	89.33±3.06	89.69±1.49	89.08±2.08	87.26±3.75
mushroom	100±0	100±0.01	99.79±0.04	100±0	100±0
primary-tumor	77.77±1.58	77.72±1.66	78.85±1.35	78.26±1.75	75.48±2.33
segment	99.66±0.17	98.82±0.26	98.51±0.46	99.09±0.43	98.85±0.32
sick	98.97±0.37	97.28±1.33	95.91±2.35	94.46±2.65	99.07±0.35
sonar	89.98±8.08	86.65±8.53	85.48±10.82	79.72±12.51	77.01±8.59
soybean	99.38±0.75	96.77±1.79	99.53±0.6	99.33±0.64	91.43±2.6
splice	98.47±0.63	97.9±0.71	99.41±0.22	99.41±0.22	98.14±0.72
vehicle	88.88±2.12	88.49±2.87	80.81±3.51	85.83±2.9	86.5±2.28
vote	98.72±1.12	97.88±1.27	96.56±2.09	98.82±1.61	96.77±2.96
vowel	99.7±0.25	96.12±0.87	95.81±0.84	98.66±0.68	91.28±2.46
waveform-5000	91.42±0.89	91.62±0.65	95.27±0.58	93.35±1.32	80.83±1.24
zoo	89.88±4.05	89.42±4.44	89.88±4.05	89.88±4.05	88.88±4.5
Mean	89.30±3.53	88.58±3.75	89.61±3.54	89.99±3.34	85.92±4.71

that C4.4 is the state-of-the-arts decision tree learning algorithm designed for yielding accurate rankings.

3. ICLNB performs better than NBTree in AUC. The $w/t/l$ values between ICLNB and NBTree are 4/28/4, 6/27/3, and 5/31/0 at $k = 5, 10, 30$, respec-

Table 2. Summary on t-test of experimental results: AUC comparisons on KNN, NB, NBTree, and C4.4

		KNN	NB	NBTree	C4.4
K=5	NB	10/21/5			
	NBTree	12/23/1	7/28/1		
	C4.4	5/24/7	4/20/12	2/20/14	
	ICLNB	12/13/1	9/22/5	4/28/4	11/23/2
K=10	NB	6/26/4			
	NBTree	8/26/2	7/28/1		
	C4.4	2/25/9	4/20/12	2/20/14	
	ICLNB	10/25/1	9/24/3	6/27/3	13/22/1
K=30	NB	5/25/6			
	NBTree	8/25/3	7/28/1		
	C4.4	3/23/10	4/20/12	2/20/14	
	ICLNB	11/20/5	9/22/5	5/31/0	14/21/1

tively. As k gets larger, ICLNB performs better than NBTree with larger margin. This fact is quite interesting, since NBTree is similar to ICLNB in the sense that both have naive Bayes as a local model. It indicates that KNN would have a better potential than decision trees in ranking.

From Table 3 and 4, we can see that ICLNB achieves a significant improvement to all other KNN-related algorithms compared in AUC scores. Now, we summarize the highlights as follows:

1. ICLNB outperforms all other three algorithms significantly in AUC. For example, compared to LWNB, ICLNB wins in 6 data sets, ties in 29 data sets and loses in 1 data set, at both $k = 5$ and $k = 10$.
2. KNNDW outperforms LWNB $k = 5$ (wins in 7 data sets, loses in 1 data sets); and there is no significant difference when $k = 10$ and $k = 30$. This fact shows that weighting instances does not help to improve the probability estimates, although it results in a significant improvement in classification.
3. Both KNNDW and LWNB are significantly better than KNN. It shows that there is considerable space for improving the probability estimates of KNN.

4 Conclusions

In this paper, we have conducted a systematic empirical study on the ranking performance of KNN-related algorithms. We found that KNN-related algorithms performs well compared to naive Bayes and decision trees. We proposed an approach of combining KNN with naive Baye to improving the ranking performance of KNN, and presented an instance cloning based method to deal with the problem of lack of training data for the local naive Bayes. Our experimental results showed that our new algorithm ICLNB significantly outperforms the KNN-related algorithms KNNN, KNNDW and LWNB. It also performs better

Table 3. Experimental results on AUC ICLNB: instance cloning local naive Bayes; KNN: *k*-nearest neighbor; KNNDW: *k*-nearest neighbor with distance weighted; LWNB: locally weighted naive Bayes. The value of **K** in each algorithm is **10**

Datasets	ICLNB	KNN	KNNDW	LWNB
anneal	95.46±3.77	94.52±3.94	96.04±1.61	94.95±4.93
anneal.ORIG	94.77±3.74	92.19±7.17	94.41±3.61	94.38±2.35
audiology	71.11±0.7	70.93±0.74	71.14±0.6	71.06±0.65
autos	94.13±2.69	89.55±2.95	94.05±2.85	94.18±2.95
balance-scale	72.2±2.91	65.86±2.94	65.86±2.94	73.25±3.57
breast-cancer	61.03±12.37	62.92±12.49	65.04±12.54	62.44±12.56
breast-w	99.41±0.74	98.37±1.59	98.99±1.15	99.07±1.55
colic	82.19±4.82	86.74±5.7	87.32±4.55	84.37±4.2
colic.ORIG	76.39±6.37	76.95±6.5	75.22±7.88	73.62±10.18
credit-a	90.09±3.33	91.59±3.55	90.91±3.51	88.75±3.85
credit-g	75.11±5.64	75.97±5.23	76.06±3.89	74.53±5.02
diabetes	79.28±6.08	78.35±5.68	78.79±5.08	72.74±4.03
glass	84.43±6.05	82.53±4	85.65 ±6.55	82.88±7.01
heart-c	83.57±1.05	83.8±0.77	83.7±0.93	83.5±1.14
heart-h	83.51±0.86	83.6±0.79	83.5±0.74	83.38±0.97
heart-statlog	88.27±3.51	90.6 ±4.82	90.33±4.89	88.13±6.81
hepatitis	85.2±14.49	86.97±9.3	84.75±10.92	79.75±10.1
hypothyroid	84.99±10.95	82.25±10.83	79.22±10.68	80.07±12.51
ionosphere	98.14±1.27	95.11 ±4.24	94.16 ±2.52	96.89 ±1.4
iris	98.75±2.12	97.83±3.12	98.08±3.33	97.33±3.87
kr-vs-kp	99.54±0.36	99.07±0.46	99.55±0.21	99.38±0.29
labor	95±11.25	95.83±10.58	96.67±7.03	98.33±5.27
letter	99.75±0.05	99.25±0.15	99.44±0.07	99.43±0.06
lymph	89.61±2.02	89.33±3.06	89.64±2.36	89.19±2.76
mushroom	100±0	100±0.01	100±0	100±0
primary-tumor	77.77±1.58	77.72±1.66	76.9±1.98	76.99±2.68
segment	99.66±0.17	98.82±0.26	99.17±0.21	99.32±0.22
sick	98.97±0.37	97.28±1.33	98.08±0.85	98.03±1.64
sonar	89.98±8.08	86.65±8.53	88.57±8.27	90.27±7.45
soybean	99.38±0.75	96.77±1.79	99.28±0.76	99.31±0.8
splice	98.47±0.63	97.9±0.71	98.35±0.6	97.87±0.64
vehicle	88.88±2.12	88.49±2.87	88.2±2.91	87.84±2.91
vote	98.72±1.12	97.88±1.27	97.76±1.44	98.05±1.89
vowel	99.7±0.25	96.12±0.87	99.28±0.33	99.74±0.19
waveform-5000	91.42±0.89	91.62±0.65	91.44±0.67	87.36±1.17
zoo	89.88±4.05	89.42±4.44	89.88±4.05	89.88±4.05
Mean	89.30±3.53	88.58±3.75	89.04±3.40	88.51±3.66

than the state-of-the-arts learning algorithms naive Bayes, C4.4 and NBTree. Our study suggests that KNN and its variants could be a good model for the data mining applications that requires accurate rankings.

Table 4. Summary on *t*-test of experimental results: AUC comparisons on KNN-related algorithms, including KNN, KNNDW, LWNB

		KNN	KNNDW	LWNB
K=5	KNNDW	8/27/1		
	LWNB	6/27/3	1/28/7	
	ICLNB	12/23/1	6/30/0	6/29/1
K=10	KNNDW	2/28/1		
	LWNB	6/27/3	2/31/3	
	ICLNB	10/25/1	6/30/0	6/29/1
K=30	KNNDW	8/26/2		
	LWNB	9/25/2	4/30/2	
	ICLNB	11/20/5	9/24/3	6/27/3

References

1. Aha, David W., Dennis Kibler, Marc K. Albert. 1991. Instance-Based Learning Algorithms. Machine Learning, vol. 6, pp. 37-66.
2. http://prdownloads.sourceforge.net/weka/datasets-UCI.jar
3. Provost, F. J., Domingos, P.: Tree Induction for Probability-Based Ranking. Machine Learning **52(3)** (2003) 199-215
4. Kohavi, R.: Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96). AAAI Press (1996) 202-207
5. Frank, E., Hall, M., Pfahringer, B.: Locally Weighted Naive Bayes. Proceedings of the Conference on Uncertainty in Artificial Intelligence (2003). Morgan Kaufmann(2003), 249-256.
6. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press (1997) 43-48
7. Hand, D. J., Till, R. J.: A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning **45** (2001) 171-186
8. Ling, C. X., Yan, R. J.: Decision Tree with Better Ranking. Proceedings of the 20th International Conference on Machine Learning. Morgan Kaufmann (2003) 480-487
9. Huang, J., Lu, J., Ling, C., X.: Comparing Naive Bayes, Decision Trees, and SVM with AUC and Accuracy. Proceedings of the Third IEEE International Conference on Data Mining. IEEE Computer Society Press(2003), 553-556.
10. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufmann: San Mateo, CA (1993)
11. Zhang, H., Su, J.: Naive Bayesian Classifiers for Ranking. Proceeding of ECML 2004. Springer (2004) 501-512
12. Xie, Z., Hsu, W., Liu, Z., Lee, M.: SNNB: A Selective Neighborhood Based Naive Bayes for Lazy Learning. Proceedings of the Sixth Pacific-Asia Conference on KDD. Springer (2002) 104-114
13. Zheng, Z., Webb, G. I.,: Lazy Learning of Bayesian Rules. Machine Learning, **41(1)** (2000) 53-84
14. Witten, I. H., Frank, E.: Data Mining –Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann (2000)

One Dependence Augmented Naive Bayes*

Liangxiao Jiang[1], Harry Zhang[2], Zhihua Cai[1], and Jiang Su[2]

[1] Faculty of Computer Science, China University of Geosciences,
Wuhan, P.R.China 430074

[2] Faculty of Computer Science, University of New Brunswick,
P.O. Box 4400, Fredericton, NB, Canada E3B 5A3

Abstract. In real-world data mining applications, an accurate ranking is as important as an accurate classification. Naive Bayes has been widely used in data mining as a simple and effective classification and ranking algorithm. Since its conditional independence assumption is rarely true, numerous algorithms have been proposed to improve naive Bayes, for example, SBC[1] and TAN[2]. Indeed, the experimental results show that SBC and TAN achieve a significant improvement in term of classification accuracy. However, unfortunately, our experiments also show that SBC and TAN perform even worse than naive Bayes in ranking measured by AUC[3, 4](the area under the Receiver Operating Characteristics curve). This fact raises the question of whether we can improve Naive Bayes with both accurate classification and ranking? In this paper, responding to this question, we present a new learning algorithm called One Dependence Augmented Naive Bayes(ODANB). Our motivation is to develop a new algorithm to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. We experimentally tested our algorithm, using the whole 36 UCI datasets recommended by Weka[5], and compared it to Naive Bayes, SBC and TAN. The experimental results show that our algorithm outperforms all the other algorithms significantly in yielding accurate ranking, yet at the same time outperforms all the other algorithms slightly in terms of classification accuracy.

1 Introduction

Classification is one of the most important tasks in data mining. Learning Bayesian classifiers is a process of constructing a special Bayesian networks from a given set of preclassified instances, each of which is represented by a vector of attribute values. Assume $A_i, i = 1, 2, \ldots, n$ are n attributes which take values $a_i, i = 1, 2, \ldots, n$ respectively. Those attributes will be used collectively to predict the value c of the class C. Thus, the Bayesian classifier represented by a Bayesian network can be defined as:

* This work was supported by Excellent Youth Foundation of China University of Geosciences(No.CUGQNL0505) and Natural Science Foundation of Hubei of China(No.2001ABB006 and No.2003ABA043).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 186–194, 2005.

$$\arg\max_{c \in C} P(c)P(a_1, a_2, \ldots, a_n|c) \tag{1}$$

Assume all attributes are independent given the class. That is:

$$P(a_1, a_2, \ldots, a_n|c) = \prod_{i=1}^{n} P(a_i|c) \tag{2}$$

The resulting classifier is called a naive Bayesian classifier, or simply naive Bayes:

$$\arg\max_{c \in C} P(c) \prod_{i=1}^{n} P(a_i|c) \tag{3}$$

It is obvious, Naive Bayes is a probability-based classification model which is based on the assumption that attributes are conditionally mutually independent given the class label. Although Naive Bayes has conceptual and computational simplicity etc many advantages, its unrealistic attribute independence assumption leads to its probability estimations will not be correct[6], if there exists some strong dependent relations among attributes.

Thinking of the limitation of Naive Bayes, in real-world data mining applications, many researchers propose to learn an optimal Bayesian networks to overcome Naive Bayes's limitation of unrealistic attribute independence assumption throughout. Unfortunately, however, it has been proved that learning an optimal Bayesian networks is NP-hard[7]. Therefore, researchers have made a substantial amount of effort to improve Naive Bayes. Research work to improve the Naive Bayes can be broadly divided into two approaches: 1) selecting attributes subsets in which attributes are mutual conditionally independent at most; 2) relaxing the conditional independence assumption by extending the structure of Naive Bayes to represent the dependencies among attributes.

In classification, the predictive ability of a classifier is typically measured by its predictive accuracy on the testing examples. In fact, most classifiers (including Naive Bayes) can also produce probability estimations or "confidence" of the class prediction. Unfortunately, however, this information is completely ignored in classification. This is often taken for granted since the true probability is unknown for the testing examples anyway.

In many data mining applications, however, the classifier's accuracy are not enough, because they cannot express the information how "far-off" (be it 0.45 or 0.01?) is the prediction of each example from its target. For example, in direct marketing, we often need to promote the top X% of customers during gradual roll-out, or we often deploy different promotion strategies to customers with different likelihood of buying some products. To accomplish these tasks, we need more than a mere classification of buyers and non-buyers. We often need a ranking of customers in terms of their likelihood of buying. Thus, a ranking is more desirable than just a classification.

A natural question is how to evaluate a classifier in terms of its ranking performance, rather than classification accuracy. Recently, the area under the Receiver Operating Characteristics curve [3, 4], or simply AUC, has been used for this purpose and received a considerable attention. AUC compares the classifiers' performance cross the entire range of class distributions and error costs and is a good "summary" for comparing two classifiers. Hand and Till [8] show that, for binary classification, AUC is equivalent to the probability that a randomly chosen example of class $-$ will have a smaller estimated probability of belonging to class $+$ than a randomly chosen example of class $+$. They present a simple approach to calculating the AUC of a classifier G below.

$$\hat{A} = \frac{S_0 - n_0(n_0 + 1)/2}{n_0 n_1}, \tag{4}$$

where n_0 and n_1 are the numbers of negative and positive examples respectively, and $S_0 = \sum r_i$, where r_i is the rank of i_{th} positive example in the ranked list. From Equation 4, it is clear that AUC is essentially a measure of the quality of a ranking. For example, the AUC of a ranking is 1 (the maximum value of AUC) if there is no positive example preceding a negative example.

In this paper, Our motivation is to develop a new algorithm to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. In order to achieve our goal, we present a new learning algorithm called One Dependence Augmented Naive Bayes(ODANB). The experimental results show that we have learned improved Naive Bayes with both accurate classification and ranking.

The rest of the paper is organized as follows. In Section2, we introduce the related work on improving Naive Bayes. In Section3, we present our new learning algorithm called One Dependence Augmented Naive Bayes(ODANB) and make a simple analysis on its advantages and disadvantages. In Section4, we describe the experimental setup and results in detail. In Section 5, we draw a conclusion.

2 Related Work

Naive Bayes is a simple but effective classifier. Although its conditional independence assumption is often violated, it performs surprisingly well in classification[9]. This fact raises the question of whether a Naive Bayesian classifier with less restrictive assumptions can perform even better.

In order to tackle this question effectively, we need an appropriate language and efficient machinery to represent and manipulate independence assertions. Both are provided by Bayesian networks[10]. So, learning Bayesian networks from data has become a rapidly growing field of research. Unfortunately, however, it has been proved that learning an optimal Bayesian networks is NP-hard[7]. In order to escape Bayesian networks's learning complexity, learning improved Naive Bayes has attracted much attention from researchers. Research work to improve Naive Bayes can be broadly divided into two categories just as described in introduction.

The first category aims to improve Naive Bayes by selecting attributes subsets in which attributes are mutual conditionally independent. For example, Langley and Sage[1] presented an algorithm called Selective Bayesian Classifiers(simply

SBC). They used a forward greedy search method to select an attribute subset through the whole space of attributes. They use Naive Bayes' accuracy to evaluate alternative subsets of attributes and consider adding each unselected attribute which can improve the classifier's accuracy at most on each iteration. Their experimental results proved their hypotheses that their algorithm will improve Naive Bayes' accuracy in domains that involve correlated attributes without reducing Naive Bayes' accuracy in domains that don't.

The second category aims to improve Naive Bayes by extending the structure of Naive Bayes to represent dependencies among attributes. For example, Friedman and Goldszmidt[2] singled out an algorithm called Tree Augmented Naive Bayes(simply TAN). They hypothesized the structure among all attributes only is tree-like structure, in which the class node directly points to all attributes nodes and each attribute except the root node of the tree has only one parent from another attribute node. As a result, significant improvement in accuracy is achieved for some datasets compared to Naive Bayes.

3 One Dependence Augmented Naive Bayes: ODANB

At first, let us introduce an important definition of conditional mutual information used in our classification algorithm.

Let X,Y,Z are three variables,then the conditional mutual information between X and Y given Z can be defined by the following equation. Roughly speaking, this function measures the information that Y provides about X when the value of Z is known. Some more information about conditional mutual information can been found in [2].

$$I_P(X;Y|Z) = \sum_{x,y,z} P(x,y,z) \log \frac{P(x,y,z)P(z)}{P(x,z)P(y,z)} \tag{5}$$

In order to improve Naive Bayes' performance measured by accuracy and AUC, we present a novel classification algorithm called One Dependence Augmented Naive Bayes(ODANB) to weaken the attribute independence assumption by adding a parent $A_{ip}, i = 1, 2, \ldots, n$ for some attributes $A_i, i = 1, 2, \ldots, n$. Our algorithm classifies instance using the formulation:

$$\arg\max P(C) \prod_{i=1}^{n} P(A_i|A_{ip}, C) \tag{6}$$

where

$$P(A_i|A_{ip}, C) = \begin{cases} P(A_i|A_m, C) \; I_P(A_i; A_m|C) \geq average \\ P(A_i|C) \qquad I_P(A_i; A_m|C) < average \end{cases} \tag{7}$$

where A_m satisfies

$$I_P(A_i; A_m|C) = \max I_P(A_i; A_j|C), j \neq i = 1, 2, \ldots, n \tag{8}$$

and *average* is defined as

$$average = \frac{1}{n(n-1)} \sum_{i=1}^{n} \sum_{j \neq i = 1}^{n} I_P(A_i; A_j | C) \tag{9}$$

Now, let us look back how to calculate $P(A_i|A_{ip}, C)$. At first, we calculate the conditional mutual information $I_P(A_i; A_j|C), i \neq j$ between each pair of attributes, and calculate their average conditional mutual information. Secondly, we take the attribute with maximum conditional mutual information and above (including equal) the average conditional mutual information as one attribute's parent. At last, we calculate this attribute's conditional probability given class and its parent. Of course, if all conditional mutual information between the one attribute and all the other attributes are below the average conditional mutual information, then this attribute hasn't parent. So, we only need to calculate its conditional probability given class.

Compared to Tree Augmented Naive Bayes(TAN)[2], our classification algorithm(ODANB) has the following advantages at least:

1. Our experimental results measured by classification accuracy and AUC have already proved our classification algorithm has better performance than all the other algorithm used to compare.
2. ODANB's effectiveness and efficiency is higher than TAN, because it substitutes calculating each attribute's maximum conditional mutual information for TAN's searching a maximum conditional mutual information weighted spanning tree. ODANB's time complexity only is $o(n^2 \cdot N + n^2)$, where n is the number of attributes and N is the number of training instances. It is lower than TAN's time complexity $o(n^2 \cdot N + n^2 \cdot \log n)$.
3. ODANB is easier for researchers to understand and implement, because its learning process doesn't suffer from TAN's searching a maximum conditional mutual information weighted spanning tree.

4 Experimental Methodology and Results

We run our experiments on the 36 UCI data sets recommended by Weka[5]. All the preprocessing stages of data sets are carried out by the Weka[11]. They mainly include the following three processes:

1. We use the filter of ReplaceMissingValues in Weka to replace the missing values of attributes.
2. We use the filter of Discretize in Weka to discretize numeric attributes.
3. It is well-known that, if the number of values of an attribute is almost equal to the number of instances in the data set, this attribute does not contribute any information to classification. So we use the filter of Remove in Weka to delete these attributes. In these 36 data sets, there only exists three this type of attributes, namely Hospital Number in data set horse-colic.ORIG, Instance Name in data set Splice and Animal in data set zoo.

We conduct our experiments to compare our algorithm (ODANB) on accuracy and AUC with Naive Bayes, SBC and TAN. All algorithms are implemented within the Weka framework[11]. In our all experiments, the accuracy of each classifier is based on the percentage of successful predictions on the test sets of each

Table 1. Experimental results on classification accuracy and standard deviation. ODANB: One Dependence Augmented Naive Bayes; NB: Naive Bayes; SBC: Selective Bayesian Classifiers; TAN: Tree Augmented Naive Bayes with smoothed parameter of 5.0

Datasets	ODANB	NB	SBC	TAN
anneal	96.55±1.11	94.32±2.38	96.88±2.5	96.66±2.35
anneal.ORIG	90.31±2.57	87.53±4.69	88.75±3.72	87.98±3.62
audiology	62.27±10.02	71.23±7.03	76.01±7.05	75.16±8.45
autos	78.55±6.92	64.83±11.18	67.71±11.27	76.07±10.01
balance-scale	91.36±1.38	91.36±1.38	91.36±1.38	86.08±3.18
breast-cancer	69.61±8.45	72.06±7.97	73.45±8.91	66.82±7.01
breast-w	96.99±1.85	97.28±1.84	96.42±2.26	96.71±1.79
colic	81.25±5.46	78.81±5.05	81.77±4.89	77.18±7.04
colic.ORIG	68.76±4.55	75.26±5.26	75.53±6.15	75.51±7.15
credit-a	82.9±3.54	84.78±4.28	85.51±4.16	84.64±5.03
credit-g	73.4±4.58	76.3±4.76	74.1±3.87	73.4±4.12
diabetes	73.84±7.31	75.4±5.85	75.53±5.07	75.13±4.71
glass	60.28±9.31	60.32±9.69	57.99±6.89	55.71±10.81
heart-c	80.46±10.31	84.14±4.16	82.47±7.61	77.53±7.41
heart-h	79.66±5.97	84.05±6.69	79±9.77	79.97±6.39
heart-statlog	80±11.07	83.7±5	79.26±9.75	81.11±3.68
hepatitis	85.13±7.36	83.79±8.79	80.63±6.8	83.83±8.05
hypothyroid	92.63±0.82	92.79±1.02	93.53±0.66	92.79±1.06
ionosphere	90.9±5.1	90.89±3.49	91.17±4.12	90.6±3.83
iris	94.67±8.2	94.67±8.2	97.33±4.66	90.67±11.42
kr-vs-kp	90.52±1.54	87.89±1.81	94.34±1.23	93.18±1.6
labor	90±14.05	93.33±11.65	77±11.91	88±11.46
letter	77.89±0.89	70±0.81	70.57±0.88	80.45±0.91
lymph	82.43±7.18	85.67±9.55	79±6.84	84.38±9.1
mushroom	99.94±0.09	95.57±0.45	99.67±0.23	99.77±0.12
primary-tumor	44.26±4.06	46.89±4.32	46.02±5.19	48.37±5.83
segment	94.2±1.12	88.92±1.95	90.43±1.96	86.36±2.36
sick	97.59±0.48	96.74±0.53	97.59±0.69	97±0.4
sonar	77.02±11.28	77.5±11.99	70.71±12.97	71.62±12.64
soybean	91.51±3.94	92.08±2.34	91.79±2.72	93.41±2.1
splice	93.07±2.34	95.36±1	94.76±1.6	95.39±1.35
vehicle	71.04±2.8	61.82±3.54	60.65±4.73	69.86±3.47
vote	94.04±3.9	90.14±4.17	95.18±3.93	93.12±4.02
vowel	91.82±2.31	67.07±4.21	68.69±3.47	83.43±3.84
waveform-5000	81.26±0.91	79.96±1.92	81.32±1.54	81.52±1.21
zoo	95.18±8.15	94.18±6.6	93.18±7.93	97.09±4.69
Mean	83.37±5.03	82.41±4.88	82.09±4.98	82.96±5.06

data set, and multi-class AUC has been calculated by measure[8]. The accuracy and AUC of each classifier was measured via the ten-fold cross validation for all data sets. Runs with the various classifiers were carried out on the same training sets and evaluated on the same test sets. In particular, the cross-validation folds are the same for all the experiments on each data set. Throughout, we compare

Table 2. Experimental results on AUC and standard deviation. ODANB: One Dependence Augmented Naive Bayes; NB: Naive Bayes; SBC: Selective Bayesian Classifiers; TAN: Tree Augmented Naive Bayes with smoothed parameter of 5.0

Datasets	ODANB	NB	SBC	TAN
anneal	96.53±0.22	95.9±1.3	94.7±3.92	92.97±2.51
anneal.ORIG	95.17±2.76	94.49±3.67	94.35±4.31	85.42±7.04
audiology	70.84±0.58	70.96±0.73	70.98±0.67	70.16±0.55
autos	93.07±4.06	89.18±4.93	90.43±3.43	90.28±2.59
balance-scale	84.46±4.1	84.46±4.1	84.46±4.1	76.47±7.56
breast-cancer	66.57±11.08	69.71±15.21	67.67±12.63	67.4±10.4
breast-w	99.04±1.05	99.19±0.87	99.16±0.62	98.74±1.32
colic	84.48±5.99	83.71±5.5	84.86±7.13	50.6±8.29
colic.ORIG	72.53±5.61	80.67±6.98	81.82±4.9	62.89±7.73
credit-a	90.19±3.79	92.09±3.43	87±3.75	63.3±13.3
credit-g	75.65±6.28	79.27±4.74	77.41±4.67	60.18±6.84
diabetes	80.88±5.95	82.31±5.17	82.79±5.04	74.18±5.87
glass	79.94±6.88	80.5±6.65	80.97±8.37	84.79±4.34
heart-c	83.85±0.79	84.1±0.54	83.87±0.64	82.96±1.12
heart-h	83.23±0.84	83.8±0.7	82.83±1.38	82.69±0.72
heart-statlog	88.18±9.27	91.3±4.19	87.98±6.91	80.12±11.94
hepatitis	86.04±12.18	88.99±8.99	83.62±12.29	53.83±14.97
hypothyroid	86.5±8.64	87.37±8.52	85.25±8.16	84.03±12.22
ionosphere	97.67±1.71	93.61±3.36	92.26±5.26	72.05±7.4
iris	98.58±2.67	98.58±2.67	99±1.46	94.17±5.51
kr-vs-kp	97.13±0.9	95.17±1.29	96.41±0.78	87.21±1.49
labor	91.67±18	98.33±5.27	65.83±32.5	68.33±40.41
letter	98.45±0.16	96.86±0.24	97.03±0.23	94.5±0.25
lymph	89.02±2.62	89.69±1.49	88.14±3.35	85.56±6.98
mushroom	100±0	99.79±0.04	99.98±0.02	99.87±0.04
primary-tumor	78.18±0.78	78.85±1.35	78.88±1.45	76.39±1.9
segment	99.55±0.23	98.51±0.46	98.93±0.42	95.35±1.06
sick	97.48±0.88	95.91±2.35	94.5±4.28	73.25±2.73
sonar	81.64±12.5	85.48±10.82	79.89±13.1	67.4±13.83
soybean	99.46±0.72	99.53±0.6	99.08±0.74	96.73±1.59
splice	99.05±0.57	99.41±0.22	99.14±0.36	97.72±0.68
vehicle	87.97±3.13	80.81±3.51	81.31±4.02	76.86±3.8
vote	98.16±1.47	96.56±2.09	94.26±4.14	93.49±1.38
vowel	99.49±0.2	95.81±0.84	96.12±0.59	92.33±1.23
waveform-5000	94.38±0.62	95.27±0.58	95.12±0.76	78.9±2.03
zoo	89.88±4.05	89.88±4.05	89.06±4.49	89.88±4.05
Mean	89.30±3.92	89.61±3.54	87.92±4.75	80.58±5.99

Table 3. Results of two-tailed t-test on accuracy and AUC. An entry w/t/l means that the algorithm at the corresponding row wins in w data sets, ties in t data sets, and loses in l data sets, compared to the algorithm at the corresponding column. The significantly different probability of two-tailed t-test is 0.95

		NB	SBC	TAN
accuracy	SBC	6/29/1		
	TAN	6/26/4	4/28/4	
	ODANB	10/25/1	6/27/3	5/27/4
AUC	SBC	4/31/1		
	TAN	1/12/23	0/16/20	
	ODANB	9/25/2	8/26/2	22/14/0

our algorithm with each other algorithm via two-tailed t-test with significantly different probability of 0.95, because we speak of two results for a dataset as being "significantly different" only if the difference is statistically significant at the 0.05 level according to the corrected two-tailed t-test[12].

Table 1 show the accuracy of each classifier on the test sets of each data set, the average accuracy are summarized at the bottom of the table. Table 2 show the AUC of each classifier on the test sets of each data set, the average AUC are summarized at the bottom of the table. Table 3 shows the results of two-tailed t-test between each pair of algorithms, each entry $w/t/l$ means that the algorithm at the corresponding row wins in w datasets, ties in t datasets, and loses in l datasets, compared to the algorithm at the corresponding column.

The detailed results displayed in Table 1—Table 3 show that our algorithm outperforms all the other algorithms used to compare measured by accuracy and AUC. Now, we summarize the highlights as follows:

1. Our algorithm's performance on classification measured by accuracy outperforms all the other algorithms. ODANB's average classification accuracy is 83.37, but the best algorithm of the other algorithms is TAN with average classification accuracy of 82.96. Moreover, the $w/t/l$ value between ODANB and TAN is 5/27/4.
2. Our algorithm's performance on ranking measured by AUC outperforms all the other algorithms. Although Our algorithm's average AUC(89.30) is a little lower than that of NB(89.61), the $w/t/l$ value between ODANB and NB is 9/25/2.
3. TAN improves NB's performance on classification measured by accuracy, but they perform even worse than naive Bayes in ranking measured by AUC. The average AUC of NB(89.61) is higher than that of TAN(80.58) significantly. Moreover, the $w/t/l$ value between TAN and NB is 1/12/23.

5 Conclusions

Naive Bayes delivers fast and effective classification with a clear theoretical foundation. However, It is hampered by the limitations of the attribute independence assumption. The current work is motivated by the desire not only to improve Naive Bayes' performance not only on classification measured by accuracy but also on ranking measured by AUC. In this paper, we present a novel classification algorithm called One Dependence Augmented Naive Bayes(ODANB) by adding an attribute parent for some attributes. Our experimental results show that our classification algorithm outperforms Naive Bayes, SBC and TAN measured by accuracy and AUC. In a word, we believe that we have been successful in our goal of developing a data mining algorithm that retains the computational simplicity and direct theoretical foundation of naive Bayes while alleviating the limitations of its attribute independence assumption.

References

1. Langley, P., Sage, S. Induction of selective Bayesian classifiers. in Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, 1994, pp. 339-406.
2. Friedman, Geiger, and Goldszmidt. "Bayesian Network Classifiers", Machine Learning, Vol. 29, 131-163, 1997.
3. Bradley, A. P. (1997). The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern Recognition,30,1145-1159.
4. Provost, F., Fawcett, T.: Analysis and visualization of classifier performance: comparison under imprecise class and cost distribution. Proceedings of the Third International Conference on Knowledge Discovery and Data Mining. AAAI Press (1997) 43-48
5. http://prdownloads.sourceforge.net/weka/datasets-UCI.jar
6. Bennett, P.N., Assessing the Calibration of Naive Bayes' Posterior Estimates. In Technical Report No. CMU-CS100-155. 2000.
7. Chickering, D. M. (1996). Learning Bayesian networks is NP-Complete. In Fisher, D. and Lenz, H., editors, Learning from Data: Artificial Intelligence and Statistics V, pages 121-130. Springer-Verlag.
8. Hand,D. J., and Till,R. J., A simple generalisation of the area under the ROC curve for multiple class classification problems. Machine Learning, 45:171-186, 2001.
9. Domingos, P., Pazzani M.: Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. Machine Learning **29** (1997) 103-130
10. Pearl, J. (1988). Probabilistic Reasoning in Intelligent Systems. San Francisco, CA: Morgan Kaufmann.
11. Witten,I. H., and Frank,E., Data Mining-Practical Machine Learning Tools and Techniques with Java Implementation. Morgan Kaufmann, 2000.
12. Nadeau, C. Bengio, Y. (1999). Inference for the generalization error. In Advances in Neural In- formation Processing Systems 12 (pp. 307-313). MIT Press.

A Genetic *k*-Modes Algorithm for Clustering Categorical Data

Guojun Gan, Zijiang Yang, and Jianhong Wu

Department of Mathematics and Statistics, York University,
Toronto, Ontario, Canada M3J 1P3
{gjgan, zyang, wujh}@mathstat.yorku.ca

Abstract. Many optimization based clustering algorithms suffer from the possibility of stopping at locally optimal partitions of data sets. In this paper, we present a genetic k-Modes algorithm(GKMODE) that finds a globally optimal partition of a given categorical data set into a specified number of clusters. We introduce a k-Modes operator in place of the normal crossover operator. Our analysis shows that the clustering results produced by GKMODE are very high in accuracy and it performs much better than existing algorithms for clustering categorical data.

1 Introduction

As a primary tool of data mining, cluster analysis divides data into meaningful homogeneous groups. Many clustering algorithms have been proposed and studied[1, 2, 3, 4], and optimization (minimizing an object function) has been among popular approaches. Unfortunately, some optimization based clustering algorithms, such as the k-Means algorithm[5] and the k-Modes algorithm[6], may stop at a local minimum of the optimization problem.

To be more precise, let us consider a given database $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ with n objects each described by d categorical variables. Chaturvedi et al.[6] formulated the k-Modes algorithm to be a bilinear clustering model as:

$$X_{n\times d} = W_{n\times k} Z_{k\times d} + \text{error}, \tag{1}$$

where X is the data matrix (x_{ij}) (x_{ij} is the j-component value of $\boldsymbol{x}_i$), W is a binary membership matrix of n objects in k mutually exclusive, non-overlapping clusters(the (i, j) entry of W is 1 if $\boldsymbol{x}_i$ is in the jth cluster, 0 if otherwise), and Z is a matrix of modes(the (j, l) entry of Z is the mode of the jth cluster in the lth dimension). Note that a mode is the most likely value while a center is the mean. For example, the mode of $(1, 1, 1, 0)$ is 1, while the center of $(1, 1, 1, 0)$ is 0.75. The data matrix X in Equation (1) is known, whereas both W and Z are unknown and they are estimated iteratively to minimize an L_p-norm based loss function $L_p = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{d} |x_{ij} - \hat{x}_{ij}|^p$, where x_{ij} and $\hat{x}_{ij}$ are the (i, j)th entry of X and $\hat{X} = WZ$. Note that in the limiting case as $p \to 0$, the L_p-norm based loss

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 195–202, 2005.

function becomes the simple matching distance[7]. The k-Modes algorithm starts with an initial Z, and then iterates between estimating W given the estimates of Z and estimating Z given the estimates of W. This process is repeated until two successive values of the L_0 loss function are equal.

Some difficulties are encountered while using the k-Modes algorithm. One difficulty is that the algorithm can only guarantee a locally optimal solution[6]. To find a globally optimal solution for the k-Modes algorithm, genetic algorithm (GA)[8], originally introduced by Holland[9], has been used. In GA's, the parameters of the search space are encoded in the forms of [illegible] called [illegible]. A GA maintains a [illegible] (set) of N coded strings for some fixed [illegible] N and evolves over [illegible]. During each generation, three genetic operators, i.e. [illegible], [illegible] and [illegible], are applied to the current population to produce a new population. Each string in the population is associated with a fitness value depending on the value of the objective function. Based on the principle of survival of the fittest, a few strings in the current population are selected and each is assigned a number of copies, and then a new generation of strings are yielded by applying crossover and mutation to the selected strings.

GAs have been successfully applied to clustering[10, 11]. In particular, Krishna and Murty proposed a genetic k-Means algorithm(GKA)[12]. This GKA, incorporating GA into the k-Means algorithm, is very effective in recovering the inherent cluster structures and searches faster than some other evolutionary algorithms used for clustering. Unfortunately, GKA works only for numerical data sets. In the present paper, we develop a genetic clustering algorithm (called GKMODE) by integrating a k-modes algorithm[6] introduced by Chaturvedi et al and the genetic algorithm. We must emphasize here that GKMODE is inspired by the GKA, but focuses on clustering categorical data.

2 The Genetic *k*-Means Algorithm

The GKA[12] is a hybrid clustering algorithm that integrates the k-Means algorithm and GA's. GKA is similar to the conventional GA's except that it uses the k-Means operator(KMO), one step k-Means, instead of the crossover operator. Hence GKA retains the best features of GA's and is efficient for clustering.

Denote by $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ a set of n objects with d numerical attributes. (note that we used D as a categorical data set in Section 1). Let $C_1, C_2, ..., C_k$ be k mutually exclusive, non-overlapping clusters of D and let $w_{ij} = 1$ if $\boldsymbol{x}_i \in C_j$, 0 if otherwise, for $i = 1, 2, ..., n$ and $j = 1, 2, ..., k$. Then the matrix $W = (w_{ij})$ has the following properties:

$$w_{ij} \in \{0, 1\} \text{ and } \sum_{j=1}^{k} w_{ij} = 1. \tag{2}$$

Let the within-cluster variation of C_j be $S^{(j)}(W) = \sum\limits_{i=1}^{n} w_{ij} \sum\limits_{l=1}^{d} (x_{il} - z_{jl})$, and the total within-cluster variation, also called squared Euclidean(SE) measure, be

$S(W) = \sum\limits_{j=1}^{k} S^{(j)}(W) = \sum\limits_{j=1}^{k}\sum\limits_{i=1}^{n} w_{ij} \sum\limits_{l=1}^{d}(x_{il} - z_{jl})$, where x_{il} is the l-th component of object $\boldsymbol{x}_i$, and z_{jl} is the l-th component of $\boldsymbol{z}_j$, the center of C_j defined as $z_{jl} = \left(\sum\limits_{i=1}^{n} w_{ij}x_{il}\right) / \left(\sum\limits_{i=1}^{n} w_{ij}\right)$ for $l = 1, 2, ..., d$. The objective is to find $W^* = (w^*_{ij})$ such that $S(W^*) = \min\limits_{W} S(W)$.

In GKA, the search space consists of all the matrices that satisfy (2). A matrix W is encoded as a string s_W of length n such that $s_W(i) = j$ if object $\boldsymbol{x}_i$ belongs to the jth cluster. The initial population $\mathcal{P}(0)$ is selected randomly. To avoid illegal strings, i.e. partitions with empty clusters, $\lfloor \frac{n}{k} \rfloor$ randomly chosen data points are assigned to each cluster and the rest of the points are assigned to randomly chosen clusters.

The selection operator randomly selects a chromosome from the previous population according to the distribution given by $P(s_i) = F(s_i)/\sum\limits_{i=1}^{N} F(s_i)$, where N is the population size, $F(s_i)$ represents fitness value of the string s_i in the population and is defined by

$$F(s_W) = \begin{cases} f(s_W) - (\bar{f} - c\sigma), & \text{if } f(s_W) - (\bar{f} - c\sigma) \geq 0; \\ 0, & \text{otherwise,} \end{cases}$$

where $f(s_W) = -S(W)$, $\bar{f}$ and σ denote the mean and standard deviation of $f(s_W)$ in the current population, respectively, c is a constant between 1 and 3.

The mutation operator changes an allele value depending on the distance between the cluster center and the corresponding data point. To apply the mutation operator to the allele $s_W(i)$ corresponding to object $\boldsymbol{x}_i$, for example, the $s_W(i)$ is replaced with a value chosen randomly from the distribution: $p_j = P(s_W(i) = j) = (c_m d_{max} - d_j)/\left(kc_m d_{max} - \sum\limits_{l=1}^{k} d_l\right)$, where d_j is the Euclidean distance between $\boldsymbol{x}_i$ and $\boldsymbol{z}_j$, $c_m > 1$ and $d_{max} = \max\limits_{1\leq j\leq k} d_j$. To avoid empty clusters, an allele is mutated only when $d_{s_W(i)} > 0$.

KMO is just one step of the k-Means algorithm: (**a**) calculate Z for the given matrix W; (**b**) form $\hat{W}$ by reassigning each data point to the cluster with the nearest center. KMO may result in illegal strings, which can be avoided by some techniques, such as placing in each empty cluster an object from the cluster with maximum within-cluster variation. Lu et al. (2004) proposed a fast genetic k-Means algorithm(FGKA)[13] in which illegal strings are permitted. Using the finite Markov chain theory, GKA is proved to converge to the global optimum.

3 GKMODE

GKMODE is similar to GKA except that k-Modes Operator is used instead of KMO and, most important, illegal strings are permitted. As in GKA, GKMODE has five basic elements: [illegible] and k [illegible]. The search space is the space of all binary membership matrices W

that satisfy (2). Coding in GKMODE is exactly the same as in GKA. The initial population $\mathcal{P}(0)$ is randomly generated as in FGKA[13]. We now describe the genetic operators used in GKMODE in detail.

3.1 The Selection Operator

To describe the selection operator, let us start with the definition of fitness value of a string. The fitness value of a string s_W depends on the value of the loss function $L_0(W)$, the limiting case of $L_p(W)$ as $p \to 0$. Since the objective is to minimize the loss function $L_0(W)$, a string with relatively small loss must have relatively high fitness value. In addition, illegal strings are less desirable and should be assigned low fitness values. As in [13], we defined the fitness value $F(s_W)$ of a string s_W as follows,

$$F(s_W) = \begin{cases} cL_{max} - L_0(s_W), & \text{if } s_W \text{ is legal;} \\ e(s_W)F_{min}, & \text{otherwise,} \end{cases} \tag{3}$$

where c is a constant in the interval $(0, 3)$, L_{max} is the maximum loss of strings in the current population, F_{min} is the smallest fitness value of the legal strings in current population if it exists, otherwise it is defined as 1, and $e(s_W)$ is the legality ratio defined as the ratio of the number of non-empty clusters in s_W over k(so that $e(s_W) = 1$ if s_W is legal).

The selection operator randomly selects a string from the current population according to the distribution given by $P(s_i) = F(s_i)/\sum\limits_{j=1}^{N} F(s_i)$, where N is the population size. The population of the next generation is determined by N independent random experiments, i.e. apply the selection operator N times.

3.2 The Mutation Operator

In GKMODE, mutation changes a string value based on the distances of the cluster mode from the corresponding data point. It performs the function of moving the algorithm out of a local minimum. The closer a data point to a cluster mode, the higher the chance of changing the data point to that cluster.

Precisely, let s_W be a solution string and let $\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_k$ be the cluster modes corresponding to s_W. During mutation, the mutation operator replaces $s_W(i)$ with a cluster number randomly selected from $\{1, 2, ..., k\}$ according to the distribution: $p_j = [c_m d_{max}(\boldsymbol{x}_i) - d(\boldsymbol{x}_i, \boldsymbol{z}_j)]/\sum\limits_{l=1}^{k} [c_m d_{max}(\boldsymbol{x}_i) - d(\boldsymbol{x}_i, \boldsymbol{z}_l)]$, where $c_m > 1$ is a constant, $d(\boldsymbol{x}_i, \boldsymbol{z}_j)$ is the simple matching distance between $\boldsymbol{x}_i$ and $\boldsymbol{z}_j$, and $d_{max}(\boldsymbol{x}_i) = \max\limits_{1 \le j \le k} d(\boldsymbol{x}_i, \boldsymbol{z}_j)$. As in FGKA[13], $d(\boldsymbol{x}_i, \boldsymbol{z}_j)$ is defined as 0 if the jth cluster is empty. In general, mutation occurs with some mutation probability P_m specified by users. By applying the mutation operator, an illegal string may be converted to a legal one and a data point is moving towards a closer cluster with a higher probability.

3.3 The *k*-Modes Operator

In GKA, KMO is used in place of the crossover operator in order to speed up the convergence process. In GKMODE, the k-Modes operator, one step of the k-Modes algorithm[6], is introduced for the same reason. Let s_W be a solution string, k-Modes operator on s_W which yields $s_{\hat{W}}$ consisting of the following two steps: **(a) Estimate** Z**:** Given estimates of W, the mode matrix Z is determined as follows. The (j, l) entry z_{jl} of Z should be the mode of $(x_l : \boldsymbol{x} \in C_j)$, where x_l is the l-component of $\boldsymbol{x}$ and $C_j = \{\boldsymbol{x}_i : s_W(i) = j,\ 1 \le i \le n\}$. The mode matrix Z formed above optimizes the L_0 loss function[6]. **(b) Estimate** W**:** Given estimates of Z, the binary membership matrix W is determined as follows. The loss function $L_0(W)$ can be written as $L_0(W) = \sum_{i=1}^{n} f_i$, where $f_i (1 \le i \le n)$ is defined as $f_i = \sum_{j=1}^{d} \delta(x_{ij}, z_{s_W(i)j})$ ($\delta(x, y) = 0$ if $x = y$, 1 otherwise.). Note that f_i is a function only of $s_W(i)$. Thus to minimize L_0, one can separately minimize f_i with respect to parameter $s_W(i)$ for $i = 1, 2, ..., n$. Since $s_W(i)$ has only k possible values, i.e. $\{1, 2, ..., k\}$, we can try all these k values and select the value that minimizes f_i, i.e. $s_W(i) = \arg \min_{1 \le l \le k} \sum_{j=1}^{d} \delta(x_{ij}, z_{lj})$. To account for illegal string, we define $\delta(x_{ij}, z_{lj}) = +\infty$ if the lth cluster is empty[13]. This new definition here is introduced in order to avoid reassigning all data points to empty clusters. Thus illegal strings remain illegal after the application of k-Modes operator.

4 Experimental Results

GKMODE and the k-Modes algorithm are both coded in Matlab scripting language. Since Matlab is quite slow for loops, GKMODE is also coded in C++ programming language. Our experiments were conducted on a PC with 2.2 Hz CPU and 512M RAM.

4.1 Data Sets

The soybean disease data[14] is used to test our algorithm. We choose this data set to test for the algorithm for three reasons. First, all attributes of the data set can be treated as categorical; Second, the true clustering of the data set is known; Third, the value of the objective function corresponding to the true clustering is the global minimum.

We also tested the algorithm on the Mushroom data, the Congress Voting data and the Zoo data[14]. The true clusterings of the Congress Voting data and the Zoo data have objective function values 1988 and 149, respectively, while the clusterings produced by GKMODE(with parameters $G_{max} = 10, P_m = 0.4, N = 10$) have objective function values 1701 and 132, respectively. The Mushroom data is big, and the algorithm did not stop in 5 hours. Due to the space limit, the results for these three data sets are not presented here.

4.2 Clustering Quality Measures

We used the corrected Rand index[15] to assess the recovery of the underlying cluster structure. Let $D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_n\}$ be a data set, and let $\mathcal{P} = \{C_1, C_2, ..., C_{k_1}\}$ and $\mathcal{P}' = \{C'_1, C'_2, ..., C'_{k_2}\}$ be two clusterings of D. Denote by n_{ij} the number of points simultaneously in C_i and C'_j, i.e. $n_{ij} = |C_i \cap C'_j|$, then the corrected Rand index is defined as

$$\gamma = \frac{\binom{n}{2} \sum\limits_{i=1}^{k_1} \sum\limits_{j=1}^{k_2} \binom{n_{ij}}{2} - \sum\limits_{i=1}^{k_1} \binom{|C_i|}{2} \sum\limits_{j=1}^{k_2} \binom{|C'_j|}{2}}{\frac{1}{2}\binom{n}{2} \left[\sum\limits_{i=1}^{k_1} \binom{|C_i|}{2} + \sum\limits_{j=1}^{k_2} \binom{|C'_j|}{2} \right] - \sum\limits_{i=1}^{k_1} \binom{|C_i|}{2} \sum\limits_{j=1}^{k_2} \binom{|C'_j|}{2}}.$$

The corrected Rand index γ ranges from 0 when the two clusterings have no similarities(i.e. when one consists of a single cluster containing the whole data set and the other only clusters containing single points), to 1 when the two clusterings are identical. Since we know the true clustering of the data set, the true clustering and the resulting clustering are used to calculate γ.

4.3 Results

In the following tests, we select the constants $c = 1.5$, $c_m = 1.5$ and the input number of clusters $k = 4$ for GKMODE. We tested the algorithm for different values of the following parameters: mutation probability P_m, population size N and maximum number of generations G_{max}.

To compare the k-Modes algorithm and GKMODE, we run each of them 100 times. All objects are correctly clustered into the 4 given clusters by GKMODE for these 100 runs. The average clustering accuracy of GKMODE is 100%. However, the average clustering accuracy of the k-Modes algorithm is about 71% and the number of correct clusterings is 26 out of 100. The results show that the GKMODE produces a more accurate clustering result than the k-Modes algorithm. GKMODE is also better than the tabu search based k-Modes algorithm[16], in which the number of correct clusterings is 67 out of 100.

Table 1 gives the clustering results of GKMODE under different sets of parameters. For each set of the parameters (N, P_m, G_{max}), GKMODE is ran 100 times. In these tests, we choose a wide range of the mutation probability, and we see from the table that the average clustering accuracy of GKMODE is above 88% and the number of correct clusterings is at least 49 out of 100. Because of the limit of the number of generations, the algorithm stops before achieving the global optimum in some cases. Even in the worst case, GKMODE is better than the k-Modes algorithm.

From Table 1, we have following observations: **(a)** When N and G_{max} are fixed, the average clustering accuracy tends to decrease when the mutation probability P_m increases except for some cases. **(b)** When N and P_m are fixed, the average clustering accuracy increases when the maximum number of generations increases except for two cases. This makes sense. But larger values of G_{max} make the algorithm run longer. Therefore, there is a trade-off between the run-

Table 1. Clustering results of GKMODE for different parameters, the algorithm runs 100 times for each parameter setting. The input number of clusters is 4. $\bar{\gamma}$ is the average accuracy, $N_{\gamma=1.0}$ is the number of runs that $\gamma = 1.0$

N	P_m	G_{max}	$\bar{\gamma}$	$N_{\gamma=1.0}$	N	P_m	G_{max}	$\bar{\gamma}$	$N_{\gamma=1.0}$
10	0.2	5	0.9982	99	20	0.2	5	1.0	100
	0.2	10	0.9988	99		0.2	10	1.0	100
	0.3	5	0.9962	95		0.3	5	1.0	100
	0.3	10	1.0	100		0.3	10	1.0	100
	0.4	5	0.9212	59		0.4	5	1.0	100
	0.4	10	1.0	100		0.4	10	0.9995	99
	0.5	5	0.9013	56		0.5	5	1.0	100
	0.5	10	1.0	100		0.5	10	1.0	100
	0.6	5	0.8868	52		0.6	5	1.0	100
	0.6	10	1.0	100		0.6	10	0.9977	99
	0.7	5	0.9785	76		0.7	5	0.9962	96
	0.7	10	0.9994	99		0.7	10	0.9973	99
	0.8	5	0.9344	49		0.8	5	0.9673	71
	0.8	10	0.9863	86		0.8	10	0.9961	94

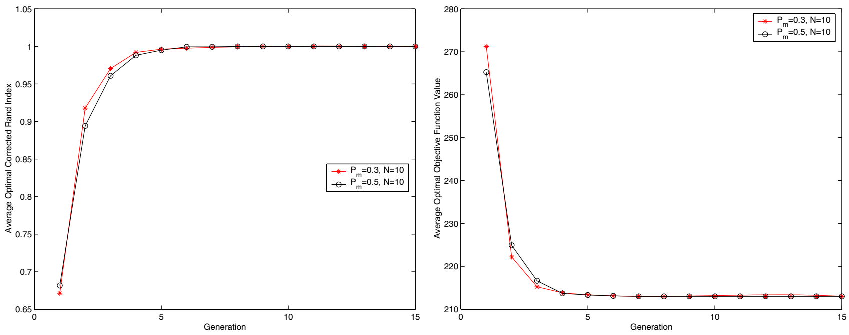


Fig. 1. Average optimal corrected rand index changes(Left) and Average optimal objective function value changes(Right) over generations for 100 runs

ning time and the maximum number of generations. **(c)** When P_m and G_{max} are fixed, the average clustering accuracy of a relatively large population size N is in general higher than that of a relatively small population size N.

We also study the average convergence of the clustering accuracy and the objective function value over generations for two different mutation probabilities. In both cases, GKMODE converges very fast to the extent that it will reach the global optimal clustering in five generations. The convergence of clustering accuracy and the convergence of objective function value are shown in Figure 1.

5 Conclusions

We have introduced the genetic k-Modes algorithm(GKMODE) for finding a globally optimal partition of a given categorical data set into a specified number of clusters. This incorporates the genetic algorithm into the k-Modes algorithm, and our experimental results show that GKMODE is very effective in recovering the underlying cluster structures from categorical data if such structures exist. Note that GKMODE requires the number of clusters k as an input parameter, how to incorporate validity indices for selecting k into GKMODE remains an interesting and challenging problem.

References

[1] Jain, A., Murty, M., Flynn, P.: Data clustering: A review. ACM Computing Surveys **31** (1999) 264–323

[2] Murtagh, F.: A survey of recent advances in hierarchical clustering algorithms. The Computer Journal **26** (1983) 354–359

[3] Cormack, R.: A review of classification. Journal of the Royal Statistical Society. Series A (General) **134** (1971) 321–367

[4] Gordon, A.: A review of hierarchical classification. Journal of the Royal Statistical Society. Series A (General) **150** (1987) 119–137

[5] Hartigan, J.: Clustering Algorithms. John Wiley & Sons, Toronto (1975)

[6] Chaturvedi, A., Green, P., Carroll, J.: k-modes clustering. Journal of Classification **18** (2001) 35 – 55

[7] Huang, Z.: Extensions to the k-means algorithm for clustering large data sets with categorical values. Data Mining and Knowledge Discovery **2** (1998) 283–304

[8] Filho, J., Treleaven, P., Alippi, C.: Genetic-algorithm programming environments. IEEE Computer **27** (1994) 28–43

[9] Holland, J.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Ann Arbor, MI (1975)

[10] Maulik, U., Bandyopadhyay, S.: Genetic algorithm-based clustering technique. Pattern Recognition **33** (2000) 1455–1465

[11] Hall, L., Özyurt, I., Bezdek, J.: Clustering with a genetically optimized approach. IEEE Trans. on Evolutionary Computation **3** (1999) 103–112

[12] Krishna, K., Narasimha, M.: Genetic k-means algorithm. Systems, Man and Cybernetics, Part B, IEEE Transactions on **29** (1999) 433–439

[13] Lu, Y., Lu, S., Fotouhi, F., Deng, Y., Brown, S.: FGKA: a fast genetic k-means clustering algorithm. In: Proceedings of the 2004 ACM symposium on Applied computing, ACM Press (2004) 622–623

[14] Blake, C., Merz, C.: UCI repository of machine learning databases (1998) `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

[15] Hubert, L., Arabie, P.: Comparing partitions. Journal of Classification **2** (1985) 193–218

[16] Ng, M., Wong, J.: Clustering categorical data sets using tabu search techniques. Pattern Recognition **35** (2002) 2783–2790

A Fast Fuzzy Clustering Algorithm for Large-Scale Datasets*

Lukui Shi[1, 2] and Pilian He[1]

[1] Department of Computer Science and Technology, Tianjin University,
Tianjin 300072, China
[2] School of Computer Science and Engineering, Hebei University of Technology,
Tianjin 300130, China
lkshi@eyou.com, plhe@tju.edu.cn

Abstract. The transitive closure method is one of the most frequently used fuzzy clustering techniques. It has $O(n^3\log_2 n)$ time complexity and $O(n^2)$ space complexity for matrix compositions while building transitive closures. These drawbacks limit its further applications to large-scale databases. In this paper, we proposed a fast fuzzy clustering algorithm to avoid matrix multiplications and gave a principle, where the clustering results were directly obtained from the λ-cut of the fuzzy similar relation of objects. Moreover, it was dispensable to compute and store the similar matrix of objects beforehand. The time complexity of the presented algorithm is $O(n^2)$ at most and the space complexity is $O(1)$. Theoretical analysis and experiments demonstrate that although the new algorithm is equivalent to the transitive closure method, the former is more suitable to treat large-scale datasets because of its high computing efficiency.

1 Introduction

Clustering is an automatic unsupervised classification method, which partitions a set of objects into clusters such that objects within a group are more similar to each other than patterns in different clusters. In traditional clustering techniques, each object is assigned to one and only one class. Hence, the clusters are disjoint. However, in practice, it is often desirable to allow partial memberships so that an object can be assigned to more than one class with a degree of belief that the sample belongs to each class. Therefore, it is necessary to introduce the fuzzy set theory [1] into clustering analysis to deal with such problems.

Among the existing fuzzy clustering algorithms, the transitive closure method (TCM) [2], [3] is one of the most often-used fuzzy clustering methods. It is a hierarchical fuzzy clustering algorithm based on fuzzy similar relations and fuzzy equivalent relations. In this procedure, a fuzzy similar matrix is firstly constructed from sample data. Then its transitive closure is computed, which is a fuzzy equivalent matrix. Finally, the hierarchical clustering results are gained under different threshold

* This work was supported by Science-Technology Development Project of Tianjin (No. 04310941R).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 203 – 208, 2005.

values. The transitive closures of fuzzy relations are usually calculated with the square method, whose time complexity and space complexity are separately $O(n^3\log_2 n)$ and $O(n^2)$. It is not fit for large-scale datasets. Another hierarchical fuzzy clustering algorithm, the maximum spanning tree method [4], [5] using graphic theory, is a variant of TCM and has the same complexity as TCM. To reduce the computing complexity, two fast algorithms [6], [7] are proposed. Their time complexity is reduced to $O(n^2)$. However, in both methods similar matrices of objects need to be computed and stored in advance, which leads to not only extra time consumptions but also large memory requirements, particularly for large databases.

In this paper, a fast fuzzy clustering algorithm is put forward, which is equivalent to TCM. It has $O(n^2)$ time complexity at most and $O(1)$ space complexity for not computing similar matrices beforehand.

2 TCM: Transitive Closure Method

TCM is a hierarchical clustering algorithm based on fuzzy equivalent relations. For describing TCM, We assume hereafter that $X=\{x_1, x_2, \ldots, x_n\}$ denotes a set of n objects to be clustered and the ith sample in X is represented by $x_i =(x_{i1}, x_{i2}, \ldots, x_{im})$. Suppose that R: $X\times X \to [0,1]$ is a fuzzy relation on a set X, the similarity between two objects x_i and x_j is defined by the membership function $\mu_R(x_i, x_j)$, which satisfies that $0\leq\mu_R(x_i, x_j)\leq 1$ and $\mu_R(x_i, x_j)= \mu_R(x_j, x_i)$. The membership function can be built with lots of approaches, such as inner product, cosine, correlation coefficient, etc. The choice of methods depends on practical problems. In general, the bigger $\mu_R(x_i, x_j)$ is, the more similar the two samples are. For a fuzzy relation R, the set $R_\lambda=\{<x, y> \mid \mu_R(x, y) \geq\lambda\}$ is called its λ-cut that is a crisp relation. Now let R and S be two fuzzy relations on $X\times Y$ and $Y\times Z$ respectively, then $R°S$ is their composite relation and its membership function $\mu_{R°S}(x, z)$ is defined by

$$\mu_{R°S}(x, z) = \max \{\min [\mu_R(x, y), \mu_S(y, z)], \text{for all } y\in Y\} . \tag{1}$$

It is well known that a fuzzy equivalent relation can determine a fuzzy classification, while a crisp equivalent relation can decide a definite classification. In fact, an arbitrary λ-cut of a fuzzy equivalent relation is a crisp equivalent relation, which can induces an explicit partition under a certain threshold value λ. However, the fuzzy relation generated with membership functions is reflective and symmetric, namely, it is a fuzzy similar relation. We cannot obtain results from it. Nevertheless, its transitive closure is a fuzzy equivalent relation. Therefore, to group objects, it is indispensable to generate the transitive closure $t(R)$ of the fuzzy similar relation R constructed from samples. TCM is just based on the idea. In TCM, the transitive closure is usually produced with the square method, where the following matrix synthesis operations are computed in turn

$$R^2, R^4, R^8, \ldots .$$

The procedure continues until $R^{k}\circ R^k$ is equal to R^k for the first time. Thus, the matrix R^k is the transitive closure of R.

The transitive closure $t(R)$ of R is a fuzzy equivalent relation. For an arbitrary threshold value $\lambda\in[0,1]$, T_λ, the λ-cut of $t(R)$, is a crisp equivalent relation, which can partition the sample set into several clusters. Because a cluster induced from $R_{\lambda 2}$ is included in a cluster induced from $R_{\lambda 1}$ while $\lambda_1\leq\lambda_2$, the hierarchical classifications will be generated while λ varies from 0 to 1. The algorithm is as follows.

Algorithm 1: TCM.

```
Step 1: Construct the fuzzy similar matrix from the given sample set.
Step 2: Compute the transitive closure with the square method.
Step 3: Generate clustering results for different threshold value $\lambda$.
```

In TCM, the time cost of single matrix multiplication is $O(n^3)$ for a set of n data points and at most $\log_2 n$ matrix compositions are made. Therefore, its time complexity is $O(n^3\log_2 n)$. Moreover, constructing and storing the similar matrix not only expends lots of CPU time but also occupies huge memory. As a result, it is difficult for TCM to handle large-scale datasets. In the algorithm given in this paper, its time complexity is much lower than that of TCM and it is not necessary to build the similar matrix of objects in advance.

3 FTCM: A Fast Fuzzy Clustering Algorithm

In TCM, the high computing complexity results from synthetic operations of matrices. Hence, diminishing or avoiding operations of matrix compositions is an effective approach to improve the computing efficiency. In our algorithm, we directly acquire results from the λ-cut of a fuzzy similar relation instead of computing its transitive closure. The following lemmas are important for validating the correctness of our algorithm.

Lemma 1. Let R denote a fuzzy relation on a set X and $t(R)$ be its transitive closure. Then the λ-cut of $t(R)$ is equal to the transitive closure of its λ-cut, namely, $(t(R))_\lambda= t(R_\lambda)$.

Proof: We will prove this equation by showing that each side is a subset of the other side. Suppose that $\lambda\in[0,1]$. By the definition of the fuzzy transitive closure and λ-cut, for arbitrary ordered pairs $<x, y>\in(t(R))_\lambda$ there is a chain of objects $x_1, x_2, \ldots, x_k$ such that $\mu_R(x, x_1)\geq\lambda$, $\mu_R(x_1, x_2)\geq\lambda$, …, $\mu_R(x_k, y)\geq\lambda$. It follows that $<x, x_1>$, $<x_1, x_2>$, …, $<x_k, y>\in R_\lambda$, namely, $<x, y>\in t(R_\lambda)$. Hence, we conclude that $(t(R))_\lambda\subseteq t(R_\lambda)$.

Now suppose that $<x, y>\in t(R_\lambda)$. Then there exists a chain of objects $x_1, x_2, \ldots, x_k$ such that $<x, x_1>$, $<x_1, x_2>$, …, $< x_k, y>\in R_\lambda$. By the definition of λ-cut, it follows that $\mu_R(x, x_1)\geq\lambda$, $\mu_R(x_1, x_2)\geq\lambda$, …, $\mu_R(x_k, y)\geq\lambda$. From this, we can see that $\mu_{t(R)}(x, y)\geq\lambda$, i.e., $<x, y>\in(t(R))_\lambda$. It is concluded that $t(R_\lambda)\subseteq(t(R))_\lambda$. Therefore, $(t(R))_\lambda=t(R_\lambda)$.

Lemma 2. Suppose that R is a fuzzy relation on a set X and all the elements in a subset B of X belong to the same cluster under a threshold value λ. If there are two elements x and y in X such that $x\in B$, $y\notin B$ and $\mu_R(x, y)\geq\lambda$, the element y should be contained in the same cluster as all the elements in B, namely, the element y should be appended to B.

Proof: According to conditions, because all the elements in B belong to the same class for a given λ, it can be concluded that $\mu_R(z, x)\geq\lambda$ for an arbitrary element z in B. By the definition of the fuzzy transitivity, it follows that $\mu_R(z, y)\geq\lambda$. Hence, the object y will be partitioned into the same cluster as all the elements in B.

Lemma 3. Suppose that R is a fuzzy similar relation on a set X and the subsets B and C of X are two different sub-clusters for a given λ. If there are elements x, y and $z\in X$ satisfy the following conditions: 1) $x\in B$, $y\in C$; 2) $z\notin B$, $z\notin C$; 3) $\mu_R(x, z)\geq\lambda$, $\mu_R(y, z)\geq\lambda$, the sub-clusters B and C should be merged, i.e., $B'=B\cup C$.

Proof: According to lemma 2, the element z should belong to not only the same cluster as all the elements in B but also the same cluster as all the elements in C. Using the symmetry of R, we can conclude that $\mu_R(z, y)\geq\lambda$. Therefore, $\mu_R(x, y)\geq\lambda$, namely, two sub-clusters B and C ought to be merged.

Lemma 1 indicates that the λ-cut of the transitive closure of a fuzzy relation equals the transitive closure of its λ-cut. Then the clustering results can be obtained from the transitive closure of the λ-cut under a certain threshold value λ. Lemma 2 shows that how an object should be added to a cluster. Lemma 3 points out that when two sub-clusters should be merged. In addition, in practice, we usually set several certain λ, i.e., the number of values of λ can be considered as a constant. Based on such assumption, the following algorithm is acquired from these three lemmas.

Algorithm 2: FTCM.

Input: dataset X, the number of objects n, threshold value set TS.

Output: clustering results for each $\lambda\in TS$.

For each $\lambda\in TS$, step1 to step 5 are executed.

Step 1: i: =1, k:=1.

Step 2: Create a sub-cluster C_1 and add $X(1)$ to C_1, namely, $C_1=\{ X(1)\}$.

Step 3: i: $=i+1$.

Step 4: Suppose that k sub-clusters C_1, C_2, …, C_k have been gotten, objects will be clustered by the following three rules.

Rule 1: If there is $y\in C_p$, $p=1, 2, \ldots, k$, such that $\mu_R(y, X(i))\geq\lambda$, the element $X(i)$ should be inserted into C_p, i.e., $C_p= C_p\cup\{ X(i)\}$.

Rule 2: If there exist $y_1\in C_p$ and $y_2\in C_q$, p and $q=1, 2, \ldots, k$, such that $\mu_R(y_1, X(i))\geq\lambda$ and $\mu_R(y_2, X(i))\geq\lambda$, the sub-clusters C_p and C_q should be merged, i.e., $C_p= C_p\cup C_q \cup\{ X(i)\}$. Moreover, C_q will be deleted and $k := k-1$.

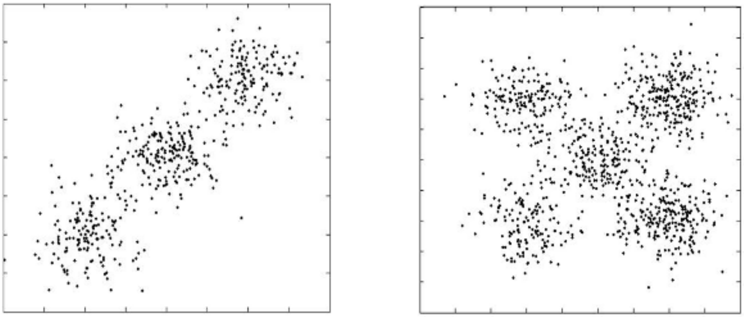
Rule 3: If it follows that $\mu_R(y, X(i))<\lambda$ for any $y\in C_p$, $p=1, 2, \ldots, k$, a new sub-cluster $C_{k+1}=\{X(i)\}$ will yield and $k := k+1$.

Step 5: If $i=n$, stop; otherwise, go to step 3.

The correctness of FTCM has been verified through the above three lemmas. Furthermore, the results from FTCM are equivalent to those from TCM. In FTCM, the frequency of comparisons with other elements is at most n-1 for each element when determining its class. In the whole process, the number of comparisons executed is at most $n(n-1)/2$ and at least n-1. Therefore, the worst time complexity of FTCM is $O(n^2)$. Simultaneously, the similar matrix between objects need not be calculated and stored beforehand. Its space complexity is $O(1)$. It is clear that the time efficiency and the space efficiency of FTCM are both greatly superior to those of TCM.

4 Experiments

In this section, we have tested the performance of FTCM. For other two improved algorithms in [6], [7], because the performance of the proposed algorithm (TFM) in [6] is close to that of the algorithm in [7], we only considered the algorithm TFM while comparing with FTCM. We have compared the executing efficiency of TCM, TFM and FTCM. The three algorithms were implemented in MATLAB6.5. All experiments have been run on a PC with 2.0GHz CPU and 256 MB RAM. We have used both two synthetic sample databases (DS1, DS2, see Fig. 1) and the real database IRIS.



(a) DS1

(b) DS2

Fig. 1. Datasets

The dataset DS1 consists of 500 objects, which includes three Gaussian clusters. The dataset DS2 has 1000 objects, which contains five Gaussian clusters. The database IRIS comprises three clusters of 50 4-dimension objects each, which is usually used to test the performance of clustering algorithms and clustering validity functions. One class is linearly separable from the other two; the latter are not linearly separable from each other. The run time of the three algorithms on three datasets is shown in Table 1.

From Table 1, it is apparent that the executing efficiency of FTCM outperforms those of TCM and TFM. Furthermore, the speed difference between FTCM and TCM also quickly increases with the size of datasets going up.

Table 1. Time cost on three datasets with three algorithms

Datasets	Size of datasets	TCM in seconds	TFM in seconds	FTCM in seconds
IRIS	150	2.20	0.20	0.06
DS1	500	224.70	1.48	0.55
DS2	1000	1693.90	12.42	4.92

5 Conclusions

The transitive closure method is usually used in fuzzy clustering analysis. Its high complexity and large memory requirements limit its applications to large-scale databases. The presented algorithm in this paper is equivalent to TCM and evidently decreases the time and space complexity because of diminishing matrix synthesis operations. The time complexity of the new algorithm is $O(n^2)$ at most and its space complexity $O(1)$. Experimental results also show that the time and space efficiency of FTCM greatly outperforms that of TCM as well as TFM and is fitter to deal with large-scale datasets.

References

1. Zadeh, L. A.: Fuzzy sets. Information and Control, Vol. 8, (1965) 338-353.
2. Zadeh, L. A.: Similarity relations and fuzzy ordering. Information Science, Vol. 3, (1971) 177-200.
3. Tamura, S., Higuchi, S., Tanaka, K.: Pattern Classification Based on Fuzzy Relations. IEEE Trans. Syst. Man Cybernet, Vol. 1, No. 1, (1971) 61-66.
4. Miyamoto, S.: Fuzzy Sets in Formation Retrieval and Cluster Analysis. Kluwer Academic Publishers, Dordrecht (1990).
5. Miyamoto, S.: Fuzzy Graphs as a Basis Tool for Agglomerative Clustering and Information Retrieval. In: O.Optiz, et al. (eds.), Information and Classification: Concepts, Methods and Applications, Springer-Verlag, Berlin, (1993) 268-281.
6. Fubao Wu, Qi Li, Wenzong Song: Transfer Algorithm to Fuzzy Clustering Analysis. Journal of Southeast University of China, Vol. 29, No.2, (1999) 105-110.
7. Jun Ma, Lu Shao: An Optimal Algorithm for Fuzzy Classification Problem. China Journal of Software, Vol. 12, No. 4, (2001) 578-581.

Clustering with Noising Method

Yongguo Liu[1,2], Yan Liu[3], and Kefei Chen[1]

[1] Department of Computer Science and Engineering,
Shanghai Jiaotong University, Shanghai 200030, P.R. China
{liu-yg, chen-kf}@cs.sjtu.edu.cn
[2] State Key Laboratory for Novel Software Technology,
Nanjing University, Nanjing 210093, P.R. China
[3] School of Applied Mathematics,
University of Electronic Science and Technology of China,
Chengdu 610054, P.R. China
liuyan@uestc.edu.cn

Abstract. The minimum sum of squares clustering problem is a nonconvex program which possesses many locally optimal values, resulting that its solution often falls into these traps. In this article, a recent metaheuristic technique, the noising method, is introduced to explore the proper clustering of data sets under the criterion of minimum sum of squares clustering. Meanwhile, K-means algorithm as a local improvement operation is integrated into the noising method to improve the performance of the clustering algorithm. Extensive computer simulations show that the proposed approach is feasible and effective.

1 Introduction

The clustering problem is a fundamental problem that frequently arises in a great variety of fields such as pattern recognition, machine learning, and data mining. In this article, we consider this problem stated as follows: Given N objects in R^m, allocate each object to one of K clusters such that the sum of squared Euclidean distances between each object and the center of its belonging cluster for every such allocated object is minimized. This clustering problem can be mathematically described as follows:

$$\min_{W,C} F(W,C) = \sum_{i=1}^{N} \sum_{j=1}^{K} w_{ij} \parallel x_i - c_j \parallel^2 \tag{1}$$

where $\sum_{j=1}^{K} w_{ij} = 1$, $i = 1,\ldots,N$. If object x_i is allocated to cluster C_j, then w_{ij} is equal to 1; otherwise w_{ij} is equal to 0. In Equation 1, N denotes the number of objects, K denotes the number of clusters, $X = \{x_1,\ldots,x_N\}$ denotes the set of N objects, $C = \{C_1,\ldots,C_K\}$ denotes the set of K clusters, and W denotes the $N \times K$ $0-1$ matrix. Cluster center c_j is calculated as follows:

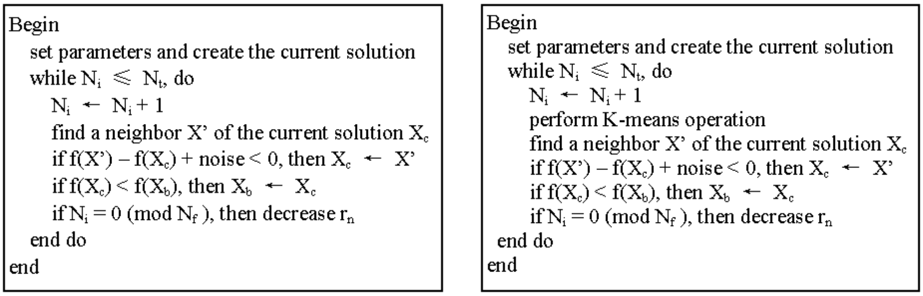
X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 209–216, 2005.

$$c_j = \frac{1}{n_j} \sum_{x_i \in C_j} x_i \tag{2}$$

where n_j denotes the number of objects belonging to cluster C_j. It is known that this problem is a nonconvex program which possesses many locally optimal values, resulting that its solution often falls into these traps. Many clustering approaches have been developed [1]. Among them, K-means algorithm is a very important one, but it is proved to fail to converge to a local minimum under certain conditions [2]. In [3], genetic algorithms are applied to the clustering problem, called GAC in this paper. GAC encodes the clustering partition as a chromosome. After a specified number of generations, the best individual obtained is viewed as the final solution. In [4], tabu search is reported to deal with this problem, called TSC in this paper. The clustering solution is encoded as a string similar to that in GAC. The best solution obtained after a specified number of iterations is viewed as the clustering result. In [5], a simulated annealing based clustering method is proposed, called SAC in this paper. By redistributing objects among clusters probabilistically, this approach can obtain the globally optimal solution under certain conditions. The noising method guiding the local heuristic search procedures to explore the solution space beyond the local optimality is a recent metaheuristic technique firstly reported in [6]. The noising method [7] has been successfully applied to traveling salesman problem, scheduling problem, and multicriteria decision, etc. In this article, our aim is to introduce the noising method to explore the proper clustering under the criterion of minimum sum of squares clustering. In the field of other metaheuristics such as tabu search, to efficiently use tabu search in various kinds of applications, researchers often combine it with the local descent approaches. In [8], Nelder–Mead simplex algorithm, a classical local descent algorithm, and tabu search are hybridized to solve the global optimization problem of multiminima functions. This idea is introduced in this article. Since K-means algorithm is simple and computationally attractive, we view it as a local improvement operation and combine it with the noising method. So, we give two ways to deal with the clustering problem in this article, one does not have K-means operation and another has. These two methods are called NMC and KNMC, respectively. The choice of the algorithm parameters is extensively discussed, and performance comparisons among six techniques are conducted on experimental data sets. As a result, spending much less computational resources than GAC, TSC, and SAC, the noising method based clustering algorithm can get feasible and effective clustering results.

2 The Proposed Method

Instead of taking the genuine data into account directly, the noising method considers the optimal result as the outcome of a series of fluctuating data converging towards the genuine ones. Like some other metaheuristics, the noising method is based on a descent. The main difference with a descent is that, when the objective function value for a given solution is considered, a perturbation called a noise is added to this value. This noise is randomly chosen in an interval of which the range decreases during the iteration process. The final solution is the best solution computed during the iteration process. In this article, noises are added to the variation of the objective function value so as to avoid the clustering problem being trapped by the locally optimal values. It means that the original value of the noise rate r_n should be chosen in such a way that, at the beginning of the iteration process, a bad neighboring solution may be accepted, as it is also the case in simulated annealing for instance. As added noises are chosen in an interval centered on 0, a good neighboring solution may be also rejected, which is different from simulated annealing. The detail discussion about the noising method can be found in [7]. Figure 1 gives the general description of NMC and KNMC. It is seen that they are both observe the architecture of the noising method.



```
Begin
  set parameters and create the current solution
  while $N_i \leqslant N_t$, do
    $N_i \leftarrow N_i + 1$
    find a neighbor X' of the current solution $X_c$
    if $f(X') - f(X_c) + noise < 0$, then $X_c \leftarrow X'$
    if $f(X_c) < f(X_b)$, then $X_b \leftarrow X_c$
    if $N_i = 0 \pmod{N_f}$, then decrease $r_n$
  end do
end
```

```
Begin
  set parameters and create the current solution
  while $N_i \leqslant N_t$, do
    $N_i \leftarrow N_i + 1$
    perform K-means operation
    find a neighbor X' of the current solution $X_c$
    if $f(X') - f(X_c) + noise < 0$, then $X_c \leftarrow X'$
    if $f(X_c) < f(X_b)$, then $X_b \leftarrow X_c$
    if $N_i = 0 \pmod{N_f}$, then decrease $r_n$
  end do
end
```

Fig. 1. General description of NMC (L) and KNMC (R)

In this article, we define the solution the same as that in GAC, TSC, and SAC, which is suitable for computing the objective function value and comparing our methods with these methods. Here, two important concepts are given: probability threshold and K-means operation. In NMC and KNMC, the probability threshold is used to provide a proper neighboring solution for the noising method so as to avoid getting stuck in local optima and find the optimal result. In [4], it is used to create the neighborhood of tabu search. It is described as follows: Given the current solution $X_c = x_1, \ldots, x_i, \ldots, x_N$, $x_i = j$, $j = 1, \ldots, K$ and the probability threshold P, for $i = 1, \ldots, N$, draw a random number $p_i \sim u(0,1)$. If $p_i < P$, then $x_i^{'} = x_i$; otherwise $x_i^{'} = k$, $k = 1, \ldots, K$, $k \neq j$. Here, $X^{'} \neq X_c$, where $X^{'}$ denotes the neighboring solution. In this paper, the probability threshold P is chosen to be 0.95, which is recommended by computer simulations in [4].

Based on the structure of the noising method, KNMC gathers the global optimization property of the noising method and the local search capability of K-means operation together. Here, K-means operation is used to fine-tune the distribution of objects belonging to different clusters and to improve the similarity between objects and their centroids. It is described as follows: Given a solution $X = x_1, \ldots, x_i, \ldots, x_N$, reassign object x_i to cluster C_k, $k = 1, \ldots, K$, iff

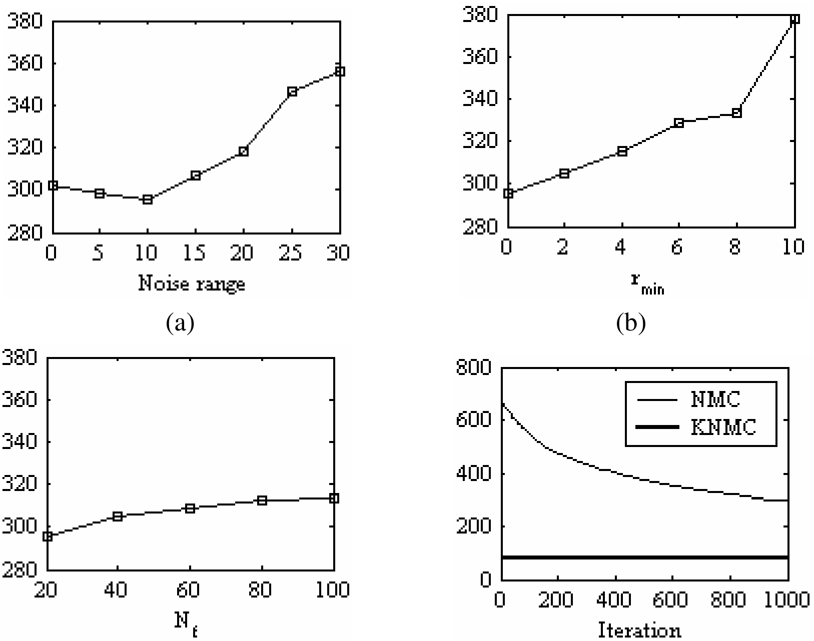
$$\| x_i - c_k \| \leq \| x_i - c_l \|,\ l = 1,\ldots,K \text{ , and } k \neq l \tag{3}$$

Then new cluster centers, $c_1^{'},\ldots,c_K^{'}$, is calculated as follows:

$$c_k^{'} = \frac{1}{n_k} \sum_{x_i \in C_k} x_i \tag{4}$$

where n_k denotes the number of objects belonging to cluster C_k. After this operation, the modified solution is viewed as the current solution X_c.

In order to explore the good performance of NMC and KNMC, we here discuss the choice of different parameters as shown in Figure 2. Each experiment includes 20 independent trials.



(a) (b)

(c) (d)

Fig. 2. Comparison of different parameters

The noise range, the first parameter we consider, is used to determine the range in which the noise rate r_n varies. Based on the noise range and the terminal noise rate $r_{\min}$, the original noise rate $r_{\max}$ can be calculated. In Figure 2(a), the average objective function values for different noise ranges are compared. It is found that this parameter is overlarge or over small will reduce the performance of the proposed approach. When the size of the noise range is equal to 10, the best performance is attained. So, we choose this parameter to be 10. It is found that, in Figure 2(b), the larger the value of $r_{\min}$ the worse. The reason is that an added noise is a random real drawn with a uniform distribution in the interval $[-r_n, +r_n]$, and the noise rate r_n is bounded by two extreme values $r_{\max}$ and $r_{\min}$, then we may make r_n decrease down to 0 so as to get back the genuine function at the end of the noising method. So, in this paper, we choose $r_{\min}$ to be 0, and then $r_{\max}$ is equal to be 10. To control the decrease speed of added noises, we discuss the number of iterations at the fixed noise rate denoted by N_f as shown in Figure 2(c). We find a slow decrease speed can obtain slightly better results than a quick one. In this paper, we take N_f to be 20. In Figure 2(d), NMC and KNMC are compared. It is seen that KNMC equipped with K-means operation is obviously superior to NMC.

3 Experiment Analysis

Before conducting simulation experiments, we analyze the time complexities of algorithms employed in this paper. For TSC, the time complexity is $O(GN_nmN)$, where N_n is the size of the neighborhood and G is the number of iterations. For GAC, the time complexity is $O(GPmN)$, where P denotes the population size and G denotes the number of generations. The time complexity of SAC is $O(GN_sKmN)$, where N_s denotes the number of iterations at the fixed temperature and G denotes the number of iterations during the process that the annealing temperature drops. In our methods, creating the neighboring solution takes $O(N)$ time and K-means operation takes $O(KmN)$ time. Hence, the time complexity of NMC is $O(N_tmN)$ and the time complexity of KNMC is $O(N_tKmN)$, where N_t is the total number of iterations. It is seen that the computational cost of KNMC is higher than that of NMC. However, equipped with K-means operation, the performance of the noising method based clustering algorithm is greatly improved. Furthermore, compared with TSC, GAC, and SAC, KNMC spends the least computational resources. That is, its computational cost of is $K/60$ of GAC, $K/20$ of TSC, and about 1/28 of SAC, respectively. In most cases, K is a small constant. Therefore, the computational cost of KNMC is still very low.

Performance comparisons between our methods and other techniques are conducted in Matlab on an Intel Pentium III processor running at 800MHz with 128MB real memory. Five data sets representing different distribution of objects are chosen to test the adaptability of the proposed method: two artificial data sets (Data-52, Data-62) and three real life data sets (Iris, Crude Oil, and Vowel). Data-52 is a two-dimensional data set having 250 overlapping objects where the number of clusters is five. Data-62 is a two-dimensional data set having 300 nonoverlapping objects where the number of clusters is six. Iris represents different categories of irises having four feature values. The four feature values represent sepal length, sepal width, petal length, and petal width in centimeters. It has three classes with 50 samples per class [9]. Crude Oil has 56 objects, five features and three classes [10]. Vowel consists of 871 Indian Telugu vowel sounds having three features and six classes [11]. In computer simulations, experimental results of NMC and KNMC are obtained after 1000 iterations. Each experiment for all algorithms in this paper includes 20 independent trials. The detail settings of parameters in GAC, TSC, and SAC can be found in their corresponding references. The average and minimum values of the clustering results obtained by six methods for five data sets are shown as Table 1.

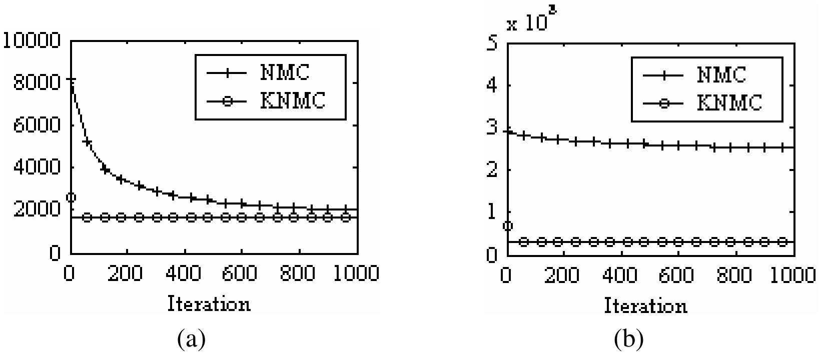
Table 1. Results of six clustering algorithms for five data sets

	Data-52	Data-62	Iris	Crude Oil	Vowel
	Avg(min)	Avg(min)	Avg(min)	Avg(min)	Avg(min)
K-means	488.95(488.09)	1469.85(543.17)	91.76(78.94)	1656.98(1647.19)	32782041.43(30724312.47)
GAC	1464.19(1269.87)	9959.12(8714.13)	96.44(83.40)	1649.99(1647.19)	158113587.46(143218656.39)
TSC	2590.21(2517.35)	19155.37(18406.54)	282.64(256.70)	2122.05(1952.38)	248267782.61(245672828.41)
SAC	488.02(488.02)	821.56(543.17)	78.94(78.94)	1647.24(1647.19)	32243759.40(30724196.02)
NMC	2654.52(2557.31)	19303.58(18005.98)	302.99(242.15)	1995.44(1787.43)	250796549.46(245737316.31)
KNMC	488.69(488.02)	1230.02(543.17)	85.37(78.94)	1647.27(1647.19)	31554139.24(30718120.60)

For Data-52, the optimal value is 488.02, which is only found by SAC and KNMC. Since objects of this data set are overlapping, other four approaches cannot reach the best result in all runs. For Data-62, the optimal result is 543.17. K-means, SAC, and KNMC can attain this value. In most cases, K-means gets stuck at suboptimal values. For Iris, the best value is 78.94, which is attained by K-means, SAC, and KNMC. But K-means is found to achieve this value in 4 of all trials. SAC and KNMC can attain the best value more stably. For Crude Oil, the best value is 1647.19. In this experiment, the performance of KNMC is close to that of SAC. But for Vowel, KNMC is the best one among all methods. It is seen that, for most data sets, KNMC can obtain best values and is superior to others except SAC. Noticeably, GAC, TSC, and NMC fail to attain the best values for most data sets even once and their best values obtained are far worse than the optimal ones. However, we find that these three algorithms can still obtain improved results if more iterations are executed. Meanwhile, the performance of NMC is close to that of TSC but its computational cost is only 1/20 of TSC, which shows NMC is promising to a certain extent.

According to Table 1, we find combining K-means algorithm with the noising method to deal with the clustering problem can take their respective advantages: the global optimization ability of the noising method and the local search capability of K-means algorithm. By combing these two methods, we are able to obtain better results than those obtained by K-means algorithm and NMC. Meanwhile, in most cases, SAC is better than KNMC. But we should remember that the cost of KNMC is only about

1/28 of SAC and the performance of KNMC is very close to even superior to that of SAC. Here, we may greatly increase the computational resource such as the number of iterations in order to attain the results similar to or superior to those of SAC, but we do not think that it is a good way to attain the optimal result by only adding this parameter. For example, in [3], the specified number of iterations where GAC obtains the best result for Crude Oil is up to 10000, which is 20 times computational resources than that of KNMC. Even so, there are still 22% of trials where it cannot obtain the best result and the performance of GAC is still inferior to that of KNMC.



(a) (b)

Fig. 3. Comparison of NMC and KNMC for Crude Oil and Vowel

Since Iris has been used to choose the proper parameter settings for the clustering algorithm based on the noising method, in order to understand the performance of KNMC and NMC better, we here use Crude Oil and Vowel to show the iteration process. One should remember that the definition of an iteration is different from one algorithm to another. In NMC and KNMC, one iteration corresponds actually to one neighbor, while it corresponds to 20 neighbors for TSC, and to 6 neighbors for GAC. For SAC, one iteration also corresponds to one neighbor but this iteration is over after 100 iterations are performed at a specified annealing temperature. The clustering results of KNMC and NMC for Crude Oil and Vowel are shown as Figures 3(a) and (b), respectively. It is seen that applying the noising method to solve the clustering problem under consideration is feasible and effective. Moreover, in order to improve the performance of the clustering algorithm based on the noising method and accelerate the convergence speed, we introduce K-means operation to modulate the distribution of objects among clusters. To avoid getting stuck in local optima, we adopt the probability threshold to provide diverse neighboring solutions and explore the global optimal result. It is seen that KNMC equipped with K-means operation can attain better results for Crude Oil and Vowel much sooner than NMC. Compared with GAC and TSC, KNMC spends much less computational cost and achieve much better clustering results. Compared with that of SAC, the performance of KNMC is promising. The more important fact is that the computational cost of KNMC is much less than that of SAC. In this article, how to attain the optimal result as much as possible in the finite number of iterations by properly establishing the clustering algorithm is the main goal. This is our aim in future research work.

4 Conclusions

In this paper, we introduce the noising method to solve the clustering problem under the criterion of minimum sum of squares clustering, and develop two clustering approaches, NMC and KNMC. The choice of the algorithm parameters is extensively discussed, and performance comparisons between our methods and other techniques are conducted on experimental data sets. As a result, with the much less computational cost than GAC, TSC, and SAC, KNMC can get much better clustering results sooner than GAC and TSC, and obtain results close to those of SAC. In future, the estimation of the number of clusters has to be incorporated in NMC and KNMC, and different local search procedures should be tested within the noising method framework.

Acknowledgements

This research was partially supported by National Natural Science Foundation of China (#90104005, #60273049) and State Key Laboratory for Novel Software Technology at Nanjing University.

References

1. Jain, A.K., Dubes, R.: Algorithms for clustering data. Prentice-Hall, New Jersey (1988)
2. Selim, S.Z., Ismail, M.A.: K-means-type algorithm: generalized convergence theorem and characterization of local optimality. IEEE Transactions on Pattern Analysis and Machine Intelligence. 6 (1984) 81-87
3. Murthy, C.A., Chowdhury, N.: In search of optimal clusters using genetic algorithms. Pattern Recognition Letters. 17 (1996) 825-832
4. Al-sultan, K.S.: A tabu search approach to the clustering problem. Pattern Recognition. 28 (1995) 1443-1451
5. Bandyopadhyay, S., Maulik, U., Pakhira, M.K.: Clustering using simulated annealing with probabilisitc redistribution. International Journal of Pattern Recognition and Artificial Intelligence. 15(2001) 269-285
6. Charon, I., Hudry, O.: The noising method: a new method for combinatorial optimization. Operations Research Letters. 14(1993) 133-137
7. Charon, I., Hudry, O.: The noising method: a generalization of some metaheuristics. European Journal of Operational Research. 135(2001) 86-101
8. Chelouah, R., Siarry, P.: A hybrid method combining continuous tabu search and Nelder–Mead simplex algorithms for the global optimization of multiminima functions. European Journal of Operational Research. 161 (2005) 636-654
9. Fisher, R.A.: The use of multiple measurements in taxonomic problem. Annals of Eugenics. 7 (1936) 179-188
10. Johnson, R.A., Wichern, D.W.: Applied multivariate statistical analysis. Prentice-Hall, New Jersey (1982)
11. Pal, S.K., Majumder, D.D.: Fuzzy sets and decision making approaches in vowel and speaker recognition. IEEE Transactions on System, Man and Cybernetics. SMC-7 (1977) 625-629

Extracting the Representative Failure Executions via Clustering Analysis Based on Markov Profile Model

Chengying Mao and Yansheng Lu

College of Computer Science and Technology,
Huazhong University of Science and Technology,
430074 Wuhan, P. R. China
maochy@yeah.net

Abstract. During the debugging of a program to be released, it is unnecessary and impractical for developers to check every failure execution. How to extract the typical ones from the vast set of failure executions is very important for reducing the debugging efforts. In this paper, a revised Markov model used to depict program behaviors is presented firstly. Based on this model, the dissimilarity of two profile matrixes is also defined. After separating the failure executions and non-failure executions into two different subsets, iterative partition clustering and a sampling strategy called priority-ranked n-per-cluster are employed to extract representative failure executions. Finally, with the assistance of our prototype CppTest, we have performed experiment on five subject programs. The results show that the clustering and sampling techniques based on revised Markov model is more effective to find faults than Podgurski's method.

1 Introduction

In the activities of software development and maintenance, sufficient testing is generally employed according to some criteria to ensure the software's high reliability. Although many test suit minimization or optimization techniques have been proposed [1], testers still use proper redundancy for test cases to improve the trustiness of their testing. Therefore, the final test suit will be so huge that it brings great harassment to testing and debugging activities. As reported in [2], the efforts for detection and correction of faults consume about 50% to 80% of the development and maintenance budget. Testing can be implemented automatically in some ways, however, debugging mainly depends on manual operations. Hence, it is especially important for developers to classify and understand the behaviors caused by the executions of test cases, that is, perform clustering analysis on the executions of the program under test. In general, the failures in the same cluster are just caused by the same defect. Then developers can select one or more representative failure executions for debugging, so as to locate the faults. The strategy of debugging and fault localization via clustering analysis not only can significantly decrease the cost, but also has equal capability of revealing faults as before.

Owing to so enormous executions of program under test (with many corresponding failure executions), it is unnecessary and impractical for developers to debug the program for every execution that may cause a failure, because quite a few failure execu-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 217 – 224, 2005.

tions result from the same fault. There are a few ways to improve the debugging effort for programs. One of them is to reduce the research domain of executions that will be used for locating faults. At present, researchers deal with the work in two aspects: one is to select suspicious statements or blocks that might contain faults through extracting information from the program executions [3,4,5]. The other style has been seldom investigated, which is to search out a batch of representative ones from the enormous failure executions by data mining techniques, with which the following debugging can gain savings but will not lose the power of revealing faults. Data mining techniques have been proved highly effective to detect the program alarms [6]. Here, we use clustering analysis to mine the representative failure executions.

In this paper a model to represent each execution with the help of the collected test executions information is constructed firstly. We adopt the Markov model proposed by J. F. Bowring et al. [7], and do some revisions to facilitate the clustering analysis. As a consequence, an excellent clustering analysis method compared with A. Podgurski's is presented. In addition, a more practical sampling strategy for typical failure executions has also been addressed.

2 Related Work

Several previous researches have addressed issues closely related to failure classification and prioritization. Agrawal et al. adopt program slicing and dicing to facilitate the fault localization, and develop some prototypes, such as *xSlice* and *Spyder* [3,4]. Jones et al. [5] develop a visualization tool (*Tarantula*) of test information to assist fault localization. Their underlying target is program code, and they only use the collective features of test executions (called *program spectra* [8]).

Brun et al. [9] present a technique of generating machine learning model of program properties known to result in errors, and apply the model to classify and rank properties that may lead the users to errors. With different research purpose from ours, their method is used to find out the latent code errors that can't be effortlessly explored by all test cases. And to determine the properties of a program is not easy.

Bowring et al. [7] construct a Markov model to describe program behavior. This paper revises their model in a certain extent to facilitate the clustering analysis. The previous work that is closest in spirit and method to our work is that of Podgurski [10,11,12]. Their work uses clustering techniques, such as pattern classification and multivariate visualization, to classifying reported software failures so as to detect relative faults. There are three primary differences between our techniques and this previous work: (1) This paper adopts revised Markov models to describe the dynamic execution behaviors driven by test cases. (2) Podgurski's method mixes the failure executions with non-failure executions together. Perhaps this style takes out a successful execution which is not beneficial to debugging. (3) The priority-ranked sampling strategy in our work can prioritize the failure executions used for debugging preferably, which can speed up the process of correcting faults.

3 Markov Profile Model of Test Execution

First of all, some definitions about software structural testing are presented as follows:

Definition 1. *Control Flow Graph* (CFG) is a directed graph in which nodes represent statements or basic blocks[1] and edges represent the flow of control.

Definition 2. *Test Execution* of a test case is a record of program behavior scenario when the program is running under this test case.

Definition 3. *Execution Profile* of a test case is a representation of the control flow information in relevant test execution, including execution count of each edge at the run-time.

Program execution driven by an input is similar to the event transition features in the stochastic process. We draw an approximate assumption that the event transition between nodes holds the *Markov property*. So-called Markov property, i.e., the probability distribution of future states of a process depends only upon the current state: $P\{X_{k+1}=y_{k+1}|X_k=y_k,X_{k-1}=y_{k-1}, \ldots,X_0=y_0\}=P\{X_{k+1}=y_{k+1}|X_k=y_k\}=p_{ij}$. In a Markov model, the transition probability between the source node and target node can be defined as the relative execution frequency, or profile of the branch.

The construction of Markov model has the following three steps:

- Step 1. Prepare for modeling: the static analysis of program's CFG, to instrument program in proper sites and generate test cases for testing.
- Step 2. Perform the testing and recode the execution profile produced by each test case and the extrinsic behavior of program under test.
- Step 3. Transform the execution profile of each test case into a Markov model.

The third step is the core of constructing model. Here, we briefly demonstrate it by a sample (see Figure 1). As for the detailed discussion, please refer to the algorithm *BuildModel* in Reference [7]. The Markov model built from a program execution can be represented as a matrix, called *profile matrix*. Profile matrix is similar to the adjacent matrix of a CFG. We treat the execution count as the weight of edges between two nodes firstly, and the Markov model can be gained after employing the row-normalization on the matrix. The result is shown in Figure 2(a).

To facilitate the clustering analysis in latter stage, we carry out some revisions on the existing Markov model. First, the profile matrix is modified into a binary-like form: In the second step, we change the record fashion of execution count.

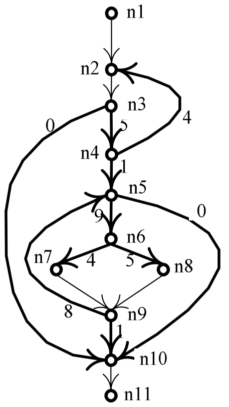
(1) For the ordinary branches, including the furcated edges such as *if-else*, *switch* etc., if one edge has been exercised, we set its execution count as 1, or else set as 0.

(2) For any *for* loop structure, if the edges in its domain can't be exercised, the execution counts of them are 0. If one edge has been exercised in times of the upper-bound of the *for* statement, its execution count can be set as 2, others as 1.

(3) For a *do-while* loop structure, if an edge in this domain has been come through once, whose execution count is 1. However, if it has been executed more then once (twice or more), its execution count is assigned as 2.

(4) For a *while* loop structure, it is similar to the condition (3).

[1] A basic block, is a sequence of consecutive statements or expressions containing no transfers of control except at the end, so that if one element of it is executed, all are.



Legend:
(1) The bold edges are instrumented branches;
(2) n2-n4 is a do-while structure, n5 is a for statement and its upper-bound is 9;
(3) Some nodes (such as n2, n7 and n8) have only one outgoing edge, so the transition probabilities from them to their target nodes are 1.

Fig. 1. Execution profile of a sample program segment

$$
\begin{array}{c|cccccccccc}
 & n1 & n2 & n3 & n4 & n5 & n6 & n7 & n8 & n9 & n10 \\
\hline
n1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n3 & 0 & 0 & 0 & 5/5 & 0 & 0 & 0 & 0 & 0 & 0/5 \\
n4 & 0 & 4/5 & 0 & 0 & 1/5 & 0 & 0 & 0 & 0 & 0 \\
n5 & 0 & 0 & 0 & 0 & 0 & 9/9 & 0 & 0 & 0 & 0/9 \\
n6 & 0 & 0 & 0 & 0 & 0 & 0 & 4/9 & 5/9 & 0 & 0 \\
n7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
n8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
n9 & 0 & 0 & 0 & 8/9 & 0 & 0 & 0 & 0 & 0 & 1/9 \\
n10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

(a) Profile matrix of the Markov model

$$
\begin{array}{c|cccccccccc}
 & n1 & n2 & n3 & n4 & n5 & n6 & n7 & n8 & n9 & n10 \\
\hline
n1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
n3 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0 \\
n4 & 0 & 2 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
n5 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 \\
n6 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\
n7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
n8 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
n9 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
n10 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{array}
$$

(b) Profile matrix of the revised Markov model

Fig. 2. Profile matrix vs. revised profile matrix

The revised profile matrix (no need of row-normalization) is shown as Figure 2(b).

On the other hand, the information of behavior label (*b* for short) produced by an execution can be extended. Besides the execution result, we add the reasons of program termination into it. So $b=(f_1, f_2)$, where f_1={*"pass", "failure"*} and f_2={*"correct return value", "incorrect return value", "memory out",…*}. According to the actual project instance, the label can be customized as *n*-dimensional vector $b=(f_1, f_2, \ldots, f_n)$.

4 Clustering Analysis and Sampling for Test Executions

4.1 Clustering Analysis

The benefit of performing the clustering analysis on program debugging is as follows: Debuggers can sample a small set of representative failures from those clusters instead of checking every failure execution.

In this paper, the dissimilarity is mainly scaled by calculating the discrepancy between any two execution profiles.

Definition 4. *Profile Dissimilarity* is a metric of comparing the codes exercised by two test cases, i.e., the degree of difference between two program paths traversed by these test cases.

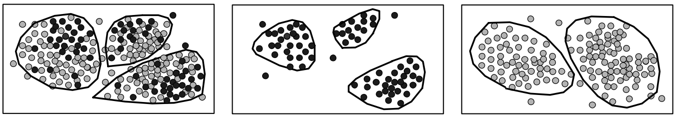
Here, the Manhattan distance is introduced to calculate the dissimilarity based on our revised Markov model. We call it binary-like metric, expressed as formula (1), where *n* is the dimension of revised profile matrix, *M* and *M'* are two profile matrixes of test executions.

$$dsm(M, M') = \sum_{i=1}^{n} \sum_{j=1}^{n} \left| m_{ij} - m_{ij}' \right| \tag{1}$$

We also present another more reasonable metric, calculating the dissimilarity from a mixed profile matrix. It is achieved by concatenating the previous profile matrix and the revised matrix. This mixed matrix possesses the merits of two former matrixes: it not only reflects the execution frequency of each edge but also emphasizes edge's feature of being traversed. Based on this form of mixed matrix, the dissimilarity between tow executions (*X* and *X'*) is calculated as formula (2).

$$dsm(X, X') = \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{2n} \left| x_{ij} - x_{ij}' \right| \tag{2}$$

The clustering analysis on failure executions can be carried out in the following way: Firstly, all test executions can be classified into failure and non-failure executions. Aiming at the failure execution subset, it can be initially divided into *k* clusters according to the information of the second factor f_2 in each behavior label, where *k* is chosen beforehand. Then, the quality of the partitioning is improved iteratively by reassigning test executions to different clusters. When the reformative increment decreases to a small value ε, we terminate this iterative process. The result is the partition of failure executions. If test cases are not so sufficient for uncovering the faults in program, it's necessary to classify the non-failure executions. In this case, the common *divisive hierarchical clustering* method [13] is adopted, and the size of each cluster of non-failure executions can be enlarged properly. Obviously, our process of clustering analysis is greatly different from Podgurski's. The comparison is schematically demonstrated in the following figure.



(a) Clustering method of Podgurski

(b1) Clustering for failure executions

(b2) Clustering for non-failure executions

(b) Our clustering analysis method (two steps)

Fig. 3. An exhibitive diagram of two clustering methods. The gray points represent non-failure executions, and black points represent the failure executions. The distances between the points approximate the dissimilarities. Note that the distributions of points' positions are different between (a) and (b),because we adopt another dissimilarity measure compared with Podgurski's

Determining the number (k) of partitions to be constructed is a crucial step in clustering analysis. Rather than used a fixed number, the cluster count k is set as the proportion of the number of test executions. For instance, the 5%, 10% and 20% of the executions can be treated as candidate cluster counts. In general, the cluster count (k_f) of failure executions is larger than the count (k_{nf}) of non-failure executions, i.e., $k_{nf}<k_f$.

4.2 Sampling Strategy

Set $E_f=\{C1,C2, ..., Ck_f\}$ as the status of the clusters of failure executions. Similarly, the clusters of non-failure executions are represented as $E_{nf}=\{L1, L2, ..., Lk_{nf}\}$. Before the sampling, these two subsets are prioritized according to the cardinalities of the clusters in them respectively. Therefore, two new permutations are formed as following: $E_f'=\{C1',C2', ..., Ck_f'\}$ ($|C1'|>|C2'|>...>|Ck_f'|$) and $E_{nf}'=\{L1',L2', ..., Lk_{nf}'\}$ ($|L1'|>|L2'| >...>|Lk_{nf}'|$). Here, we adopt a *priority-ranked n-per-cluster* strategy for sampling representative failure executions, where n is a number from 1 to $|Ck_f'|$ (or $|Lk_{nf}'|$). During the sampling from the subset of failure executions (i.e., E_f'), the selection starts with the largest cluster and progresses to the smaller ones. If the sampled executions don't conform to the debug requirements, we will employ the next round sampling until the requirements is satisfied or the round number of selection reaches the previously fixed n.

For the case of insufficient testing, in order to uncover all faults by debugging, some non-failure executions should be filtered to supplement the need of debugging. Certainly, debuggers use these non-failure executions only when they have debugged all sampled failure executions but failed to reveal defects any more. During the sampling of non-failure executions, one-per-cluster sampling strategy is employed, i.e., it selects a random execution from each cluster without replacement.

5 Preliminary Empirical Study

The goal of our experiment is to validate the efficiency of our clustering analysis method of failure executions. The experiment mainly includes the following steps: (1) Performing the mutant operations on the subject programs, here, each version of program is fed with more than one fault. (2) Preparing a large set of test cases. (3) Automatically testing the subject programs and recording the trajectory of each test case and the behavior of program termination. Here, we use the prototype *CppTest* implemented by the software testing group in our lab [14]. (4) Classifying and sampling the failure execution for debugging activity via the Podgurski's clustering analysis and ours, respectively.

In our experiment, five programs are treated as subject programs (see Table 1), and the latter three programs are all downloaded from a VisualC++ education Website (www.vccode.com). The test suit of these programs is generated by the statistical testing technique [14].

When our method is used for clustering and sampling, we set k_f as 10%. Because the test cases of the last program are not plentiful, so we set k_{nf} =5% for it. The sampling strategy used in this experiment is priority-ranked two-per-cluster sampling. For the sake of contrast, we also implement Podgurski's clustering method, and the k is also assigned to 10%. The sampling strategy in this case is two-per-cluster sampling.

Table 1. Overview of the subject programs

Program	Description	Language	LOC	#Test cases	#Mutants	#Failures
Sort[3,15]	a program for sorting order	C	541	997	20	653
Account [16]	an account management program in bank	C++	631	1072	25	431
Chat	a chat program	C++	2102	2231	79	1581
CDPlayer	a music player	C++	2185	1640	61	987
Wordpad	a simplified words processing software	C++	6307	3524	152	2846

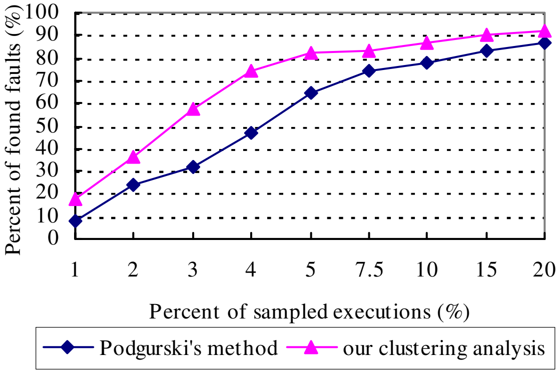


Fig. 4. Comparison of efficiencies of two clustering methods

In this paper, we use the formula (3) to measure the efficiency. The result of contrast experiment is shown in Figure 4, where the curves are drawn according to the average values of the filtering efficiencies of five subject programs.

$$Eff = \frac{\textit{the percent of failures found}}{\textit{the ratio of the sampled executions to all failure executions}} \tag{3}$$

As illustrated in the above figure, we can draw a conclusion that our clustering and sampling techniques for debugging based on revised Markov profile model achieve some effective improvement to Podgurski’s clustering method.

6 Conclusions and Future Work

Through describing each execution by binary-like profile matrix, a revised Markov model used to depict program behaviors is presented. After separating the failure executions and non-failure executions into two different subsets, we employ iterative partition clustering to analyze them respectively. In addition, the sampling strategy called priority-ranked *n*-per-cluster is utilized to extract representative failure executions (maybe including a few non-failure executions). Finally, with the assistance of our prototype CppTest, we have performed a contrast experiment on five subject programs. The results show that the clustering and sampling techniques based on revised Markov model is more effective to find faults than Podgurski's method.

A critical review of our methods highlights some directions for future research: (1) the *hidden Markov models* (HMMs) can be explored to extend the current behavior-modeling technique. (2) Apart from the control flow information, the investigation of adding the indications provided by data flow testing to our clustering analysis technique is also considerable.

References

1. Rothermel, G., Untch, R. H., Harrold, M. J.: Prioritizing Test Cases for Regression Testing, IEEE Transaction on Software Engineering. IEEE Press, New York (2001) 929-948
2. Collofello, J. S., Woodfield, S. N.: Evaluating the Effectiveness of Reliability-Assurance Techniques. Journal of Systems and Software. Elsevier Science, New York (1989) 191-195
3. Agrawal, H., Horgan, J. R., London, S., Wong, W. E.: Fault Localization using Execution Slices and Dataflow Tests. In: Proc. of IEEE Software Reliability Engineering. IEEE Press, New York (1995) 143-151
4. DeMillo, R. A., Pan, H., Spafford, E. H.: Critical Slicing for Software Fault Localization, In: Proc. of ISSTA'96. ACM Press, New York (1996) 121-134
5. Jones, J. A., Harrold, M. J., Stasko, J.: Visualization of Test Information to Assist Fault Localization. In: Proc. of ICSE'02. ACM Press, New York (2002) 467-477
6. Julisch, K., Dacier, Marc: Mining Intrusion Detection Alarms for Actionable Knowledge. In: Proc. of KDD'02. ACM Press, New York (2002) 366-375
7. Bowring, J. F., Rehg, J. M., Harrold, M. J.: Active Learning for Automatic Classification of Software Behavior. In: Proc. of ISSTA'04. ACM Press, New York (2004) 195-205
8. Reps, T., Ball, T., Das, M., Larus, J.: The Use of Program Profiling for Software Maintenance with Applications to the Year 2000 Problem. In: Proc. of the 6th European Software Engineering Conference. IEEE Press, New York (1997) 432-449
9. Brun, Y., Ernst, M. D.: Finding Latent Code Errors via Machine Learning over Program Executions. In: Proc. of ICSE'04. ACM Press, New York (2004) 480-490
10. Dickinson, W., Leon, D., Podgurski, A.: Finding Failures by Cluster Analysis of Execution Profiles. In: Proc. of ICSE'01. IEEE Press, New York (2001) 339-348
11. Podgurski, A., Leon, D., Francis, P., et al.: Automated Support for Classifying Software Failure Reports. In: Proc. of ICSE'03. ACM Press, New York (2003) 465-475
12. Dickinson, W., Leon, D., Podgurski, A.: Pursuing Failure: the Distribution of Program Failures in a Profile Space. In: Proc. of the 8th European Software Engineering Conference. ACM Press, New York (2001) 246-255
13. Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, Inc., San Fransisco, CA USA (2001)
14. Mao, C. Y., Lu, Y. S.: Employing Statistical Testing Techniques on Object-Oriented Classes. In: Proc. of 2nd Excellent Ph.D. Candidates Annual Forum of China Association for Science and Technology. CAST Press, Beijing China (2004) 402-409 (in Chinese)
15. Wong, W. E., Horgan, J. R., London, S., Mathur, A. P.: Effect of Test Set Minimization on Fault Detection Effectiveness. Software-Practice and Experience. John Wiley & Sons, Inc., Hoboken, NJ (1998) 347-369
16. Chen, H. Y., Tse, T.H., Chen, T. Y.: TACCLE: A Methodology for Object-Oriented Software Testing at the Class and Cluster Levels. ACM Transactions on Software Engineering and Methodology. ACM Press, New York (2001) 56-109

Improvement on the Approximation Bound for Fuzzy-Neural Networks Clustering Method with Gaussian Membership Function

Weimin Ma[1,2] and Guoqing Chen[1]

[1] School of Economics and Management,
Tsinghua University, Beijing, 100084, P.R. China
{mawm, chengq}@em.tsinghua.edu.cn
[2] School of Economics and Management,
Xi'an Institute of Technology,
Xi'an, Shaanxi Province, 710032, P.R. China

Abstract. A great deal of research has been devoted in recent years to the designing Fuzzy-Neural Networks (FNN) from input-output data. And some works were also done to analyze the performance of some methods from a rigorous mathematical point of view. In this paper, a new approximation bound for the clustering method, which is employed to design the FNN with the Gaussian Membership Function, is established. It is an improvement of the previous result in which the related approximation bound was somewhat complex. The detailed formulas of the error bound between the nonlinear function to be approximated and the FNN system designed based on the input-output data are derived.

1 Introduction

FNN systems are hybrid systems that combine the theories of fuzzy logic and neural networks. Designing the FNN system based on the input-output data is a very important problem [1, 2, 3]. Some Universal approximation capabilities for a broad range of neural network topologies have been established by researchers like Cybenko [4], Ito [5], and T.P.Chen [6]. Their work concentrated on the question of denseness. Some Approximation Accuracies of FNN have also been established in [7]. More results concerning the approximation of Neural Network can be found in [8, 9, 10, 11, 12, 13].

In paper [3], an approach so called Nearest Neighborhood Clustering was introduced for training of Fuzzy Logic System. And this kind of system was proved to be universal approximation in [2]. The relevant approximation accuracy with $\left[\overline{d}_x + \left(1+\frac{\sqrt{n}}{n}\right) r + \frac{2\sqrt{n}\sigma^2}{r^n}\left(2r+\sqrt{n\pi}\sigma\right)^{n-1}\right] \cdot \sum_{i=1}^{n} \left\| \frac{\partial f}{\partial x_i} \right\|_\infty$ of the clustering method with Triangular Membership Function and Gaussian Membership Function is obtained in paper [7]. In this paper, we obtain a new upper bound, $\left(r + (2^n\overline{M}+1)\overline{d}_x\right) \sum_{i=1}^{n} \left\| \frac{\partial f}{\partial x_i} \right\|_\infty$ by taking advantage of the similar

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 225–231, 2005.

techniques of paper [7], for the approximation by using the clustering method with Gaussian Membership Function.

Comparing to the work of paper [7], some improvements are made in the proof of the main theorem of this paper to get the better approximation bound. First of all, we modify the method of dividing space to prove the second case concerning the equations (15) and (16). Following that, the property of decreasing monotonically of some functions are used to prove instead of the technique concerning the lemma 1 and 2 in that paper. In addition, some corrections are made for that paper. Base on above improvement, we get a new approximation bound for the clustering method with Gaussian Membership Function. The notations of this paper is similar with that paper.

The remainder of this paper is organized as follows. In section 2, for the integrality and illustration of parameters, we briefly introduce the clustering method. Section 3 presents the main result about the approximation bound as well as the last section makes some concluding remarks.

2 Clustering Method with Gaussian Membership Function

Before introducing the main results, we firstly introduce some basic knowledge on the designing FNN systems with clustering method, which was proposed in [2] and also be found in [7]. Given the input-out data pairs

$$(x_0^q, y_0^q), \quad q = 1, 2, \ldots \tag{1}$$

where $x_0^q \in U = [\alpha_1, \beta_1] \times \ldots \times [\alpha_n, \beta_n] \subset R^n$ and $y_0^q \in U_y = [\alpha_y, \beta_y] \subset R$. If the data are assumed to be generated by an unknown nonlinear function $y = f(x)$, the clustering method in [2] can help us to design a FNN system to approximate the function $f(x)$. For convenience to illustrate the main result of this paper, we describe this method in a brief way as follows.

Step 1. To begin with the first input-output pair (x_0^1, y_0^1), select a radius parameter r, establish a cluster center with letting $x_c^1 = x_0^1$, and set $y_c^1(1) = y_0^1$, $B^1(1) = 1$.

Step 2. For the kth input-out pair (x_0^k, x_0^k), $k = 2, \ldots$, suppose there are M clusters with centers at $x_c^1, x_c^2, \ldots, x_c^M$. Find the nearest cluster center $x_c^{l_k}$ to x_0^k to satisfy

$$|x_0^k - x_c^{l_k}| = \min_l |x_0^k - x_c^l|, \quad l = 1, 2, \ldots, M. \tag{2}$$

Then there are two cases

Case 1. If $|x_0^k - x_c^{l_k}| \geq r$, establish x_o^k as a new cluster center with $x_c^{M+1} = x_0^k$, $y_c^{M+1}(k) = y_0^k$ and $B^{M+1}(k) = 1$, and keep $y_c^l(k) = y_c^l(k-1)$ and $B^l(k) = B^l(k-1)$ for any l.

Case 2. If $|x_0^k - x_c^{l_k}| < r$, do the following:

$$y_c^{l_k}(k) = \frac{y_c^{l_k}(k-1)B^{l_k}(k-1) + y_0^k}{B^{l_k}(k-1)+1} \tag{3}$$

$$B^{l_k}(k) = B^{l_k}(k-1) + 1 \tag{4}$$

and meanwhile set

$$y_c^l(k) = y_c^l(k-1), \quad B^l(k) = B^l(k-1), \text{ for } l \neq l_k. \tag{5}$$

Step 3. Then the FNN system can be constructed as:

$$\begin{aligned}\hat{f}_k(x) &= \frac{\sum\limits_{l=1}^{\overline{M}} \left[y_c^l(k) \cdot \exp\left(-\frac{|x - x_c^l|^2}{\sigma^2} \right) \right]}{\sum\limits_{l=1}^{\overline{M}} \exp\left(-\frac{|x - x_c^l|^2}{\sigma^2} \right)} \\ &= \frac{\sum\limits_{l=1}^{\overline{M}} \left[y_c^l(k) \cdot \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2} \right) \right]}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2} \right)}\end{aligned} \tag{6}$$

where $\overline{M} = M + 1$ for case 1 and $\overline{M} = M$ for case 2.
Step 4. Repeat by going to Step 2 with $k = k + 1$.

The above FNN system is constructed using singleton fuzzier, product inference engine and center-average defuzzifier, as detailed in [2]. Some intensive simulation results concerning this method for various problems can also be found in that paper. This paper concentrates on the approximation bound for this method with Gaussian membership function.

3 Approximation Bound with Gaussian Membership Function

The following theorem gives the approximation bound of FNN system $\hat{f}_k(x)$ of (6) which is constructed by using the clustering method with Gaussian membership function.

Theorem 1. [illegible] $f(x)$ [illegible] U [illegible] ( ) [illegible] $\hat{f}_k(x)$ [illegible] ( )

$$|f(x) - \hat{f}_k(x)| \leq \left(r + (2^n \cdot \overline{M} + 1) \cdot \overline{d}_x \right) \cdot \sum_{i=1}^{n} \left\| \frac{\partial f}{\partial x_i} \right\|_\infty \tag{7}$$

[illegible] $\|\cdot\|_\infty$ [illegible] $\|d(x)\|_\infty = \sup_{x\in U}|d(x)|$, $\overline{d}_x = \max\left\{d_x, \frac{\sigma}{\sqrt{2}}\right\}$ [illegible] d_x [illegible] x [illegible]

$$d_x = \min_l |x - x_c^l| = |x - x_c^{l_x}| \tag{8}$$

[illegible] From (6) we have

$$|f(x) - \hat{f}_k(x)| \le \frac{\sum\limits_{l=1}^{\overline{M}} \left[|f(x) - y_c^l(k)| \cdot \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right]}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \tag{9}$$

Paper [7] obtain the following result when the relevant approximation bound for the clustering method with triangular membership function was discussed:

$$|f(x) - y_c^l(k)| \le \sum_{i=1}^{n} \left(\left\| \frac{\partial f}{\partial x_i} \right\|_\infty \cdot \left(|x_i - x_{c,i}^l| + r \right) \right) \tag{10}$$

Combining the (9) and (10), we have

$$\begin{aligned} |f(x) - \hat{f}_k(x)| &\le \frac{\sum\limits_{l=1}^{\overline{M}} \left[\left(\left\| \frac{\partial f}{\partial x_i} \right\|_\infty \cdot \left(|x_i - x_{c,i}^l| + r \right) \right) \cdot \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right]}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \\ &\le \sum_{i=1}^{n} \left\{ \left\| \frac{\partial f}{\partial x_i} \right\|_\infty \cdot \left[r + \frac{\sum\limits_{l=1}^{\overline{M}} \left(|x_i - x_{c,i}^l| \cdot \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right)}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \right] \right\} \end{aligned} \tag{11}$$

Now, we just focus on analyzing the term

$$\frac{\sum\limits_{l=1}^{\overline{M}} \left(|x_i - x_{c,i}^l| \cdot \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right)}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)}$$

on the right-hand side of (11). We only consider the case $i = 1$ and the proof remains the same for $i = 2, \ldots, n$.

Employing the similar method in [7], given any point $x = (x_1, x_2, \ldots, x_n) \in U$, we divide space U in to 2^n areas and define some sets concerning the cluster center as follows:

$$\begin{aligned}
&U_1^x = \{\overline{x} \in U : \overline{x}_1 - x_1 \geq 0, \ldots, \overline{x}_n - x_n \geq 0\} \\
&U_2^x = \{\overline{x} \in U : \overline{x}_1 - x_1 \geq 0, \ldots, \overline{x}_n - x_n < 0\} \\
&\quad \cdots \\
&U_{2^n-1}^x = \{\overline{x} \in U : \overline{x}_1 - x_1 < 0, \ldots, \overline{x}_n - x_n \geq 0\} \\
&U_{2^n}^x = \{\overline{x} \in U : \overline{x}_1 - x_1 < 0, \ldots, \overline{x}_n - x_n < 0\} \qquad (12)
\end{aligned}$$

And define some sets concerning the cluster centers

$$\begin{aligned}
V^x &= \{\overline{x} \in U : |\overline{x}_1 - x_1| < \overline{d}_x\}, \\
\overline{V}^x &= \{\overline{x} \in U : |\overline{x}_1 - x_1| \geq \overline{d}_x\}, \\
V_m^x &= \overline{V}^x \cap U_m^x, \qquad (m = 1, \ldots, 2^n). \qquad (13)
\end{aligned}$$

Apparently, there are two cases need to consider.

Case 1: $x_c^l \in V^x$, which indicates $|x_{c,1}^l - x_1| < \overline{d}_x$, we have

$$\begin{aligned}
&\frac{\sum\limits_{x_c^l \in V^x} \left(|x_1 - x_{c,1}^l| \cdot \prod\limits_{j=1}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right)}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \\
&\leq \frac{\overline{d}_x \cdot \sum\limits_{x_c^l \in V^x} \prod\limits_{j=1}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)}{\sum\limits_{l=1}^{\overline{M}} \prod\limits_{j=1}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \\
&\leq \overline{d}_x \qquad (14)
\end{aligned}$$

Case 2: $x_c^l \in \overline{V}^x = \bigcup_{m=1}^{2^n} V^x$. We only consider the case $x_c^l \in V_1^x$. For the cases $x_c^l \in V_2^x, \ldots, V_{2^n}^x$, the same result can be obtained. For any l that satisfied to $x_c^l \in V_1^x$, according to the definition of V_1^x, we have

$$x_{c,1}^l - x_1 \geq \overline{d}_x \ \text{ and } \ x_{c,j}^l - x_j \geq 0, \quad j = 1, \ldots, n \qquad (15)$$

From (12), (13) and (15), we have

$$\begin{aligned}
&\sum_{x_c^l \in V_1^x} \left[|x_1 - x_{c,1}^l| \cdot \prod_{j=1}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right] \\
&\quad = \sum_{x_c^l \in V_1^x} \left[|x_1 - x_{c,1}^l| \cdot \exp\left(-\frac{|x_1 - x_{c,1}^l|^2}{\sigma^2}\right) \cdot \prod_{j=2}^n \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)\right]
\end{aligned}$$

$$\begin{aligned}
&\leq \sum_{x_c^l \in V_1^x} \left[\overline{d}_x \cdot \exp\left(-\frac{\overline{d}_x^2}{\sigma^2}\right) \cdot \prod_{j=2}^{n} \exp\left(-\frac{0^2}{\sigma^2}\right) \right] \\
&= \overline{d}_x \cdot \exp\left(-\frac{\overline{d}_x^2}{\sigma^2}\right) \cdot \sum_{x_c^l \in V_1^x} 1 \\
&\leq \overline{M} \cdot \overline{d}_x \cdot e^{\left(-\frac{\overline{d}_x^2}{\sigma^2}\right)} \qquad (16)
\end{aligned}$$

The inequalities of (16) holds for (15) and the following reason: $g(x) = x \cdot e^{-\frac{x^2}{\sigma^2}}$ decreases monotonically in $[(\sigma/\sqrt{2}), \infty)$ because

$$\frac{d(g(x))}{d(x)} = \frac{\sigma^2 - 2 \cdot x^2}{\sigma^2} \cdot e^{-\frac{x^2}{\sigma^2}} \leq 0 \qquad (17)$$

Considering cases $x_c^l \in V_2^x, \ldots, V_{2^n}^x$, we have

$$\sum_{x_c^l \in \overline{V}^x} \left[|x_1 - x_{c,1}^l| \cdot \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right) \right] \leq 2^n \cdot \overline{d}_x \cdot \overline{M} \cdot e^{\left(-\frac{\overline{d}_x^2}{\sigma^2}\right)} \qquad (18)$$

On the other hand, it follows from (8) that

$$\sum_{l=1}^{\overline{M}} \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right) = \sum_{l=1}^{\overline{M}} \exp\left(-\frac{|x - x_c^l|^2}{\sigma^2}\right) \geq e^{\left(-\frac{d_x^2}{\sigma^2}\right)} \qquad (19)$$

From (18) and (19), we have

$$\begin{aligned}
&\frac{\sum_{x_c^l \in \overline{V}^x} \left[|x_1 - x_{c,1}^l| \cdot \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right) \right]}{\sum_{l=1}^{\overline{M}} \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \\
&\leq 2^n \cdot \overline{d}_x \cdot \overline{M} \cdot e^{\left(-\frac{\overline{d}_x^2 - d_x^2}{\sigma^2}\right)} \\
&\leq 2^n \cdot \overline{d}_x \cdot \overline{M} \qquad (20)
\end{aligned}$$

The last inequality holds for the definitions of $\overline{d}_x$ and d_x and the decreases monotonically of function e^{-x}.

Combining (13), (14) and (20), it can be shown that

$$\frac{\sum_{l=1}^{\overline{M}} \left[|x_1 - x_{c,1}^l| \cdot \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right) \right]}{\sum_{l=1}^{\overline{M}} \prod_{j=1}^{n} \exp\left(-\frac{|x_j - x_{c,j}^l|^2}{\sigma^2}\right)} \leq (2^n \cdot \overline{M} + 1) \cdot \overline{d}_x \qquad (21)$$

From (11) and (21), we get the desired result. □

4 Concluding Remarks

In this paper, the new upper bound, $\left(r + (2^n \cdot \overline{M} + 1) \cdot \overline{d}_x\right) \cdot \sum_{i=1}^{n} \left\| \frac{\partial f}{\partial x_i} \right\|_\infty$, concerning the approximation bound of clustering method with Gaussian Membership Function is proved in a rigorous mathematical way. The techniques employed in the proof of the theorem are expected to be used to obtain or improve other approximation bound of other methods of FNN.

Acknowledgements. The work was partly supported by the National Natural Science Foundation of China (70401006, 70231010) and the China Postdoctoral Science Foundation (2003034014).

References

1. Leng, G., Prasad, G. and McGinnity, T. M.: An On-line Algorithm for Creating Self-organizing Fuzzy Neural Networks Neural Network. **17** (2004) 1477-1493
2. Wang, L.X. and Mendel, J.M., W.: Fuzzy Basis Functions, Universal Approximation, and orthogonal least squares learning. IEEE Trans. Neural Network. **3** (1992) 807–814
3. Wang, L.X.: Training of Fuzzy Logic System Using Nearest Neighborhood Clustering. Proc. 1993 IEEE Int. Conf. Fuzzy Syst. San Francisco, CA, (1993) 13–17
4. Cybenko, G.: Approximation by Superpositions of a Sigmoidal Function. Math. Contr. Signals. Syst.. **2** (1989) 303–314
5. Ito, Y.: Approximation of Continuous Functions on R^d by Linear Combination of Shifted Rotations of Sigmoid Function with and without Scaling Neural Networks. Neural Networks. **5** (1992) 105–115
6. Chen, T.P., Chen, H.: Approximation Capability to Functions of Several Variables, Nonlinear Functions, and Operators by Radial Function Neural Networks. IEEE Trans. Neural Networks. **6** (1995) 904–910
7. Wang, L.X. and Chen, W.: Approximation Accuracy of Some Neuro-Fuzzy Approches. IEEE Trans. Fuzzy Systems. **8** (2000) 470–478
8. Maiorov, V., Meir, R.S.: Approximation Bounds for Smooth Functions in $C(R^d)$ by Neural and Mixture Networks. IEEE Trans. Neural Networks. **9** (1998) 969–978
9. Burger, M., Neubauer, A.: Error Bounds for Approximation with Neurnal Networks. J. Approx. Theory. **112** (2001) 235–250
10. Wang, J.J., Xu, Z.B., Xu, W.J.: Approximation Bounds by Neural Networks in L^p_ω. Proc. of 1st International Symposium on Neural Networks, F.Yin, J.Wang, and C.Guo (Eds.): ISSN 2004, LNCS **3173** (2004) 1–6
11. Barron, A.R.: Universal Approximation Bound for Superpositions of a Sigmoidal Function. IEEE Trans. Inform. Theory. **39** (1993) 930–945
12. Mhaskar, H.N.: Neural Networks for Optimal Approximation for Smooth and Analytic Functions. Neural Comput.. **8** (1996) 164–177
13. Kurkova, V., Sanguineti, M.: Comparison of Worst Case Errors in Linear and Neural Network Approximation. IEEE Trans. Inform. Theory. **48** (2002) 264–275

Optimal Fuzzy Modeling Based on Minimum Cluster Volume*

Can Yang and Jun Meng

College of Electrical Engineering, Zhejiang University,
Hangzhou 310027, P. R. China
yangcan_1220@163.com, junmeng@zju.edu.cn

Abstract. This paper proposes a new fuzzy modeling method, which involves the Minimum Cluster Volume clustering algorithm. The cluster centers founded are naturally considered to be the centers of Gaussian membership functions. Covariance matrix obtained from the result of cluster method is made use to estimate the parameters σ for Gaussian membership functions. A direct result of this method are compared in our simulations with published methods, which indicate that our method is powerful so that it solves the multi-dimension problems more accurately even with less complexity of our fuzzy model structure.

1 Introduction

Clustering of numerical data forms the basis for many classification and system modeling algorithms. The purpose of clustering is to obtaind natural groupings of data from a large data set, producing a concise representation of a system's behavior. Fuzzy clustering algorithms are less prone to local minima than crisp clustering algorithm since they make soft decisions via memberships. The Fuzzy c-Means clustering (FCM) algorithm [1] is effective when all clusters are roughly spherical with similar volumes. Other algorithms have been developed to cluster data, taking into account clusters with different shapes and positions. For example, the Gustafson-Kessel (GK) algorithm [2] tries to accommodate ellipsoidal cluster by extending clusters along their longest axis. However, this often leads to a wrong clustering result when clusters are close to each other, because the extensions can grab points belonging to other clusters. To overcome this, a Minimum cluster Volume algorithm (MCV) [3] is used for fuzzy modeling in this paper.

Yager and Filev [4] developed the mountain-clustering algorithm to obtain the structure of fuzzy model that is used to be the initial model for back propagation tuning algorithm. Although this method is simple and effective, the computation grows exponentially with the increase of the dimension.

* The Project Supported by Zhejiang Provincial Natural Science Foundation of China. No.601112.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 232 – 239, 2005.

Chiu [5] developed a subclustering method to obtain the structure of initial fuzzy model and the consequent parameters of the model were estimated by least squares method. Although it is an efficient method, we find out that the initial shapes of membership functions are not always the most appropriate, because the Gauss memberships on the same dimension have the same σ value in (1) but that is not realistic.

$$\mu_i = e^{-(\frac{x-c_i}{\sqrt{2}\sigma})^2} \tag{1}$$

Further more, Babuska [6] has done a lot of work about cluster modeling. He takes advantage of GK algorithm to obtain a fuzzy partition matrix. To obtain parameterized membership functions, the fuzzy partition matrix must be projected point-wise to each dimension and smoothed by a filter, so the trapezoidal and exponential membership functions can be determined by some optimization method such as Nelder-Mead Simplex Method [7]. The consequent parameters are identified by least squares method. Though a quite good fuzzy model can be obtained, his method involves nonlinear search or function approximation so that the performance will be affected by its initial state.

In this paper, we take the advantage of MCV and develop a more reasonable approach to obtain the parameters to describe membership functions. The parameterization process does not involve any nonlinear search so that it is effective and efficient. With this process, we can build an initial structure of fuzzy model, which can further be trained by ANFIS [8] if necessary.

2 Minimum Cluster Volume (MCV) Algorithm

Although MCV is less sensitive to initialization than EM [9] as Krishnapuram and Kim [3] pointed out, local minima still exist due to the complexity of MCV cost surface. To overcome this, FCM algorithm [1] is used for initializing MCV center location, instead of the more traditional method of random selection of the initial centers.

Basically, the MCV algorithm is a hybrid of fuzzy and hard clustering algorithm. The first is a simple fuzzy membership update calculation, while the second is a hard membership update calculation. The membership update calculation is chosen based on the value of the Mahalanobis distance MD_{ji}. The fuzzy update process ($MD_{ji} > L \forall i$) is performed using the following calculation that is the same with FCM update law

$$u_{ij} = \frac{(D_{ij})^{1/(1-m)}}{\sum_{r=1}^{C} (D_{rj})^{1/(1-m)}} \tag{2}$$

while with the hard update process, the membership can be set only to 1 or 0. Note that adding the hard clustering procedure allows the points close to the center to achieve a full membership within that cluster, which increases their effects on the cluster's volume. With these two update procedures in hand, we proceed as in the FCM case with Picard style iteration, alternating between calculating membership values and center locations until the maximum change in membership values is less than a defined value.

Given the two-dimensional data set, Fig.1 shows the results of MCV, FCM and GK algorithms. Fig.1 shows the advantage of MCV compared with FCM and GK, as its partition is more similar to those done by human being. So MCV is used as basic clustering method for the fuzzy modeling in this paper.

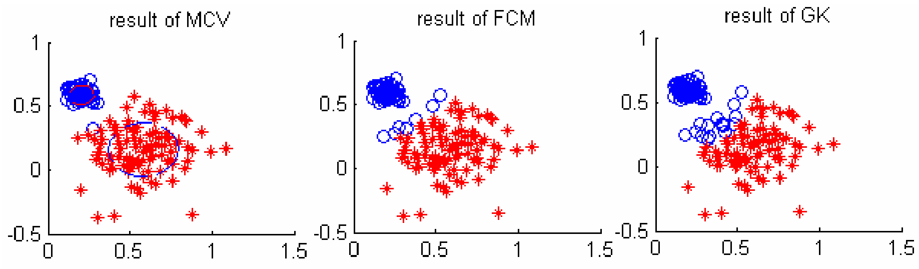


Fig. 1. Comparison of MCV, FCM, GK

3 Estimation for Membership Functions

As what we can expect from the result of MCV algorithm, the centers *V*, the fuzzy partition matrix U and the covariance matrix *C* for the center *V* can be written as

$$[V,U,C]=MCV(cluster\_data\_set,\ cluster\_n) \tag{3}$$

where *cluster_data_set* is the data set to be clustered and *cluster_n* is the cluster number.

Obviously, the cluster centers can be chosen to be the centers of Gaussian membership functions as others do [4, 5, 6]. The left problem is how to determine parameter σ for Gaussian membership function. A series of experiments have been done by different approaches, such as Nelder-Mead Simplex Method [7] and Genic algorithm [10, 11].

Here we propose a new method that could deal with general cases based on the root of diagonal elements of each cluster's covariance matrix. The method we propose is of particular importance for its accuracy and effectiveness, as well as no need for any nonlinear search or fit.

The covariance matrix for cluster *i* is defined as

$$C_{fi} = \frac{\sum_{j=1}^{N} u_{ij}^{m}(x_j - v_i)(x_j - v_i)^T}{\sum_{j=1}^{N} u_{ij}^{m}} \tag{4}$$

where x_j is *j-th* data in data set, v_i is the *i-th* center, u_{ij} is defined by (2), and *m* is the fuzziness coefficient which is set to be 2 here.

Cluster Covariance matrix provides information about the shape of the cluster. Take 2-dimensional random variable as an example for simplification without losing the generality. Suppose *(X, Y)* subject to normal distribution, so that the covariance matrix for *(X, Y)* can be written as:

$$C = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} = \begin{bmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{bmatrix} \tag{5}$$

Here, σ_1, σ_2 is the standard deviation of random variable *X* and *Y*, respectively, and ρ is the related coefficient of them. As it is well known, the standard deviation is a critical description of the distribution of the data set. So it is reasonable to use the standard deviation to estimate the parameters σ for Gaussian membership functions. So parameter σ_{ik} for *i*-th center and *k*-th dimension can be calculated as:

$$\sigma_{ik} = \sqrt{Diag(C_{fi})} \quad 1 \le k \le d \tag{6}$$

where *d* is the dimension of the data set.

These σ values provide the shapes of each membership for each center on each dimension. With this method, we can get the membership on the same dimension with different centers and shapes.

Essentially, the proposed method combines two processes together, the projection of membership functions from *d*-dimension to each dimension and the reshaping of the membership functions, which has been mentioned in [6].

Until now, we have obtained the membership functions (its parameters) through cluster method partitioning universe of discourse. Obviously, each cluster means a fuzzy rule to describe the system behavior. It is quite easy for us to identify the fuzzy model with such membership functions initialized by our method. The detail of the fuzzy model identification has been clearly discussed in [5].

4 Simulations

First we consider a classic nonlinear function approximation in 3-dimension to show its power and illustrate some of its properties. Then we will consider a benchmark problem involving the prediction of a chaotic time series and compare the performance of our method with the published results of other methods.

4.1 Classic Nonlinear Function Approximation

Consider a classic nonlinear function in 3-D, as shown below, to be modeled.

$$z(x, y) = \frac{\sin(x)\sin(y)}{xy} \tag{7}$$

From the grid points within the range [-10 10]×[-10 10] in the input space from the above function (7), 441 training data points are obtained first with an interval of 0.5 from –10 to 10. Here, we just specify 16 clusters to obtain the fuzzy model and Fig.4 shows the result of our modeling and the membership of our model. The detail of the result of our modeling and other modeling method are listed in Table1 It is very clear that our model with 16 rules is a little bit better than subcluster model with 18 rules. Also our model with 25 rules is much better than subcluster model with 22 rules. (Here the problem of determining the number of rules/clusters is involved. If one does not want to fix the number of rules in advance it is possible to use a cluster validity measure to compute the number of rules/clusters. Cluster validity measures have been discussed in [1, 13].)

Table 1. Comparison of the methods

Method	Model structure	Train epochs	RMSE
Our model	16 rules	Without training	0.0345
	25 rules		0.0168
Subcluster model	18 rules	Without training	0.0376
	22 rules		0.0326
ANFIS	16 rules	100	0.0479

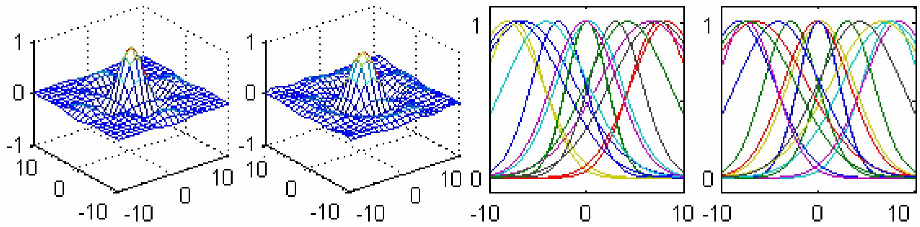


Fig. 2. The result of the modeling and the membership of the model (a) training data set; (b) our model with 16 rules; (c) MFs on x-axis; (d) MFs on y-axis

Table 1 shows that the proposed method can obtain a better model with more simple structure. This is because that the centers of Gaussian membership functions are determined more reasonably by MCV algorithm and the parameters σ of Gaussian membership functions are obtained by equation (6). Compared with Babuska [6], the proposed method does not need the processes of projecting, smoothing and parameterizing of membership functions. Also, compared with subcluster modeling method, our method can obtain different σ values for different clusters in the same dimension, which gives more reasonable descriptions of the size and shape of the data set. In the following simulation, the same result has also been found.

4.2 Prediction of a Chaotic Time Series

We now consider a benchmark problem in model identification to predict a time series generated by chaotic Mackey—Glass differential delay equation (Mackey and Glass, 1977 [14]) defined by

$$\dot{x}(t) = \frac{0.2x(t-\tau)}{1+x^{10}(t-\tau)} - 0.1x(t) \tag{8}$$

The task is to use past values of *x* up to the time t to predict the value of *x* at some time *t+△t* in the future. The standard input for this type of prediction is *N* points in the time series spaced S apart, i.e., the input vector is *y={x(t-(N-1)S),…, x(t-2S), x(t-S), x(t)}*. To allow comparison with the published results of other methods, we use*τ=17, N=4, S=6, △ T=6*. Therefore, each data point in the training set consists of

x={x(t-18), x(t-12), x(t-6), x(t), x(t+6)}

where the first 4 elements correspond to the input variables and the last element corresponds to output variable. We will compare the performance with ANFIS proposed by Jang (1993) as well as subcluster-estimation modeling proposed by Chiu (1994).

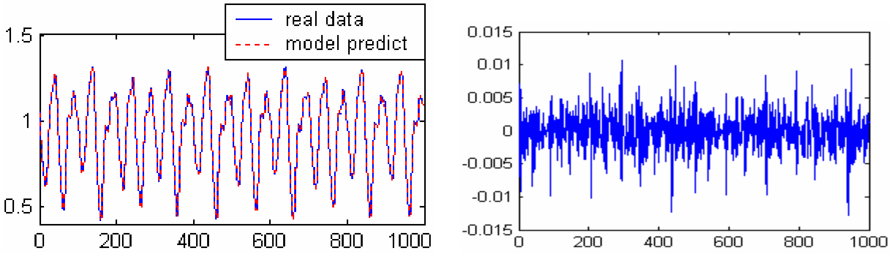


Fig. 3. Mackey—Glass time series prediction by our model with 16 rules (a) Mackey—Glass time series data and model prediction; (b) Model prediction error

For the Mackey—Glass time series problem, we used the same data set as that used by Jang (1993), which consisted of 1000 data points extracted from *t=118* to *t=1117*. The first 500 data points were used to train the model, and the last 500 data points were used to check the generalization ability of the model. The results of the three methods are compared in Table Ⅱ. Also the real data together with our model (without any training) prediction are compared in Fig.3 (a) and the prediction error are showed in Fig.3 (b).

Table 2 illustrates the comparisons among three methods. Although we have to accept the fact that subclustering estimation-based modeling is fast and very good to deal with this problem, we are satisfied for the modeling result with our method. Our fuzzy model with 16 or 20 rules is a concise description of the chaotic process. It is obvious that our model is less complicated than the model obtained by subclustering.

Table 2. Comparison of three methods

Method	Model rules	Train epochs	RMSE
Our model	16	0①	trn 0.0036 chk 0.0036
		10	trn 0.0019 chk 0.0018
	20	0	trn 0.0030 chk 0.0030
		10	trn 0.0015 chk 0.0014
Subcluster model②	25	0	trn 0.0034 chk 0.0032
		10	trn 0.0016 chk 0.0014
ANFIS③	16	499.5	trn 0.0016 chk 0.0015

Remark 1: 0 means the model built up without any training.
Remark 2: this result published in [5].
Remark 3: this result published in [8].

5 Conclusion

A new fuzzy modeling method has been presented in the paper. In the method, we take advantage of MCV clustering algorithm to partition universe of discourse. Covariance matrix generated by MCV algorithm is used to determine the critical parameters for Gaussian membership functions. The method does not involve any nonlinear search or approximation so that our modeling will not be affected by bad initial states. Different centers and shapes for the membership functions on each dimension for each cluster center are obtained by our method. This provides a possibility to have a better performance than other estimation method, especially in high dimensions. The experiments in part 4 show that our method can give a model with more concise description with simpler structure.

In a word, the method proposed in this paper can provide an accurate and robust modeling while significantly reducing model complexity.

References

1. Bezdek J.C.: Pattern recognition with fuzzy objective function. New York, Plenum Press. (1981).
2. Gustafson D.E., Kessel W.C.: Fuzzy clustering with a fuzzy covariance matrix. Proceedings of the IEEE CDC, San Diego, CA, USA, (1979) 761-766.
3. Krishnapuram R. and Kim J.: Clustering Algorithms on Volume Criteria. IEEE Trans. Fuzzy Systems, vol.8, no.2, (2000)228-236.
4. Yager, R. and Filev, D.: Generation of Fuzzy Rules by Mountain Clustering. Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, (1994) 209-219.
5. Chiu, S.: Fuzzy Model Identification Based on Cluster Estimation. Journal of Intelligent & Fuzzy Systems, Vol. 2, No. 3, Sept. 1994.
6. Babuska R.: Fuzzy Modeling for Control. Kluwer Academic Publishers, Boston, 1998.

7. Lagarias, J.C., Reeds J. A., Wright M. H., and Wright P. E.: ConvergenceProperties of the Nelder-Mead Simplex Method in Low Dimensions. SIAM Journal of Optimization, Vol. 9 Number 1, (1998) 112-147.
8. Jang J.S.R., Sun C.T., Mizutani E.: Neuro-Fuzzy and Soft computing. Prentice Hall, 1997.
9. Dempster A. P., Laird N. M., and Rubin D. B.: Maximum Llikelihood From Incomplete Data via the EM Algorithm. J. Roy. Statist. Soc., vol.B, no.39, (1977)1-38.
10. Glodberg D. E., Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, MA, 1989.
11. Holland J. H.: Adaptation in Natural and Artificial Systems. University of Michigan Press, Michigan.1975.
12. Zadeh L.A.: Fuzzy Logic Toolbox User's Guide Version 2.1.1, Berkeley, CA, MathWorks, Inc, 2001.
13. X.L.Xie and G.Beni: A Validity Measure for Fuzzy Clustering. IEEE Trans. Pattern Anal. 13 (1991) 841-847
14. Mackey M. Glass L: Oscillation and Chaos in Physiological Control Systems. Science 197:287-289.

An Efficient Clustering Approach for Large Document Collections

Bo Han[1,2], Lishan Kang[1], and Huazhu Song[3]

[1] School of Computer Science, Wuhan University,
Wuhan, Hubei 430072, P.R.China
hanboemail@yahoo.com
[2] Center for Information Science and Technology,
Temple University, Philadelphia, PA 19122, U.S.A
[3] School of Computer Science and Technology,
Wuhan University of Technology, Wuhan, Hubei 430070, P.R.China

Abstract. A vast amount of unstructured text data, such as scientific publications, commercial reports and webpages are required to be quickly categorized into different semantic groups for facilitating online information query. However, the state-of-the art clustering methods are suffered from the huge size of documents with high-dimensional text features. In this paper, we propose an efficient clustering algorithm for large document collections, which performs clustering in three stages: 1) by using permutation test, the informative topic words are identified so as to reduce feature dimension; 2) selecting a small number of most typical documents to perform initial clustering 3) refining clustering on all documents. The algorithm was tested by the 20 newsgroup data and experimental results showed that, comparing with the methods which cluster corpus based on all document samples and full features directly, this approach significantly reduced the time cost in an order while slightly improving the clustering quality.

1 Introduction

As millions of scientific publications, commercial reports and unstructured webpages available on the Internet and Information Retrieval (IR) systems, text clustering has become an increasingly important technique in real world applications [1, 2, 4, 5, 7]. For example, with an online query, an IR system generally returns a long list of documents for detailed browse. Though these documents are ranked by their relevance to the occurring of query keywords, they are not organized into groups, each with a distinctive topic. Users suffer the experience of jumping from one topic to another totally different by following the retrieval result list.

To facilitate the online queries, documents are expected to be quickly grouped into hierarchical categorizations by using clustering algorithms. With categorization topics, users can efficiently browse and identify their interested documents. This significant advantage drives researchers to perform widely study on this topic. The similarity based approaches (hierarchical agglomerative clustering, k-means etc.) aimed to

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 240–247, 2005.

minimize the similarities among documents in a cluster while maximized the similarities within clusters [3, 8]. The model based approaches attempted to produce a model to cover all the documents in a cluster and distinguish the documents in other clusters [6, 9, 10, 11]. Experiments showed that both of them obtain the competitive clustering quality while the latter approaches run faster than the formers. However, these state-of-the art text clustering approaches are based on traditional clustering algorithms which are originally designed for structured records in databases. Their running time is far beyond the users' expectation for online query. This expensive time cost is resulted from the huge size of unstructured document collections, with each document represented by a high-dimensional feature vector.

Specifically, we observe that the time cost mainly come from two sources. Firstly, each document is always presented as the popular bag-of-words-model vector, each element in the vector corresponding to a distinct word in such document. Thousands of words occur in document corpus. Consequently, they are described by high-dimensional vectors. The number of feature dimension is even much larger than the number of documents. Secondly, the computation for clustering all documents in each round is time-consuming.

In this paper, we propose an efficient clustering approach for large size of corpus. It supposes that a well-built document cluster should express a specific topic and most documents in this cluster contain representative topic words. This assumption is reasonable and understandable. With this assumption, our approach performs clustering in three stages. In the first stage, we perform permutation test and consequently order the words by their distinguishing ability among clusters. A large number of general words on the bottom of the order list are filter out and the feature dimension is drastically reduced. In the second stage, those selected informative topic words are used to evaluate all documents. The documents with no such words or with many such topic words will be regarded as the "difficult" examples for clustering. Other documents can be regarded as "easier" clustered. Next, with the difficult documents in a reduced dimension, a model-based clustering algorithm is applied to train model parameters. Since this dataset contains far less features and examples than that of original dataset, clustering algorithm converges fast, resulting in some good basic model parameters. In the final stage, the initial model parameters are used to further cluster all documents. Since these initial model parameters work well on the "difficult" examples, refined clustering will quickly converged on other "easier" documents.

We tested this approach on the 20 newsgroup data. Experimental results showed that it not only reduced the time cost in an order, but also slightly improve the clustering quality.

2 Efficient Clustering

The basic idea for our efficient clustering approach is to greatly reduce the feature dimension of documents as well as the number of examples in each clustering round. Firstly, many text classification practices have empirically proved that redundant semantic words exist in corpus and documents can be successfully categorized with just from 100 to 1000 words. Comparing with the original vocabulary size (for instance, larger than 20,000 words in large size of corpus), the feature dimension is drastically decreased. However, the general feature selection in classification can not be applied in clustering due to the lack of class labels. In this paper, we use permutation test to quickly evaluate each word and choose the possible informative ones. In the meantime, we observe that time costs of clustering also come from the expensive computation involving large number of documents in each round. Our intuition is that different documents are clustered with "difficulties" in different degree. By assuming that a cluster of documents express a specific topic and share some common topic words, we believe the documents with no topic words or including many of topic words will be more "difficult" for clustering. For example, if a piece of news is titled with words "investment" and "finance", it most probably belongs to the business news. While another piece of news contains words "golf" and "match", it mostly belongs to a piece of sport news. But if a piece of news contain all above words in its title, it is hard to distinguish their category by just observing these four words. Motivated by this fact, we can evaluate the document clustering difficulties by computing topic words. With a small number of such "difficult" examples, we can quickly derive the cluster model parameters in multinomial-model-based document clustering. Since these models can group these difficult examples, they are good initial guess for clustering all documents.

Our approach is described by the following three steps in details. For convenience, we denote document collections as A=D×W, where |D| is the number of the documents, and |W| is the vocabulary size (number of features). In this way, the matrix A represents corpus with the popular bag-of-words model in text mining. The number of clusters is denoted as K.

2.1 Identify Informative Words in Documents

The vocabulary size of document collections is very large. For example, the popular benchmark 20-Newgroup dataset includes more than 43,586 distinct words. This number is even larger than the total number of documents (19,949). In this high dimension space, we can hardly efficiently classify or cluster documents. Fortunately, studies on text mining showed that rich redundant information exists in text and we can distinguish documents by using just 100-1000 informative words. For the task of classification, given example labels, the informative words can be effectively chosen by feature selection. However, in clustering case, no document labels are available and it is difficult to select features.

Our intuition is that if a word appears in each document with almost the same possibility, it provides no information to distinguish documents with different topics. We design a word evaluation procedure by using permutation test, which recently has been recommended for feature selection in high-dimensional bioinformatics applications [12]. Permutation test is implemented by randomly shuffling data into K clusters for r times. In each shuffle, we compute the average frequency of a word in each cluster and its standard derivation s across all clusters. After repeating r time's shuffles, the average standard derivation $\sum s/r$ is obtained, which suggests the potential distinguishing ability of the corresponding word. We use this value to rank all words and then select informative features. The following algorithms describe the detailed steps,

Algorithm: Identify Informative Words

Step1: Randomly distribute all documents into K clusters. Let each cluster contains the same number of documents.

Step2: Let $FW_{ij} = 1$ if the document i contains word j; otherwise, $FW_{ij} = 0$. In which, i=1, 2, …, |D|, and j=1,2, …, |W|.

Step3: For each cluster k, we compute the word vector $CW_k = <CW_{k1}, CW_{k2}, \ldots, CW_{kw}>$ where $CW_{kj} = \sum_{\text{document } i \in \text{Cluster } k} FW_{ij}$. (j=1, 2, ……, |W|)

Step4: For a word j in the sequence CW_{kj} (k=1, 2,……, K), compute the standard derivation s for this word.

Step5: Repeat Step1 to Step4 r times; compute the average standard derivation $\sum s/r$ among clusters of a word.

Step6: Sort the words by the average standard derivation in descending order. By using the top m percent words, a new document matrix A'=D×W' is generated, where $|W'| = \frac{m \times |W|}{100}$.

By the above algorithm, only those words with bigger standard derivation are regarded as informative words and hence, they are kept in new matrix W'. Considering that 100-1000 words are enough for text classification and the possible noise involved in the corpus, we let |W'| in the range of [400, 5000].

2.2 Seeking Difficult Documents and Fast Initial Clustering

In a classification task, the class separating hyper-plane is decided by those examples which are similar with each other in feature space and thus harder to distinguish from one group to other groups. For example, support vector machines, which perform pretty well in text mining, aim to find category hyper-planes by using support vectors. The support vectors are "difficult" examples in classification. This encourages us to firstly seek a small number of difficult documents for initial clustering, which provide a good basis for further work on all document collections.

In 2.1, we select the top p words (denoted as set I) with largest standard derivation among clusters as the most informative topic words. We expect to use them to decide that, to what extend, a document is difficult for clustering. With a reasonable way, the documents including no word in I or including many words in I are regarded as the difficult examples. This is easy to be understand, if a document contains no word in I, it means we can not find the topic words to cluster this document; if a document contains many words in I, we are also confused that which cluster these documents should be grouped in. By these two heuristics, we design the following algorithm to seek "difficult" documents,

Algorithm: Seek Difficult Documents

Step1: Choose the top p informative words from W', denoted by I. We assume these words are topic words for K clusters.

Step2: Find the documents D' without words in I or containing more than t words in I.

Step3: Let A''=D'×W', which forms the most difficult documents for clustering.

Step4: Perform multinomial-model-based clustering procedure on A'', the model parameters of each cluster are computed.

In multinomial model, with naïve bayes assumption, the probability of a document d_i grouped into cluster j is computed as,

$$P(j\,|\,d_i) = \prod_{l=1}^{|W|} P_j(W_l)^{cw_l} \tag{1}$$

Where $P_j(W_l)$ is the probability of word W_l being present in cluster j. By Laplacian smoothing, it is computed as,

$$P_j(W_l) = \frac{1+\sum_i P(j\,|\,d_i)cw_i}{|V|+\sum_l \sum_i P(j\,|\,d_i)cw_i} \tag{2}$$

Since |D'| is greatly less than |D|, the initial clustering procedure can be fast performed.

2.3 Refined Clustering

In clustering practice, we see that a good guess of model parameters will significantly speed up the clustering procedure. In the second step, we used a small number of “difficult” examples to obtain some good initial model parameters. Applying them on all documents A’ which are supposed to include large number of other “easier” examples, after very few rounds, the multinomial-model-based clustering algorithm is converged and all documents can be appropriately clustered.

2.4 Analysis on Time Complexity

In multinomial-model-based clustering, the time complexity is $O(RNFK^2)$, where R is the number of rounds for clustering, N is the number of examples, F is the number of features and K is the number of clusters. Our algorithm has the similar time complexity $O(R'N'F'K^2)$, but R’ is much less than R in initial clustering and further full clustering; N’ is no more than half of N; F’ is just one fifth or one twentieth of F. Hence, the overall time cost has been saved by at least one order.

3 Experiments

3.1 Dataset

We use the popular 20-NewGroup dataset to test the clustering performance. There are 20 different usenet newsgroups among 19949 documents. The vocabulary size is as large as 43586. The data set is pre-processed by rainbow, a popular statistical text tool developed by Andrew McCallum. With rainbow, we stem the words, remove words which appearing less than 10 times in all documents and remove empty documents. The resulted dataset contains 18827 examples with 21697 words. The following table.1 lists the number of news in each class.

For the convenience of testing some parameters in our algorithm, such as m in 2.1, p and t in 2.2, we choose 100 news from each 20 groups to form a small dataset S with the same vocabulary size 21697.

Table 1. The number of news in each class

alt.atheism	799	sci.space	987
sci.crypt	864	sci.electronics	981
talk.politics.guns	910	sci.med	990
talk.politics.misc	775	Soc.religion.christian	997
comp.sys.mac.hardware	961	Talk.politics.mideast	940
talk.religion.misc'	628	misc.forsale	972
comp.graphics	973	Rec.autos	990
comp.os.ms-windows.misc	985	Rec.motorcycles	994
comp.sys.ibm.pc.hardware	982	Rec.sport.baseball	994
comp.windows.x	980	Rec.sport.hockey	999

3.2 Measures

In experiments, we used Normalized Mutual Information (NMI) as the evaluation criterion, which is a popular measure to evaluate the text clustering results given the class labels. It is computed as,

$$\mathrm{NMI} = \frac{\sum_{h,l} n_{h,l} \log(\frac{n \cdot n_{h,l}}{n_h n_l})}{\sqrt{(\sum_h n_h \log \frac{n_h}{n})\,(\sum_l n_l \log \frac{n_l}{n})}} \tag{3}$$

Here, n_h represents the number of documents in class h, n_l the number of documents in cluster l, and $n_{h,l}$ is the number of documents in class h as well as in cluster l. The NMI value is 1 when clustering results are the same as the class labels and close to 0 for a random partitioning.

3.2 Experimental Results and Analysis

In the experiments, we used the multinomial-model-based clustering algorithm with soft assignment strategy. Because the results are sensitive to the initial cluster distribution, we performed 30 times of clustering for one experiment and reported the average results. The experiments are performed in the Matlab 6.0 on the machine with CPU PIII 1.33 GHz and 256M RAM. Apply standard multinomial model based clustering algorithm on small dataset S, we have NMI=0.51 ± 0.04 and the average running time is 2578 seconds. If it is tested on the all dataset A, we have NMI=0.52 ± 0.03 and the running time is 8618.36 seconds.

Let p=50 and t=6, Fig. 1 shows the relationship between NMI and m, Fig. 2 shows the relationship between running time of clustering algorithm and m. Considering both clustering quality and running time, m $\in$[10,30] is a good choice.

Let m=20, Fig.3 shows the relationship between NMI and p, Fig.4 shows the relationship between NMI and t. We see that p=50 and t=6 is an appropriate choice.

With the setting m=20, p=50 and t=6, we cluster all documents in 20-newsgroup, the final result is NMI=0.53 ± 0.03 in 758.92 seconds.

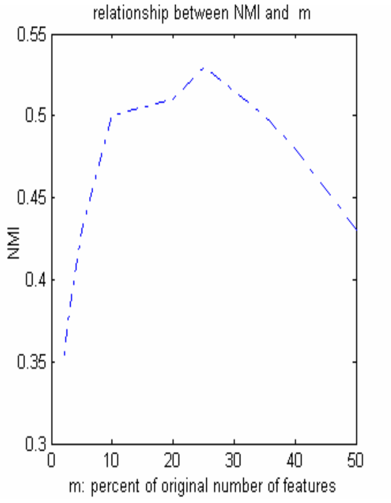


Fig.1. Relation between NMI and m

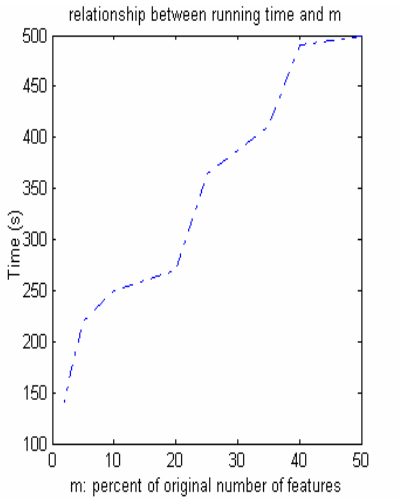


Fig. 2. Relation between running time and m

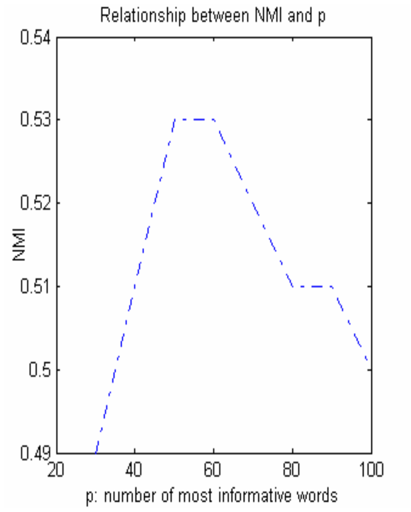


Fig. 3. Relation between NMI and p

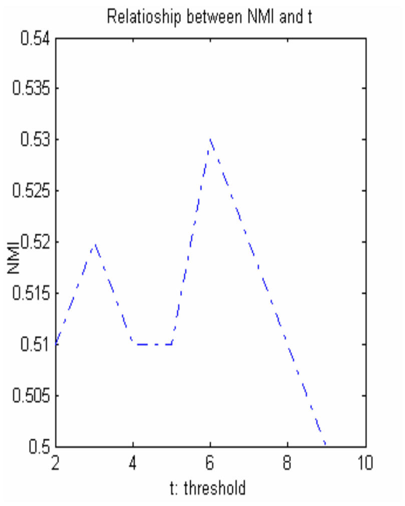


Fig. 4. Relation between NMI and t

By the above experimental results, we can see that with slightly improve the clustering quality; our algorithm can run in one tenth time of standard multinomial model based algorithm.

4 Conclusion

Clustering a large number of documents is a computationally expensive task. In this paper, we propose an efficient approach with three stages: the first stage aims to reduce feature dimension; the second one performs initial cluster on difficult examples as to obtain a good guess on the clustering model parameters; the last stage finally refines the clustering procedure on all document collections. Experimental results revealed that comparing the traditional text clustering methods, this approach not only reduces the time cost in an order, but also slightly improve the clustering accuracy.

In future work, we hope to further speed up the clustering procedure by comprehensive statistical analysis on informative words and difficult examples. We also hope to apply this approach on other datasets with large number of high-dimension examples, such as micro-array in bioinformatics.

References

1. Banerjee, A., Dhillon, I. S., Ghosh J., Sra, S.: Clustering on hyperspheres using Expectation Maximization (Technical Report TR-03-07). Dept of Computer Sciences, Uniersity of Texas (2003)
2. McCallum, A., Nigam, K., Ungar, L. H.: Efficient Clustering of High-dimensional Data Sets with Application to Reference Matching. Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2000)
3. Strehl, A., Ghosh, J., Mooney, R.: Impact of similarity measures on web-page clustering. In Proceeding of the AAAI2000 Workshop on Artificial Intelligence for Web Search, Austin, Texas, (2000)
4. Cutting, D., Kager, D., Pedersen, J., Tukey, J.W.: Scatter/Gather A cluster-based approach to browsing large document collections. Proc. ACM SIGIR (1992)
5. Dhillon, I.S.: Co-clustering documents and words using bipartite spectral graph partitioning. Proc. 7th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2001)
6. Tantrum, J., Murua, A., Stuetzle, W.: Hierarchical model-based clustering of large datasets through fractionation and refractionation. Proc. 8th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (2002)
7. Steinbach, M., Karpis, G., Kumar, V.: A comparison of document clustering techniques. KDD Workshop on Text mining (2000)
8. Zamir, O., Etzioni O.: Web document clustering: A feasibility demonstration. ACM SIGIR (1998)
9. Zhong, S., Ghosh, J.: A Comparative Study of Generative Models for Document Clustering. SDM Workshop on Clustering High Dimensional Data and Its Applicatons, San Francisco, CA (2003)
10. Zhong, S., Ghosh, J.: A unified framework for model-based clustering. Intelligent Engineering Systems Through Artificial Neural Networks (ANNIE), St. Louis, MO. (2002)
11. Zhong, S., Ghosh, J.: A unified framework for model-based clustering and its applications to clustering time sequences (Technique Report), Dept of ECE, University of Texas at Austin. (2002)
12. Hsing, T., Attoor S., Dougherty E.: Relation Between Permutation-Test *P* Values and Classifier Error Estimates. J. Machine Learning, Vol. 52 (2003) 11–30

Clustering Categorical Data Using Coverage Density

Hua Yan, Lei Zhang, and Yi Zhang

Computational Intelligence Laboratory,
School of Computer Science and Engineering,
University of Electronic Science & Technology of China
Chengdu, 610054 P.R. China
{huayan, leizhang, zhangyi}@uestc.edu.cn
http://cilab.uestc.edu.cn

Abstract. In this paper, a new algorithm based on the idea of coverage density is proposed for clustering categorical data. It uses average coverage density as the global criterion function. Large sparse categorical databases can be clustered effectively by using this algorithm. It shows that the algorithm uses less memory and time by analyzing its time and space complexity. Experiments on two real datasets are carried out to illustrate the performance of the proposed algorithm.

1 Introduction

The importance of clustering categorical data has received considerable attention in recent years [1,3,4,5]. The first algorithm for clustering categorical data is K-modes proposed in [5], which is developed from the well-known K-means [6] for clustering numerical data. The algorithm ROCK proposed in [4] uses the concepts of neighbors and links to measure the similarity of data for clustering categorical data. In [3], based on dynamical systems, STIRR algorithm was proposed. The algorithm CACTUS was proposed in [1] by using summary information. However, all of these algorithms are difficult to deal with sparse databases with huge volumes and high dimensionality.

Generally, intra-cluster similarity and inter-cluster dissimilarity are used to evaluate the quality of clustering results. The criterion function is defined by computing the intra-cluster similarity and (or) the inter-cluster dissimilarity of clusters. The criterion function can be defined locally or globally. A local criterion function uses the pair-wise similarity between data points. Local criterion functions are used by many authors, see for examples, [1,3,4,5]. There is a drawback for local criterion functions since the computation cost is usually high for large databases. Unlike the local criterion function, the global criterion function computes the optimized value in a cluster level and no pair-wise similarity computing is necessary. Generally speaking, the global criterion function method can work efficiently for clustering categorical data for large databases.

The LargeItem algorithm [8] and the CLOPE algorithm [7] are the pioneering clustering algorithms using global criterion function. The LargeItem algorithm gives a similarity measure for a cluster of transactions based on large items. An item is large

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 248–255, 2005.

if its support is larger than that of the user specified, otherwise it is called a small item. The global criterion function used by the LargeItem algorithm sums the intra-cluster dissimilarity measured by the number of small items and the inter-cluster similarity computed by the overlapping number of large items across clusters. The CLOPE algorithm [7] defines the global criterion function by using geometric properties of the cluster histograms. A larger height-to-width ratio of cluster histogram means a better clustering. In this paper, a quite simple global criterion function is proposed based on the concept of coverage density to evaluate the quality of clustering categorical data.

This paper is organized as follows. The new algorithm and its complexity analysis will be given in Section 2. Experimental results with real categorical data will be carried out in Section 3 to further illustrate the algorithm. Conclusions will be presented in Section 4.

2 Clustering with Coverage Density

Suppose a database D with domains $D_1, D_2, \cdots, D_m$ contains categorical data points $\{t_1, t_2, \cdots, t_n\}$. Let $A_1, A_2, \cdots, A_m$ be a set of categorical attributes within $D_1, D_2, \cdots, D_m$ respectively, a_i means an attribute value within attribute set A_i . So a data point t can be represented as $\{a_1, a_2, \cdots, a_m\}$ where $t \in D_1 \times \ldots \times D_m$. A clustering result C is a partition $\{C_1, C_2, \cdots, C_k\}$ of D , where $C_1 \bigcup \ldots \bigcup C_k = D, C_i \neq \varnothing, C_i \bigcap C_j = \varnothing$.

The Coverage Density (CD) is defined as the percentage of the occupied area by all data points to the whole rectangle area decided by the distinct attribute values and the number of data points. Given a cluster C_i, it is easy to compute its coverage density. Let the number of distinct attribute values is $D(C_i)$, the number of data points in the cluster C_i is $P(C_i)$, and the sum length of data points in cluster C_i is $S(C_i)$, the Coverage Density of cluster C_i is

$$CD(C_i) = \frac{S(C_i)}{P(C_i) \times D(C_i)} . \tag{1}$$

The coverage density reflects the tightness of a cluster intuitively. The higher coverage density means the higher intra-cluster similarity among the data points within a cluster. The Average Coverage Density (ACD) of a clustering result $C = \{C_1, C_2, \cdots, C_k\}$ is

$$ACD(C) = \frac{\sum_{i=1}^{k} CD(C_i)}{k} . \tag{2}$$

The higher the ACD, the better the quality of clustering result. For example, there is a database {abc, abcd, bcde, cde} and two clustering results are shown in the Fig. 1.

The first clustering result is {{abc, abcd, bcde, cde}}, the second clustering result is {{abc, abcd}, {bcde, cde}}. From the Fig. 1 we can get the $ACD(CR1) = \frac{\frac{14}{4\times 5}}{1} = \frac{7}{10}$ and the $ACD(CR2) = \frac{\frac{7}{2\times 4}+\frac{7}{2\times 4}}{2} = \frac{7}{8}$. The $ACD(CR2)$ is bigger than the $ACD(CR1)$, which means the second clustering result has a higher intra-cluster similarity than the first clustering result. So the average coverage density computing function is used as the global criterion function by the new algorithm and the optimal clustering result is got when the result of criterion function is maximized.

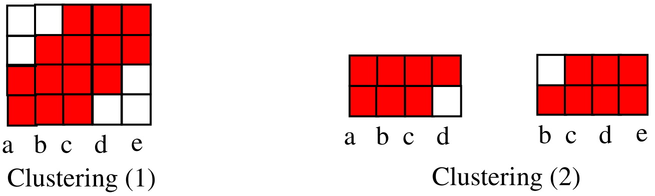


Fig. 1. Illustration of coverage densities for two clustering results

However, there exists an exception in the new method. Consider if there is only one data point in a cluster partition C_i, then the $CD(C_i) = 100\%$. So when we put every data point in a separate cluster we will get the highest average coverage density. But it is not a reasonable result obviously if you imagine the number of cluster partitions of a large database is equal to the number of its data points. The concept of Minimum Merging Coverage Density (MMCD) is introduced to deal with such kind of condition. The MMCD is an input parameter less than 100% given by the user. This parameter is used when the algorithm needs to decide that a data point should be added to an existed cluster or be a new cluster. If the existed cluster's coverage density is still higher than the MMCD after merging the new data point, the algorithm will not create a new cluster for this new data point although creating new cluster will increase the average coverage density of clustering result. So the MMCD can be viewed as a parameter that can control the number of clusters. A small MMCD can get less number of clusters than a large MMCD.

The description of algorithm CCCD (Clustering Categorical using Coverage Density) is given in the Fig.2. The initial clustering result is got in the initialization phase. There is a few more scans of database D to refine the clustering result in the iteration phase. The algorithm will stop if there are no changes on clustering result.

The space requirement of CCCD is quite small. Firstly, the CCCD need not store all the data points in RAM because the algorithm swaps reading data from disk file. It just keeps one data point in RAM every time. Secondly, the CCCD applies the RAM space dynamically for recording the key information of clusters. The key information of a cluster includes the number of data points, the size of cluster, the number of distinct items, the distinct items and their occurrences. Let's suppose each part of the key information needs 4 bytes space separately. So the total RAM space required for the CCCD is approximately $K \times (4+4+4+N \times 4)$, where K means the number of clusters, and N means the quantity of distinct items in a cluster at most. For a categorical data points set with up to 10k distinct items and with a 1k clusters need a 40MB RAM for recording occurrences of distinct items at most. The space complexity of CCCD is $O(K \times N)$.

```
Algorithm: CCCD
Input: Database D of data points; minimum merging coverage density, MMCD.
Output:
   data points labeled with cluster id; clustering result set.
Method:
   /*Phase 1 – Initialization*/
   While not end of the database file D
         read data point d from D;
         add d into existed Ci  or a new cluster Cj and compute coverage densities;
         if no existed coverage densities larger than MMCD
                   create new cluster Cj and put d into Cj;
                   write <d, j> back to D;
         else
                   put the d into Ci that maximize average coverage density;
                   write <d, i> back to D;
   end While;
   /*Phase 2 - Iteration*/
   Repeat
         moveMark = false
         locate to the start of database file D;
         While not end of database file D
         read <d, i> from D;
                   move d to existed clusters j that maximize ACD;
                   if Cj is existed
                             moveMark=true;
                             write <d,j> back to D;
         end While
   Until not moveMark;
```

Fig. 2. The CCCD algorithm description

There're two most time-consuming parts in the CCCD. One part is in the initialization phase. The other part is in the iteration phase. The algorithm needs computing the global average coverage density in these two parts to determine adding or removing a data point to a cluster. Because the direct computing of global average coverage density needs a summary operation of all cluster coverage density, we utilize the change value of coverage density after adding or removing a data point to determine the best destination cluster. The computation of new coverage density is quite simple and fast in our new algorithm due to the key information storage of each cluster. The adding update computing formula is

$$CD(C_i)\_adding = \frac{S(C_i) + d.length}{(P(C_i)+1) \times D(C_i)\_new}. \tag{3}$$

The $d.length$ stands for the length of new data point and the $D(C_i)\_new$ means the number of distinct items after adding a new data point. The removing update computing formula is

$$CD(C_i)\_removing = \frac{S(C_i) - d.length}{(P(C_i)-1) \times D(C_i)\_new}. \tag{4}$$

The $d.length$ stands for the length of removed data point and the $D(C_i)\_new$ means the number of distinct items after removing a data point. In the initialization phase, the cluster with the maximum new coverage density that is greater than MMCD is chosen, otherwise a new cluster is created. In the iteration phase, the algorithm sums the change value of source cluster and the destination cluster that has maximum change value. If the sum is positive, then move the data point to the destination cluster, otherwise, keep the data point in the source cluster. We just give the algorithm description of adding update operation in Fig.3 because of the process similarity between adding and removing operations. It is obviously that the time complexity of adding or removing updates is $O(d.length)$ from Fig. 3. So the time complexity of algorithm is $O(N \times K \times A)$, where N, K and A stand for number of data points in database, the maximum number of clusters and the average length of a data point separately.

```
float adding_update_operation(C, d)
  {
     S_new = C.S+d.length;
     P_new = C.P+1;
     D_new = C.D;
     For (i=0; i<d.length; i++) {
          If d.item[i] not exist in C.items
                   D_new ++;
                   }
     return S_new/(P_new*D_new)-C.S/(C.P*C.D);
     }
```

Fig. 3. Algorithm description of adding update operation

3 Experiments

In this section we use two real datasets from UCI machine learning repository (http://www.ics.uci.edu/~mlearn/MLRepository.html) to test the feasibility and effectiveness of our new algorithm.

3.1 Zoo

The zoo dataset contains 101 data entries for animals. Each data entry has 18 attributes (animal name, 15 Boolean attributes, 1 numeric with set of values [0,2,4,5,6,8], animal type values 1 to 7). We convert the dataset into data points with 36 distinct attribute values before the test. The animal name is ignored in our transformed file and the animal type values are kept in the data points to evaluate the correctness of the clustering result.

After running the CCCD with different MMCD values, we found an acceptable result shown in Table 1 with acceptable number of clusters and tolerable mixing of different types of animals when the MMCD value is 70%. Eight clusters have the same type animals and four clusters have small numbers of different type animals according to Table 1.

Table 1. MMCD = 70%, 12 clusters

ClusterID Type	1	2	3	4	5	6	7	8	9	10	11	12
1 (41)	25	4					9			1		2
2 (20)			20									
3 (5)					2				1			2
4 (13)		10			2			1				
5 (4)					3							1
6 (8)				2		6						
7 (10)				2							8	

3.2 Mushroom

The mushroom dataset has been used by many clustering algorithms, such as ROCK [4], LargeItem algorithm [8], the CLOPE algorithm [7], CLASD [9] and LIMBO [2], etc.. It contained 8124 instances. Each entry has 22 categorical attributes (e.g. cap-shape, cap-color, habitat etc.) and is labeled either “edible” or “poisonous”. We converted the data set into the data points with 125 distinct attribute values before the test and 2480 instances with missing value ‘?’ are ignored in the transformed data set.

We run CCCD with different parameter values of MMCD from 30% to 80%. We can get the following detail results in every test: the number of clusters, purity of clustering result and detailed results in every cluster. In every cluster we record the number of edible and poisonous mushroom instances separately. The purity is calculated by summing up the larger one of the number of edibles and the number of poisonous in every cluster. The experiment results are shown in Table 2 and Table 3. Table 2 shows the result when we set 31% as the value of MMCD. We found that the most clusters clearly belong to the "edible" or "poisonous" categories although five clusters have a small percentage of data points from the other category. Table 2 collects the purity and the number of clusters of different MMCD from 30% to 80%. From Table 3 we saw the contradiction of the purity and the number of clusters in our algorithm. In fact the increasing of the number of clusters become intolerable while we want to get much higher purity. For example, there are 708 clusters when the purity is 8124.

Table 2. MMCD = 31%, 15 clusters, purity = 7642

ClusterID	E	P	ClusterID	E	P
1	1518	229	9	0	13
2	146	27	10	0	1028
3	0	42	11	26	612
4	0	150	12	512	0
5	584	0	13	168	1764
6	1088	0	14	96	32
7	24	0	15	46	0
8	0	19			

Table 3. Results of Different MMCD

MMCD (%)	Clusters	Purity	MMCD (%)	Clusters	Purity
30	12	6854	55	103	7935
31	15	7642	58	96	7913
36	35	7124	60	119	7940
39	48	7448	65	77	8061
43	67	7494	70	119	8110
46	61	7511	75	246	8113
49	69	7698	80	708	8124
51	88	7648			

4 Conclusions

A new algorithm has been developed in this paper for clustering categorical data based on the concept of coverage density. The algorithm uses parameter MMCD to control the tightness of intra-cluster similarity. Experiments confirmed the good performance of the algorithm. This algorithm can also be used to cluster transactional data set and web usage data. The further study will be focused on constructing a better global criterion function to abate the contradiction between the purity and the number of clusters.

References

1. Venkatesh Ganti, Johannes Gehrke, and Raghu Ramakrishnan.: CACTUS: Clustering Categorical Data Using Summaries. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, (KDD)*, pages 73–83, San Diego, CA, USA, 15–18 August 1999. ACM Press.
2. P. Andritsos, P. Tsaparas, R. Miller, K. C. Sevcik.: LIMBO: Scalable Clustering of Categorical Data. International Conference on Extending DabaBase Tehnology (EDBT), pages 123-146, Heraklion Crete, Greece, 2004.
3. D. Gibson, J. Kleinberg and P. Raghavan.: Clustering categorical data: an approach based on dynamical systems. In Proceedings of the 24th VLDB Conference, New York, USA, 1998.
4. Sudipto Guha, Rajeev Rastogi, and Kyuseok Shim.: ROCK: A Robust Clustering Algorithm for Categorical Attributes. In *Proceedings of the 15th International Conference on Data Engineering, (ICDE)*, pages 512–521, Sydney, Australia, 23–26 March 1999. IEEE Press.
5. Zhexue Huang.: Extensions to the K-Means Algorithm for Clustering Large Data Sets with Categorical Values. *Workshop on Research Issues on Data Mining and Knowledge Discovery, (DMKD)*, 2(3): 283–304, 1998.
6. Jiawei Han and Michelle Kamber.: *Data Mining: Concepts and Techniques*. Morgan Kaufmann, 2001.
7. Yiling Yang, Xudong Guan, Jinyuan You.: CLOPE: A Fast and Effective Clustering Algorithm for Transactional Data. In SIGKDD ’02, July 23-26, 2002, Edmonton, Alberta, Canada.
8. Ke Wang, Chu Xu, Bing Liu. Clustering Transactions Using Large Items. In Proc. CIKM’99, Kansas, Missouri, 1999.
9. Charu C. Aggarwal, C. Procopiuc, and Philip S. Yu.: Finding Localized Associations in Market Basket Data, IEEE Trans. Knowledge and Data Eng., Vol. 14, No. 1, Jan. 2002, pp. 51-62.

A New Support Vector Machine for Data Mining

Haoran Zhang, Xiaodong Wang, Changjiang Zhang, and Xiuling Xu

College of Information Science and Engineering,
Zhejiang Normal University, Jinhua 321004, China
{hylt, wxd, zcj74922, jkxxl}@zjnu.cn

Abstract. This paper proposes a new support vector machine (SVM) with a robust loss function for data mining. Its dual optimal formation is also constructed. A gradient based algorithm is designed for fast and simple implementation of the new support vector machine. At the same time it analyzes algorithm's convergence condition and gives a formula to select learning step size. Numerical simulation results show that the new support vector machine performs significantly better than a standard support vector machine.

1 Introduction

With increasing amounts of data being generated there is a need for fast, accurate and robust algorithms for data mining. Real-world data sets are often characterized by having large numbers of examples, being highly unbalanced, corrupted by noise, and often highly non-linear. One recent technique that has been developed to address these issues is the support vector machine [1-2]. The support vector machine has been developed as robust tool for classification and regression in noisy and complex domains. The two key features of support vector machines are generalization theory [3], which leads to a principled way to choose a hypothesis; and, kernel functions, which introduce non-linearity in the hypothesis space without explicitly requiring a non-linear algorithm. For regression problems, SVM exploits the idea of mapping input data into a high dimensional reproducing kernel Hilbert space (RKHS) where a linear regression is performed. The advantages of regression SVM are: a global minimum solution as the minimization of a convex programming problem; relatively fast training speed; and sparseness in solution representation. The standard SVM regression adopts Vapnik's ε-loss function and its solution is achieved by reducing the problem to a quadratic programming problem with two constraints. Some researchers have advocated changing the objective function in SVM to simplify the required optimization problem [4-5]. In this paper, we try to modify the formulation of the standard SVM and consider a simple modified dual optimization problem, then introduce a gradient-based algorithm for fast and simple implementation of the new SVM. This paper is organized as follows. In section 2 we briefly give a unified robust loss function, based on which propose a new support vector regression, and then deduce its dual optimal problem. In sections 3 a gradient-based algorithm for new SVM is proposed, its convergence condition is given. In section 4 we take an example to demonstrate the learning performance of the proposed method and compare against standard SVM. Finally, a conclusion can be found in section 5.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 256–266, 2005.

2 A Unified Robust Loss Function and New SVM

Several robust cost functions have been used in SVM regression, such as Vapnik's ε-loss function, Huber's robust cost [6], or the ridge regression approach [7]. Here, we propose a more general robust cost function that has the above mentioned ones as particular cases. It can be expressed as the following piecewise-defined function:

$$L(e) = \begin{cases} 0 & |e| \le \varepsilon \\ \frac{1}{2}(|e| - \varepsilon)^2 & \varepsilon \le |e| \le e_c \\ c(|e| - \varepsilon) - \frac{1}{2}c^2 & |e| \ge e_c \end{cases}$$

Where $e_c = \varepsilon + c$.

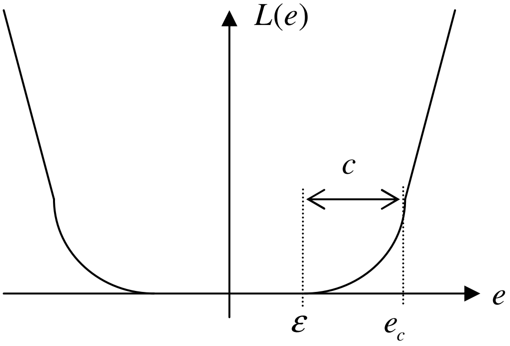


Fig. 1. an unified robust loss function

The three different intervals of unified function serve to deal with different kinds of noise. Insensitive zone $|e| \le \varepsilon$ is adequate for low-frequency variations such as wander or baseline deviations. The quadratic cost zone takes into account the observation noise, the L2 norm in this zone is appropriate for Gaussian processes. The linear cost zone limits the effect of either outliers or jitter noise. Parameter c can be selected by trial and error.

For a nonlinear regression problem, firstly we may transform it into a linear regression problem. This can be achieved by a nonlinear map $\phi(\cdot)$ from input space into a high dimensional feature space and constructing a linear regression function there, that is:

$$f(x) = w^T \phi(x) + b \tag{1}$$

We would like to find the function with the following structural risk function:

$$R_{stru} = \frac{1}{2}\|w\|^2 + C \cdot R_{emp}[f] \tag{2}$$

We take loss function $R_{emp} = L(e)$ which it is the unified robust loss function we proposed earlier.

According to Eqn. (2) the above regression problem can be transformed to the following constraint optimization problem:

$$\begin{aligned} \min \quad & \frac{1}{2}(w^T w + b^2) + C\Big(\sum_{i\in I_1}\frac{1}{2}(\zeta_i^2 + \zeta_i^{*2}) + \sum_{i\in I_2} c(\zeta_i + \zeta_i^*)\Big) \\ & y_i - w^T\phi(x_i) - b \le \varepsilon + \zeta_i \\ s.t. \quad & w^T\phi(x_i) + b - y_i \le \varepsilon + \zeta_i^* \\ & \zeta_i, \zeta_i^* \ge 0 \end{aligned} \tag{3}$$

where I_1 is the set of samples for which $0 < \zeta_i < c$ or $0 < \zeta_i^* < c$, and I_2 is the set of samples for which $\zeta_i \ge c$ or $\zeta_i^* \ge c$. We append the term b^2 to $w^T w$. Extensive computational experience, as in [8] indicates that this formulation will add advantages such as strong convexity of the objective function.

For optimization problem (3), we construct a Lagrange function from both the objective function and the corresponding constraints:

$$\begin{aligned} L = & \frac{1}{2}(w^T w + b^2) + C\Big(\sum_{i\in I_1}\frac{1}{2}(\zeta_i^2 + \zeta_i^{*2}) + \sum_{i\in I_2} c(\zeta_i + \zeta_i^*)\Big) - \sum_{i=1}^{l}\alpha_i(\varepsilon + \zeta_i - y_i + w^T\phi(x_i) + b) \\ & - \sum_{i=1}^{l}\alpha_i^*(\varepsilon + \zeta_i^* + y_i - w^T\phi(x_i) - b) - \sum_{i\in I_1}(\gamma_i\zeta_i + \gamma_i^*\zeta_i^*) - \sum_{i\in I_2}(\mu_i\zeta_i + \mu_i^*\zeta_i^*) \end{aligned} \tag{4}$$

The KKT conditions of (4) are as follows:

$$\frac{\partial L}{\partial b} = b - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0 \tag{5}$$

$$\frac{\partial L}{\partial w} = w - \sum_{i=1}^{l}(\alpha_i - \alpha_i^*)\phi(x_i) = 0 \tag{6}$$

$$\frac{\partial L}{\partial \zeta_i} = C\zeta_i - \alpha_i - \gamma_i = 0 \qquad i \in I_1 \tag{7}$$

$$\frac{\partial L}{\partial \zeta_i^*} = C\zeta_i^* - \alpha_i^* - \gamma_i^* = 0 \qquad i \in I_1 \tag{8}$$

$$\frac{\partial L}{\partial \zeta_i} = Cc - \alpha_i - \mu_i = 0 \qquad i \in I_2 \tag{9}$$

$$\frac{\partial L}{\partial \zeta_i^*} = Cc - \alpha_i^* - \mu_i^* = 0 \qquad i \in I_2 \tag{10}$$

Substituting (5), (6), (7) ,(8), (9) and (10) into (4), we have the dual optimization problem:

$$\begin{aligned} L = & -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\phi(x_i)\cdot\phi(x_j)+1) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\ & -\frac{1}{2C}\sum_{i\in I_1}(\alpha_i^2 + \alpha_i^{*2}) \end{aligned} \tag{11}$$

From KKT conditions, we also can get:

$$-\frac{1}{2C}\sum_{i\in I_2}(\alpha_i^2 + \alpha_i^{*2}) = -\frac{l_2 Cc^2}{2} \tag{12}$$

According to (12), Eqn. (11) can be re-written as:

$$\begin{aligned} L = & -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(\phi(x_i)\cdot\phi(x_j)+1) - \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\ & -\frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2 + \alpha_i^{*2}) + \frac{l_2 Cc^2}{2} \end{aligned}$$

For an optimization problem, the objective function subtracting a constant number does not change its optimization solution, so we can subtract constant number $\frac{l_2 Cc^2}{2}$ from objective function L, then we have:

$$\begin{aligned} \overline{L} = & \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)(k(x_i, x_j)+1) + \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*) \\ & +\frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2 + \alpha_i^{*2}) \end{aligned} \tag{13}$$

where $k(x_i, x_j) = \phi(x_i)^T \phi(x_j)$. From KKT conditions, we can get constraint conditions of optimization problem:

$$0 \le \alpha_i, \alpha_i^* \le Cc, \qquad i = 1, \cdots, l$$

The output of new SVM may be written as:

$$f(x)=\sum_{i=1}^{l}(\alpha_i-\alpha_i^*)(k(x_i,x)+1) \tag{14}$$

3 Designing Gradient Based Training Algorithm

For the above discussion, we get the new SVM's dual optimization problem:

$$\begin{aligned}&Min[\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i-\alpha_i^*)(\alpha_j-\alpha_j^*)(k(x_i,x_j)+1)+\varepsilon\sum_{i=1}^{l}(\alpha_i+\alpha_i^*)-\sum_{i=1}^{l}y_i(\alpha_i-\alpha_i^*)\\&+\frac{1}{2C}\sum_{i=1}^{l}(\alpha_i^2+\alpha_i^{*2})]\\&s.t.\ \ 0\le\alpha_i,\alpha_i^*\le Cc,\quad i=1,\cdots,l\end{aligned} \tag{15}$$

For (15), we derive objective function's gradient as:

$$\frac{\partial\overline{L}}{\partial\alpha_i}=\sum_{j=1}^{l}(\alpha_j-\alpha_j^*)(k(x_i,x_j)+1)+\varepsilon-y_i+\frac{\alpha_i}{C}$$

$$\frac{\partial\overline{L}}{\partial\alpha_i^*}=\sum_{j=1}^{l}(\alpha_j^*-\alpha_j)(k(x_i,x_j)+1)+\varepsilon+y_i+\frac{\alpha_i^*}{C}$$

Let:

$$E_i=y_i-\sum_{j=1}^{l}(\alpha_j-\alpha_j^*)(k(x_i,x_j)+1) \tag{16}$$

We get:

$$\frac{\partial\overline{L}}{\partial\alpha_i}=-E_i+\varepsilon+\frac{\alpha_i}{C}$$

$$\frac{\partial\overline{L}}{\partial\alpha_i^*}=E_i+\varepsilon+\frac{\alpha_i^*}{C}$$

According to the theory of gradient based algorithm, the increment of optimal variable can be set as:

$$\delta\alpha_i=-\eta\frac{\partial\overline{L}}{\partial\alpha_i}=-\eta(-E_i+\varepsilon+\frac{\alpha_i}{C})$$

$$\delta\alpha_i^*=-\eta\frac{\partial\overline{L}}{\partial\alpha_i^*}=-\eta(E_i+\varepsilon+\frac{\alpha_i^*}{C})$$

Due to the constraint condition $0 \le \alpha_i, \alpha_i^* \le Cc$, we take:

$$\Delta\alpha_i = \begin{cases} -\alpha_i & \delta\alpha_i < -\alpha_i \\ \delta\alpha_i & -\alpha_i \le \delta\alpha_i \le Cc - \alpha_i \\ Cc - \alpha_i & \delta\alpha_i > Cc - \alpha_i \end{cases} \tag{17}$$

$$\Delta\alpha_i^* = \begin{cases} -\alpha_i^* & \delta\alpha_i^* < -\alpha_i^* \\ \delta\alpha_i^* & -\alpha_i^* \le \delta\alpha_i^* \le Cc - \alpha_i^* \\ Cc - \alpha_i^* & \delta\alpha_i^* > Cc - \alpha_i^* \end{cases} \tag{18}$$

Finally we can get a training algorithm of the new SVM as follow:

(1) Initialize $\alpha_i = 0, \alpha_i^* = 0$;

(2) Compute $k(x_i, x_j) + 1$, $i, j = 1, ..., l$;

(3) Choose a sample i randomly;

(4)

(4.1) Compute $E_i = y_i - \sum_{j=1}^{l} (\alpha_j - \alpha_j^*)(k(x_i, x_j) + 1)$;

(4.2) Compute $\delta\alpha_i, \delta\alpha_i^*$;

(4.3) Compute $\Delta\alpha_i, \Delta\alpha_i^*$;

(4.4) $\begin{matrix} \alpha_i = \alpha_i + \Delta\alpha_i \\ \alpha_i^* = \alpha_i^* + \Delta\alpha_i^* \end{matrix}$;

(5) If $\max_i(|\Delta\alpha_i|, |\Delta\alpha_i^*|) < \tau$, stop program, output results; else select the sample i which make $\max_i(|\Delta\alpha_i|, |\Delta\alpha_i^*|)$, then go to (4).

Now we give a rigorous proof on convergence of above algorithm, and based on which we select learning rate η. Let $R_{ij} = k(x_i, x_j) + 1$. In each iteration we update two optimal variables α_i, α_i^*, the increment of objective function (15) is as follow:

$$\Delta\overline{L} = \overline{L}(\alpha_i + \Delta\alpha_i, \alpha_i^* + \Delta\alpha_i^*) - \overline{L}(\alpha_i, \alpha_i^*)$$

$$= (\Delta\alpha_i - \Delta\alpha_i^*) \sum_{j=1}^{l} (\alpha_j - \alpha_j^*) R_{ij} + \frac{1}{2} (\Delta\alpha_i - \Delta\alpha_i^*)^2 R_{ii} + \varepsilon(\Delta\alpha_i + \Delta\alpha_i^*) - y_i(\Delta\alpha_i - \Delta\alpha_i^*)$$

$$+ \frac{1}{2C} (2\Delta\alpha_i\alpha_i + 2\Delta\alpha_i^*\alpha_i^* + (\Delta\alpha_i)^2 + (\Delta\alpha_i^*)^2)$$

$$=\Delta\alpha_i(\varepsilon-E_i+\frac{1}{2C}(CR_{ii}+1)\Delta\alpha_i+\frac{1}{C}\alpha_i)+\Delta\alpha_i^*(\varepsilon+E_i+\frac{1}{2C}(CR_{ii}+1)\Delta\alpha_i^*+\frac{1}{C}\alpha_i^*)$$
$$-\Delta\alpha_i\Delta\alpha_i^*R_{ii} \tag{19}$$

Let:

$$u=-E_i+\varepsilon+\frac{\alpha_i}{C}$$

$$u^*=E_i+\varepsilon+\frac{\alpha_i^*}{C}$$

Formula（19）can be written as:

$$\Delta\bar{L}=\Delta\alpha_i(u+\frac{1}{2C}(CR_{ii}+1)\Delta\alpha_i)+\Delta\alpha_i^*(u^*+\frac{1}{2C}(CR_{ii}+1)\Delta\alpha_i^*)-\Delta\alpha_i\Delta\alpha_i^*R_{ii} \tag{20}$$

Now, considering the three possible updates of α_i,α_i^*.

Case 1: $\Delta\alpha_i=-\eta u,\Delta\alpha_i^*=-\eta u^*$

$$\Delta\bar{L}=-\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u)-\eta u^*(u^*-\eta\frac{1}{2C}(CR_{ii}+1)u^*)-\eta^2uu^*R_{ii}$$
$$=-[u^2(\eta-\eta^2\frac{1}{2C}(CR_{ii}+1))+u^{*2}(\eta-\eta^2\frac{1}{2C}(CR_{ii}+1))+\eta^2uu^*R_{ii}]$$
$$=-(u\quad u^*)\begin{pmatrix}a_1 & b_1\\ b_1 & a_1\end{pmatrix}\begin{pmatrix}u\\ u^*\end{pmatrix}$$

where $a_1=\eta-\eta^2\frac{1}{2C}(CR_{ii}+1)$, $b_1=\frac{\eta^2}{2}R_{ii}$. If matrix $\begin{pmatrix}a_1 & b_1\\ b_1 & a_1\end{pmatrix}$ is a positive Matrix, $\Delta\bar{L}<0$, algorithm is convergent. From $\begin{pmatrix}a_1 & b_1\\ b_1 & a_1\end{pmatrix}$ being positive, we get:

$$a_1>0,a_1^2-b_1^2>0$$

From $a_1>0$ we get: $\eta<\frac{2C}{CR_{ii}+1}$; from $a_1^2-b_1^2>0$ we get: $\eta<\frac{2C}{2CR_{ii}+1}$. Finally we have:

$$\eta<\frac{2C}{2CR_{ii}+1}$$

Case 2: $\Delta\alpha_i=-\eta u,\Delta\alpha_i^*=-\alpha_i^*$

$$\Delta\bar{L}=-\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u)-\alpha_i^*(u^*-\frac{1}{2C}(CR_{ii}+1)\alpha_i^*)-\eta u\alpha_i^*R_{ii}$$

Due to $\Delta\alpha_i^* = -\alpha_i^* > -\eta u^*$, we get:

$$\Delta\bar{L} = -\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u) - \alpha_i^*(u^* - \frac{1}{2C}(CR_{ii}+1)\alpha_i^*) - \eta u\alpha_i^* R_{ii}$$

$$< -\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u) - \frac{\alpha_i^*}{\eta}(\alpha_i^* - \frac{\eta}{2C}(CR_{ii}+1)\alpha_i^*) - \eta u\alpha_i^* R_{ii}$$

$$= -(\sqrt{\eta}u \quad \frac{\alpha_i^*}{\sqrt{\eta}})\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}\begin{pmatrix} \sqrt{\eta}u \\ \frac{\alpha_i^*}{\sqrt{\eta}} \end{pmatrix}$$

where: $a_2 = 1 - \frac{\eta}{2C}(CR_{ii}+1)$, $b_2 = \frac{\eta}{2}R_{ii}$. If matrix $\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}$ is a positive matrix, $\Delta\bar{L} < 0$, algorithm is convergent. From $\begin{pmatrix} a_2 & b_2 \\ b_2 & a_2 \end{pmatrix}$ being positive, we get:

$$a_2 > 0, a_2^2 - b_2^2 > 0$$

From $a_2 > 0$ we get: $\eta < \frac{2C}{CR_{ii}+1}$, from $a_2^2 - b_2^2 > 0$ we get: $\eta < \frac{2C}{2CR_{ii}+1}$.

Finally we have:

$$\eta < \frac{2C}{2CR_{ii}+1}$$

Case 3: $\Delta\alpha_i = -\eta u, \Delta\alpha_i^* = Cc - \alpha_i^*$

$$\Delta\bar{L} = -\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u) + (Cc-\alpha_i^*)(u^* + \frac{1}{2C}(CR_{ii}+1)(Cc-\alpha_i^*)) + \eta u(Cc-\alpha_i^*)R_{ii}$$

Due to $\Delta\alpha_i^* = Cc - \alpha_i^* < -\eta u^*$, we get:

$$\Delta\bar{L} < -\eta u(u-\eta\frac{1}{2C}(CR_{ii}+1)u)$$

$$+ (Cc-\alpha_i^*)(-\frac{Cc-\alpha_i^*}{\eta} + \frac{1}{2C}(CR_{ii}+1)(Cc-\alpha_i^*)) + \eta u(Cc-\alpha_i^*)R_{ii}$$

$$= -(\sqrt{\eta}u \quad \frac{-Cc+\alpha_i^*}{\sqrt{\eta}})\begin{pmatrix} a_3 & b_3 \\ b_3 & a_3 \end{pmatrix}\begin{pmatrix} \sqrt{\eta}u \\ \frac{-Cc+\alpha_i^*}{\sqrt{\eta}} \end{pmatrix}$$

where: $a_3 = 1 - \frac{\eta}{2C}(CR_{ii}+1)$, $b_3 = \frac{\eta}{2}R_{ii}$. If matrix $\begin{pmatrix} a_3 & b_3 \\ b_3 & a_3 \end{pmatrix}$ is a positive matrix, $\Delta\overline{L} < 0$, algorithm is convergent. From $\begin{pmatrix} a_3 & b_3 \\ b_3 & a_3 \end{pmatrix}$ being positive, we get:

$$a_3 > 0, a_3^2 - b_3^2 > 0$$

From $a_3 > 0$ we get: $\eta < \frac{2C}{CR_{ii}+1}$, from $a_3^2 - b_3^2 > 0$ we get: $\eta < \frac{2C}{2CR_{ii}+1}$. Finally we have:

$$\eta < \frac{2C}{2CR_{ii}+1}$$

We have discussed three possible values of $\Delta\alpha_i, \Delta\alpha_i^*$. The discussion method of the other possible values of $\Delta\alpha_i, \Delta\alpha_i^*$ is the same, in every case we can prove: if $\eta < \frac{2C}{2CR_{ii}+1}$, $\Delta\overline{L} < 0$, algorithm is convergent. Let $\nu = \max_i(R_{ii})$, $\eta < \frac{2C}{2C\nu+1}$. We select the learning step:

$$\eta = \frac{2C}{2C\nu+1} - \vartheta \tag{21}$$

where ϑ is a small positive number. Because optimal problem is a convex problem, the local minimum is also the global one the learning algorithm we designed will monotonically decrease and stop when a global minimum is reached.

4 Numerical Simulation and Comparison

In this section we will take a concrete example to illustrate the proposed new SVM and its training algorithm. The function under consideration is SINC function:

$$y(x) = \frac{\sin(x)}{x} \qquad x \in [-10, 10]$$

We add some Gaussian noise to the output: $\overline{y}(x) = y(x) + \delta$. We begin by generating 100 data. The data is then split into two portions, one ($(x_i, \overline{y}_i)$) for training and one ((x_i, y_i)) for testing. We take Gaussian function as the kernel function, i.e. $k(x_i, x_j) = \exp(-\frac{\left\| x_i - x_j \right\|^2}{2\sigma^2})$, and set SVM's design parameter as: $\sigma = 1$, $C = 250$, $c = 0.2$, $\varepsilon = 0.01$. The simulation result is as follow:

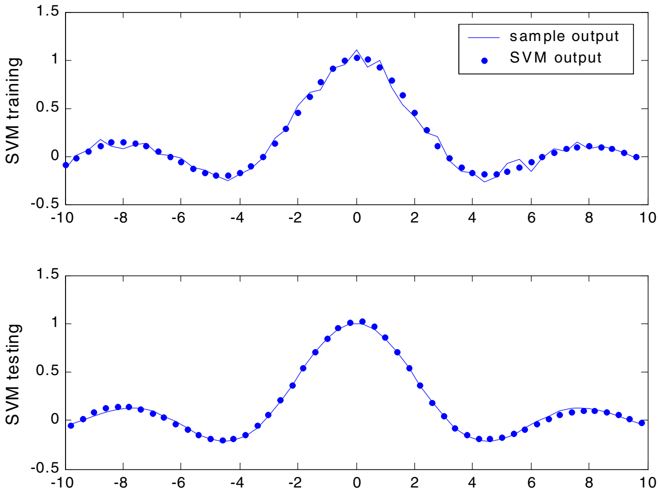


Fig. 2. the new SVM's training and testing results

During simulation, we also compare the performance of the proposed SVM and the standard SVM, their simulation error is as follow:

Table 1. The simulation error comparison between standard SVM and new SVM

Noise's variance	0.05	0.1	0.2	0.3	0.4
Testing error of standard SVM	0.0244	0.0701	0.1142	0.1821	0.2768
Testing error of new SVM	0.0182	0.0344	0.0822	0.1266	0.1875

From above simulation results, we can see the generalization of new SVM is better than that of standard SVM, the new SVM possesses high robustness to low and high noise. The reason is that we adopt a unified robust loss function in the new SVM. The loss function can deal with different kinds of noise effectively.

5 Conclusion

In this paper we propose a new regression SVM for data mining, and deduce its dual optimization problem. We then propose an algorithm to train it, and at the same time analyze algorithm's convergence condition to derive a formula to select learning step size. The experimental results presented show that our proposed learning techniques are capable of achieving better performance.

References

1. Boser, B., Guyon, I., Vapnik, V.: A Training Algorithm for Optimal Margin Classifiers. In Proceedings of Fifth Annual Workshop on Computational Learning Theory, New York: ACM Press (1992).
2. Cortes, C., Vapnik, V.: Support vector networks. Machine Learning, 20 (1995) 273–297.
3. Vapnik, V. : Statistical Learning Theory. New York: John Wiley and Sons (1997).
4. Joachims, T., Scholkopf, B.: Making large scale SVM learning practical. In Advances in Kernel Methods- Support Vector Learning, Cambridge: MIT Press (1998).
5. Mangasarian, O.L., Musicant, D.R. : Lagrangian support vector machines. Journal of Machine Learning Research, 1 (2001) 161-177.
6. Muller, K.R., Smola, A., Ratsch, G.: Advances in Kernel Methods - Support Vector Learning. Cambridge: MIT Press (1999).
7. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines and Other Kernel-based Learning Methods. Cambridge: Cambridge Univ. Press (2000).
8. Mangasarian, O.L., Musicant, D.R.: Active support vector machine classification. Technical Report 00-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin (2000).

The Infinite Polynomial Kernel for Support Vector Machine

Degang Chen[1], Qiang He[2], and Xizhao Wang[2]

[1] Department of Mathematics and Physics,
North China Electric Power University,
102206, Beijing, China
chengdegang@263.net
[2] Department of Mathematics and Computer Science,
Hebei University, Baoding, Hebei, 071002, China

Abstract. This paper develops an infinite polynomial kernel k_c for support vector machines. We also propose a mapping from an original data space into the high dimensional feature space on which the inner product is defined by the infinite polynomial kernel k_c. Via this mapping, any two finite sets of data in the original space will become linearly separable in the feature space. Numerical experiments indicate that the proposed infinite polynomial kernel possesses some properties and performance better than the existing finite polynomial kernels.

1 Introduction

Support vector machine (SVM) is a new learning theory presented by Vapnik [1,2]. From the pattern recognition viewpoint, it can briefly be stated as follows. When a given sample set K is linearly separable. The separating hyperplane with the maximal margin, the optimal separating hyperplane, is constructed in the original space. When the sample set is linearly non-separating, the input vectors are mapped into the high-dimensional feature space through some kernel functions. Then in this space an optimal separating hyperplane is constructed. The inner product in the high-dimensional feature space is just the employed kernel, so the complex computing of inner product in the high-dimensional feature space is avoided. This is one of the advantages of SVM. SVM has been shown to provide higher performance than traditional learning machines [3] and has been introduced as powerful tool for solving classification problems. In the mean time the research on SVM theory and applications has drawn more and more attention in recent years. As well known that kernel is one of the most important concepts in the theory of SVM and many efforts have been concentrated to the research of kernels. The well known kernels in the theory of SVM are homogeneous polynomial kernels, inhomogeneous polynomial kernels, Gaussian radial basis function kernels, sigmoid kernels and $B_n -$spline kernels. Both the homogeneous polynomial kernels and inhomogeneous polynomial kernels map the original data set into a finite dimensional polynomial space (feature space) and the structures of features are clear (there is a whole field of pattern recognition research

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 267 – 275, 2005.

studying polynomial classifiers [4]), but it is possible that for a fixed polynomial kernel there exists a data set which is not separable in the feature space relative to this kernel since the feature space is finite dimensional. In the mean time the Gaussian radial basis function kernels map the original data set into an infinite dimensional space and any finite data set is linear separable in the feature space with respect to this kernel [5], but the structures of the features relative to the Gaussian radial basis function kernels are difficult to analysis. This statement suggests us to consider infinite polynomial kernels for SVM. In this paper we propose an infinite polynomial kernel on the open unit ball and study the map with respect to this kernel which map the original data set into the feature space, we also prove that by this map the images of any finite data set are linear independent in the feature space, this implies any two finite subclasses of the original data set are linear separable in the feature space. Our experiment indicates that this infinite polynomial kernel can really reduce the number of support vectors thus it possesses better properties than the finite polynomial kernel. Thus this kernel can be applied to solve practical problems.

The rest of this paper is organized as follows. A brief review of the theory of SVM will be described in Section 2. The infinite Polynomial Kernels in the open unit ball will be derived in Section 3. Experiments are presented in Section 4. Some concluding remarks are given in Section 5.

2 Kernels for SVM

Let $\{(x_1, y_1),...,(x_l, y_l)\} \subset R^n \times \{+1,-1\}$ be a training set. The SVM learning approach projects input patterns x_i with a nonlinear function $\Phi : x \to \Phi(x)$ into a higher dimension space Z and, then, it separates the data in Z with a maximal margin hyperplane. Therefore, the classifier is given by $f(x) = sign(w^T \Phi(x) + b)$ and parameters w and b are obtained through the minimization of functional $\tau(w) = \frac{1}{2}\|w\|^2$ subject to $y_i(< w, x_i > +b) \ge 1$ for all $i = 1,...,l$. Since the solution of the linear classifier in Z only involves inner products of vectors $\Phi(x_i)$, we can always use the kernel trick[6], which consists on expressing the inner product in Z as an evaluation of a kernel function in the input space $< \Phi(x), \Phi(y) >= k(x, y)$. This way, we do not need to explicitly know $\Phi(\cdot)$ but just its associated kernel $k(x, y)$. When expressed in terms of kernels, the classifier results $f(x) = sign(\sum_{i=1}^{l} y_i \alpha_i k(x_i, x) + b)$, where coefficients $\{\alpha_i\}$ are obtained after a QP optimization of functional $L(w,b,\alpha) = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{l} \alpha_i \{[< x_i, w > -b] y_i - 1\}$ which can be solved by the KKT complementarity conditions of optimization theory [3].

From the above analysis it is clear that the kernel play a key role in the application of SVM, thus a deep insight to the structure of kernels is both of theoretical and practical important. There are two approaches to characterize the kernel [6]. First it can be believed as inner product in a Reproducing Kernel Hilbert Space [6]. On the other hand it is a symmetric real-valued function satisfying the well known Mercer Theorem [6]. The latter statement is always employed to examine a function to be a kernel.

Two kinds of kernels are always applied in SVM. They are translation invariant kernels and dot product kernels. The translation invariant kernels are independent of the absolute position of input x and only depend on the difference between two inputs x and x', so it can be denoted as $k(x,x')=k(x-x')$. The well known translation invariant kernel is the Gaussian radial basis function kernel $k(x,x')=\exp(-\frac{\|x-x'\|^2}{2\sigma^2})$, other translation invariant kernels are B_n –splines kernels [7], Dirichlet kernels [6] and Periodic kernels [6]. A second, important family of kernels can be efficiently described in term of dot product, i.e., $k(x,x')=k(<x,x'>)$. The well known dot product kernels are Homogeneous Polynomial Kernels $k(x,x')=<x,x'>^p$, inhomogeneous Polynomial Kernels $k(x,x')=(<x,x'>+c)^p$ with $c \geq 0$. Both Homogeneous Polynomial Kernels and inhomogeneous Polynomial Kernels map the input set into a finite dimensional Polynomial space. This implies it is possible that two classes of inputs may be non-separable in the feature space for a fixed Polynomial Kernel. For the dot product kernels, the following theorem is always useful.

Theorem 1. [8] A function $k(x,x')=k(<x,x'>)$ defined on an infinite dimensional Hilbert space, with a power series expansion $k(t)=\sum_{n=0}^{\infty} a_n t^n$ is a positive definite kernel if and only if for all n, we have $a_n \geq 0$.

This theorem implies that many kinds of dot product kernels can be considered in SVM.

3 The Infinite Polynomial Kernels in the Open Unit Ball

Since both Homogeneous Polynomial Kernels and inhomogeneous Polynomial Kernels map the input set into a finite dimensional Polynomial space and they cannot linearly separate all the data set in the feature space, they are not very satisfied at least from the theoretical viewpoint even they perform well in some practical problems. In this paper, to overcome the above weakness, we consider a class of infinite Polynomial Kernels in the open unit ball $U_n=\{x \in R^n : \|x\|<1\}$ which can make any finite data set in U_n linear separable in the high dimensional feature space.

Theorem 2. For every $x,x' \in U_n$, $p \in N-\{1\}$, define $k_c(x,x')=\frac{1-<x,x'>^p}{(1-<x,x'>)^p}$, then k_c is a kernel.

Proof. By $x,x' \in U_n$ we have $|<x,x'>|<1$. Let $k_c(t)=\frac{1-t^p}{(1-t)^p}$, then we have $k_c(t)=(1+t+...+t^{p-1})(\sum_{k=0}^{\infty} t^k)^p$ for $|t|<1$, by Theorem 1 we know $k_c(<x,x'>)$ is a kernel.

Suppose $k_c(t)=\sum_{k=0}^{\infty}a_k t^k$, then $a_k=\frac{k_c^{(k)}(0)}{k!}$. For every $x\in U_n$, define C_k to map $x\in U_n$ to the vector $C_k(x)$ whose entries are all possible k th degree ordered products of the entries of x , and define Φ_k by compensating for the multiple occurrence of certain monomials in C_k by scaling the respective entries of Φ_k with the square roots of their numbers of occurrence. Then, by the construction of C_k and Φ_k , we have $<C_k(x),C_k(x')>=<\Phi_k(x),\Phi_k(x')>=<x,x'>^k$.

Define $\Phi(x)=(1,\sqrt{a_1}\Phi_1(x),...,\sqrt{a_k}\Phi_k(x),...,)$, then we have $<\Phi(x),\Phi(x')>=k_c(x,x')$. The feature space with respect to $k_c(x,x')$ can be selected as the Hilbert space generated by $\Phi(U_n)$. The following theorem implies this space is infinite dimensional.

Theorem 3. Suppose $\{x_1,...,x_m\}\subset U_n$ satisfying $x_i\neq x_j$ if $i\neq j$, then $\Phi(x_1),...,\Phi(x_n)$ are linear independent.

Proof. Suppose $x_i=(a_{i1},a_{i2},...,a_{in})$ and $\Phi(x_1),...,\Phi(x_m)$ are linear dependent, then there exists $\alpha_1,\alpha_2,...,\alpha_m$ satisfying at least one of them is not equal to zero and $\alpha_1\Phi(x_1)+\alpha_2\Phi(x_2)+...+\alpha_m\Phi(x_m)=0$ holds. Thus we have $\sum_{i=1}^{m}\alpha_i a_{i1}^{l_1}a_{i2}^{l_2}...a_{in}^{l_n}=0$ where $l_1,l_2,...,l_n\in N\cup\{0\}$.

Let $f_i(x)=a_{i1}+a_{i2}x+...+a_{in}x^{n-1}$, $i=1,...,m$. Then there exists $n_0\in N$ such that any two of $\{f_i(n_0):i=1,...,m\}$ are different. Let $\beta_i=\{1,f_i(n_0),...,f_i^{m-1}(n_0)\}$, $i=1,...,m$, then we have $\beta_1,\beta_2,..,\beta_m$ are linear independent. But by $\sum_{i=1}^{m}\alpha_i a_{i1}^{l_1}a_{i2}^{l_2}...a_{in}^{l_n}=0$ we have $\alpha_1\beta_1+...+\alpha_m\beta_m=0$, this is a contradiction. Thus we have $\Phi(x_1),...,\Phi(x_n)$ are linear independent.

Furthermore by Theorem 3 we have the following conclusions.

Theorem 4. Suppose $\{(x_1,y_1),...,(x_l,y_l)\}\subset U_n\times\{+1\}$, $\{(x_{l+1},y_{l+1}),...,(x_m,y_m)\}\subset U_n\times\{-1\}$, then $\Phi(x_1),...,\Phi(x_l)$ and $\Phi(x_{l+1}),...,\Phi(x_m)$ are linear separable in the feature space.

Proof. $\Phi(x_1),...,\Phi(x_n)$ are linear independent implies any element in the convex hull of one class cannot be the convex combination of the elements of another class, this implies the two convex hulls have empty overlap, notice these two convex hulls are compact, so $\Phi(x_1),...,\Phi(x_l)$ and $\Phi(x_{l+1}),...,\Phi(x_m)$ are linear separable in the feature space.

Thus for any finite data set the optimal hyperplane in the feature space is always available.

Theorem 5. Suppose $\{x_1,...,x_m\} \subset U_n$ satisfying $x_i \neq x_j$ if $i \neq j$, then the Gram matrix $M = (k_c < x_i, x_j >) =< \Phi(x_i), \Phi(x_j) >$ has full rank.

Proof. If $M = (k_c < x_i, x_j >) =< \Phi(x_i), \Phi(x_j) >$ has not full rank, then there exists $\alpha_1, \alpha_2, ..., \alpha_m$ satisfying at least one of them is not equal to zero such that $\sum_{l=1}^{m} \alpha_l < \Phi(x_l), \Phi(x_i) >= 0$, $i = 1,...,m$. So we have $< \alpha_i \Phi(x_i), \sum_{l=1}^{m} \alpha_l \Phi(x_l) >= 0$, $i = 1,...,m$ which implies $< \sum_{i=1}^{m} \alpha_i \Phi(x_i), \sum_{l=1}^{m} \alpha_l \Phi(x_l) >= 0$, thus $\sum_{i=1}^{m} \alpha_i \Phi(x_i) = 0$ and $\Phi(x_1),..., \Phi(x_n)$ are linear dependent. Hence $M = (k_c < x_i, x_j >) =< \Phi(x_i), \Phi(x_j) >$ has full rank.

The feature space with respect to $k_c(< x, x' >)$ is not uniqueness, and Theorem 5 indicates that the selection of feature space(mapping) does not influence the linear independence of a finite class of data in the feature space. By the proof of Theorem 3 we can easily get the following conclusion for the finite Polynomial kernels.

Theorem 6. Suppose $\{(x_1, y_1),...,(x_l, y_l)\} \subset U_n \times \{+1\}$, $\{(x_{l+1}, y_{l+1}),...,(x_m, y_m)\} \subset U_n \times \{-1\}$, then there exists $p \in N$ such that their images are linear separable in the feature space with respect to the kernel $< x, x' >^p$ or $(< x, x' > +1)^p$.

The feature space with respect to every finite Polynomial kernel can be embedded into the feature space with respect to the kernel $k_c(x, x')$ as a subspace, this means there has more different features in the feature space with respect to the kernel $k_c(x, x')$ to be applied to pattern recognition and all these features are constructed by the entries of the input vector. Thus the kernel $k_c(x, x')$ possesses the advantages of Gaussian radial basis function kernels and Polynomial kernels, i.e., it can linearly separate any finite data set and constructions of features are clear, we hope it may perform well in practical problems than the finite Polynomial kernels, we will examine this statement by the experiments in the following section.

4 Experiments

In this section, for the purpose of examining infinite polynomial kernel, we would like to select four databases from machine learning repository (UCI). For these databases, the performance based on new kernel in previous section and finite polynomial kernel will be summarized and compared. Optdigits database includes 5620 cases with 10 classes, 1119 cases are randomly selected to demonstrate. Since the SVM is only for two-class classification problems in this paper, we unite the cases to one class, which belong to class (0,2,4,6,8), and the remaining cases are used as the other class. The four databases' characters are shown in table 1. Applying SVM Toolbox (http://www.isis.ecs.soton.ac.uk/isystems/kernel/svm.zip) to the original data of the four selected databases, one can obtain the optimal separating hyper-planes. The results of these experiments are given in table 2 to 13, where 80% of the databases are randomly selected as the training sets and the remaining 20% as the testing sets. For different types of kernels, the tables show the parameters and the corresponding performance. It is worth noting that the experimental results also depend on the many parameters chosen in the SVM Toolbox.

From tables 2,4,6,8, one can see that the training and testing accuracy are indeed enhanced using infinite polynomial kernel. However, the improvement is not significant. We speculate that the reason is that (1) the data is not enough and (2) database is linear separable very much.

Table 1. The characters of databases

Database Name	Number of samples	Number of features	Category of features
rice	105	5	Numerical
sonar	208	60	Numerical
pima	768	8	Numerical
optdigits	1119	64	Numerical

Table 2. Experiment results for rice database

	infinite polynomial kernel			finite polynomial kernel		
P	Training Accuracy	Testing Accuracy	SV Number	Training Accuracy	Testing Accuracy	SV Number
2	100	93.75	70	100	90.625	73
4	100	93.75	69	100	93.75	73
8	100	96.875	58	100	93.75	72
16	100	96.875	32	100	96.875	40
32	100	96.875	18	100	96.875	17
64	100	96.875	8	100	96.875	9

Table 3. Percentage of common support vector for various kernels for rice database

P	2	4	8	16	32	64
Infinite polynomial kernel	100	100	100	100	94.4	100
Finite polynomial kernel	95.9	94.5	80.6	80.0	100	88.9

From tables 3,5,7,9, one important feature was observed: two types of kernels use approximately the same set of support vectors, but the number of support vectors for infinite polynomial kernel is small in a way(only two cases happen that the number of support vectors for infinite polynomial kernel is bigger than the number of support vectors for finite polynomial kernel), this implies the number of support vectors is really reduced by the infinite polynomial kernel. Noticed that for the support vectors machines, less support vector means better performance of the SVM, so SVM with infinite polynomial kernel developed in this paper have better properties than SV machines with finite polynomial kernel.

Table 4. Experiment results for sonar database

	infinite polynomial kernel			finite polynomial kernel		
P	Training Accuracy	Testing Accuracy	SV Number	Training Accuracy	Testing Accuracy	SV Number
2	100	78.571	159	100	78.571	161
4	100	80.952	124	100	78.571	129
8	100	80.952	91	100	78.571	93
16	100	83.333	69	100	80.952	69
32	100	78.571	65	100	78.571	67
64	100	85.714	67	100	85.714	67

Table 5. Percentage of common support vector for various kernels for sonar database

P	2	4	8	16	32	64
Infinite polynomial kernel	99.4	100	100	100	100	100
Finite polynomial kernel	98.1	96.1	97.8	100	97	100

Table 6. Experiment results with infinite polynomial kernel for pimar database

	infinite polynomial kernel			finite polynomial kernel		
P	Training Accuracy	Testing Accuracy	SV Number	Training Accuracy	Testing Accuracy	SV Number
2	76.384	80.519	562	74.675	78.631	614
4	76.221	80.519	561	74.675	79.268	614
8	76.71	80.519	562	75.974	80.126	614
16	77.036	80.519	560	77.036	80.519	560
32	77.036	80.519	560	76.873	80.519	560
64	77.036	80.519	557	76.873	80.519	614

Table 7. Percentage of common support vector for various kernels for pima database

P	2	4	8	16	32	64
Infinite polynomial kernel	100	100	100	100	100	100
Finite polynomial kernel	91.5	91.4	91.5	100	100	90.7

Table 8. Experiment results with for optdigit database

	infinite polynomial kernel			finite polynomial kernel		
P	Training Accuracy	Testing Accuracy	SV Number	Training Accuracy	Testing Accuracy	SV Number
2	96.745	94.737	262	95.398	92.982	891
4	99.327	96.491	219	98.653	94.982	234
8	100	96.053	891	99.888	96.053	891
16	100	96.491	139	100	96.491	142
32	100	96.053	891	100	96.053	116
64	100	96.053	891	100	96.053	891

Table 9. Percentage of common support vector for various kernels for optdigit database

P	2	4	8	16	32	64
Infinite polynomial kernel	100	97.7	100	100	100	100
Finite polynomial kernel	29.4	91.5	100	97.9	13	100

5 Conclusion

The purpose of this paper is to present infinite polynomial kernel for SVM. By our theoretical analysis this kernel possesses better properties than the existing finite polynomial kernel. Our experiments results almost support our opinion. The infinite polynomial kernel can be applied to practical problems. Further research to the properties and applications of infinite polynomial kernel will be our future work.

References

1. Vapnik, V. N. The Nature of Statistical Learning Theory. New York: Springer-Verlag, 1995
2. Vapnik, V.N. Statistical Learning Theory. New York: Wiley, 1998
3. Burges, C. A tutorial on support vector machines for pattern recognition, Data Mining and Knowledge Discovery, vol. 2, no. 2, 1998

4. Schurmann, J. Pattern Classification: a unified view of statistical and neural approaches. Wiley, New York, 1996
5. Micchelli, C. A. Algebraic aspects of interpolation, Proceedings of Symposia in Applied Mathematics, 36: 81-102, 1986
6. Scholkopf, B. and Smola, A. J. Learning with Kernels, MIT Press, Cambridge, MA, 2002
7. Smola, A. J. Regression estimation with support vector learning machines, Diplomarbeit, Technische Universitat Munchen, 1996
8. Schoenberg, I. J. Positive definite functions on spheres, Duke Mathematical Journal, 9: 96-108, 1942

Routing Attribute Data Mining Based on Rough Set Theory

Yanbing Liu[1,2], Hong Tang[2], Menghao Wang[2], and Shixin Sun[1]

[1] School of Computer Science,
UEST of China, Chengdu 610010, P.R. China
[2] Chongqing University of Posts and Telecommunications,
Chongqing 400065, P.R. China
liuyb@cqupt.edu.cn

Abstract. QOSPF (Quality of Service Open Shortest Path First) based on QoS routing has been recognized as a missing piece in the evolution of QoS-based service offerings in the Internet. A data mining method for QoS routing based on rough set theory has been presented in this paper. The information system about the link is created from the subnet, and the method of rough set can mine the best route from enormous irregular link QoS data and can classify the link with the link-status data. An instance applying to the theory is presented, which verifies the feasibility that the most excellent attribute set is mined by rough set theory for compatible data table.

1 Introduction

With the development of network and application, the routing need to satisfy the QoS demand. However, because the complication of many solving schemes to QoS Routing is NPC(nondeterministic polynomial time completeness), the node can't maintain the network information timely with the continual change of network state [1]. In order to adapt new demand on computer network, it's necessary new feasible and efficient scheme. The traditional OSPF uses the cost metric, which is an unsigned16-bit integer in the range of 1 to 65,535. The default cost for interfaces is calculated based on the bandwidth in the formula 108/BW, with BW being the bandwidth of the interface expressed as a full integer of bps[2]. The traditional OSPF is used to find a "best" path with only one metric such as bandwidth or hops. This mechanism may be low the utilization of network resource and cause an imbalance of the load and it can't satisfy the QoS requirements. Therefore the QOSPF based on QoS Routing is developed.

Data mining is an interdisciplinary field, drawing work from areas including database technology, knowledge acquisition, and rough set theory. The rough set theory is a new mathematical approach to imprecision, vagueness and uncertainty [3]. The concept of reduction the decision table is very useful for feature selection. Because the decision table includes the condition attributes or features and the decision attributes of categories, the procedure of feature selection based

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 276–283, 2005.

the decision table is distinct and effective [3]. The importance of feature selection is due to the potential for speeding up the processes of both concept learning and classification and improving the quality of classification. The work presented here was to decide the link-rank by a series of the link-state attributions. The application of the rough set theory can solve this problem successfully. Usually, the link is classified by many QoS parameters such as link propagation delay, link available bandwidth, link jitter, possibility of connection and hop-counts et al. Then QOSPF can select a best path with the link rank. Data mining (reduction) techniques base on rough set can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.

The instance presented in this paper indicate that the reduction algorithm based on rough set offer an attractive approach to discovery the feature subnet selection problem in routing-reduction table. The remaining of this paper is organized as follows. Section 2 describes the related knowledge about QoS routing, QOSPF. Application of rough set theory and the algorithm to select the QoS routing-attributes are presented in Section 3. Section 4 shows an instance to realize the algorithm. A conclusion is drawn in Section 5.

2 Related Knowledge

2.1 QoS Routing

Route that can satisfy the QoS requirements of a new flow relies on both the knowledge of the flow's requirements and information about the availability of resources in the network. In addition, for purposes of efficiency, it is also important for the algorithm to account for the amount of resources the network has to allocate to support a new flow. In general, the network prefers to select the "cheapest" path among all paths suitable for a new flow, and it may even decide not to accept a new flow for which a feasible path exists, if the cost is deemed too high. Accounting for these aspects involves several metrics on which route is based. They include:

- Possibility of connection: Usually in traditional network, the possibility of link connection is high and this metric can be ignored. But in wireless network, it is a very important metric.
- Link available bandwidth: As mentioned earlier, we currently assume that most QoS requirements are derivable from bandwidth. We further assume that associated with each link is a maximal bandwidth value, e.g., the link physical bandwidth or some fraction thereof that has been set aside for QoS flows. Since for a link to be capable of accepting a new flow with given bandwidth requirements, at least that much bandwidth must be still available on the link, the relevant link metric is, therefore, the (current) amount of available bandwidth.
- Link propagation delay: This quantity is meant to identify high latency links, e.g., satellite links, which may be unsuitable for real-time requests. This

quantity also needs to be advertised as part of extended LSAs (Level Service Agreements), although timely dissemination of this information is not critical as this parameter is unlikely to change (significantly) over time.
- Link jitter: This quantity is used as a measure of the change of link delay. A path with a smaller jitter is more stable and typically preferable.

In QoS-based routing, paths for flows would be determined based on the above QoS requirements of flows. The main objective of QoS-based routing is to realize dynamic determination of feasible paths; QoS-based routing can determine a path, from among possibly many choices, that has a good chance of accommodating the QoS of the given flow. Feasible path selection may be subject to policy constraints, such as path cost, provider selection, etc[4]. It successfully optimizes resource usage. A network state-dependent QoS-based routing scheme can aid in the efficient utilization of network resources by improving the total network throughput. Such a routing scheme can be the basis for efficient network engineering.

2.2 QOSPF

OSPF is defined in RFC 2383.It is a link-state routing protocol that uses Dijkstra's shortest paths to destinations. In OSPF, each router sends s link-state advertisements about itself and its links to all its adjacent routers. Each router that receives a link-state advertisement records the information in its topology database and sends a copy of the link-state advertisement to each of its adjacency. All the link-state advertisements reach all routers in an area, which enables each router in the area to have an identical topology database that describes the routers and links within that area. The router is not sending routing tables but is sending link-state information about its interfaces. When the topology databases are complete, each router individually calculates a loop-tree, shortest-pate tree. Destinations outside the area are also advertised in link-state advertisements. These, however, do not require that routers run the SPF algorithm before they are added to the routing table[1]. Changes of all metrics need to be advertised as part of extended LSAs, so that accurate information is available to the path selection algorithm. QOSPF is QoS extensions to OSPF and support for QoS routing which can be viewed as consisting of three major components:

1. Obtain the information needed to compute QoS paths and select a path capable of meeting the QoS requirements of a given request
2. Establish the path selected to accommodate a new request
3. Maintain the path assigned for use by a given request.

3 Rough Set Methodology

3.1 Relative Reduction of the Rough Set Theory

Rough set theory is a new mathematical approach to information analysis that has been introduced by Zdzislaw Pawlak[5][6].

An information system S is a quadruple (U, A, V, f), where $U = \{x_1, x_2, \cdots, x_n\}$ denotes the set of all objects in the set of all objects in the dataset, A is the set of all attributions which are further classified into two disjoint subsets: the condition attributes $C = \{a_i | i = 1, \cdots, m\}$ and the decision attribution $D = \{d\}$, such that $A = C \cup D$ and $C \cup D = \phi$. $V = \bigcup_{a \in A} V_a$ is a set of attribute values, where V_a is the domain of attribute a. $a_i(x_j)$ denotes the value of x_j on the attribution a_i. $f : U \times A \rightarrow V$ is an information function, which appoints the attribute value of every object x in U.

A discernibility matrix is a $n \times n$ matrix in which the classes are diagonal. In the matrix, the (condition) attributes which can be used to discern between the classes in the corresponding row and column are inserted [7-10].The information system's discernibility matrix $M[C_D(i, j)]_{n \times n}$ where $C_D(i, j)$ is defined as

$$C_D(i, j) = \begin{cases} \{a_k | a_k \in C \wedge a_k(x_i) \neq a_k(x_j)\}, & d(x_i) \neq d(x_j), \\ 0, & d(x_i) = d(x_j), \end{cases} \tag{1}$$

where $i, j = 1, \cdots, n$. From the definition of discernibility matrix, when $|C_D(i, j)| = 1$, the attribute in $C_D(i, j)$ is one of the core attribute set. All the attribute in $C_D(i, j)$ where $|C_D(i, j)| = 1$ consist of the core attribute set, and it may be null. $C_D(i, j) = 0$ when $C_D(i, j)$ contain a core attribute. Then a new simple matrix can be got.

$$\begin{bmatrix} C_{11} & C_{12} & \cdots & C_{1n} \\ C_{21} & C_{22} & \cdots & C_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ C_{n1} & C_{n2} & \cdots & C_{nn} \end{bmatrix}$$

We can get

$$L_{ij} = \bigvee_{a_i \in C_{ij}} a_i; \quad L = \bigwedge_{C_{ij} \neq 0, C_{ij} \neq \phi} L_{ij}; \quad L' = \bigvee_i L_i. \tag{2}$$

The reduced attribute set is $L_i \cup Core(A)$, $Core(A)$ denotes the core attribute set.

3.2 The Mining Attribute Algorithm to Select the QoS Routing-Attributes

This paper makes the elucidation of the algorithm to classify link-rank to realize QOSPF.

First stage is data preprocessing.

Step1: From the historical routing record of the subnet and the QoS information about the links, the information system about the links can be built.

Step2: Then we can draw the discernibility matrix about the information system and can conclude the routing-reduction table.

Second stage is running the mining process.

Step3: Reduction of attribute based on the information system.

Step4: The logical rules can be concluded from the routing-reduction table. Creation the rule of rough decision,save the rule in rule set.

Step5: With the knowledge of QOSPF, the link with QOS attributes can be mined.

With the help of data mining based on rough set theory, the simplification of the routeing-reduction table is to simplify the condtion attributes in routing-reduction table, after that the routing-reduction possesses the ability of the routing-reduction table before simpilficaton, but possesses more important condition attributes[11,12].

4 Instance Study

Figure 1 is an example of subnet's link to validate the above arithmetic. The number on the link denotes link-num. Routing is determined by the link's QoS attribution parameters, such as: available bandwidth, propagation delay, link jitter, bit error ratio and connection possibility. We presume the standard of classification is the link rank that can be described by I, II, ..., VI based on historical routing data. The routing algorithm is used to select the best path from A to D with the QoS attributes.

4.1 Build the Information System

Table 1 shows the information system from Figure 1, all the values in the table denote the measure of the attributes (attribute-weight abstract from the real world).

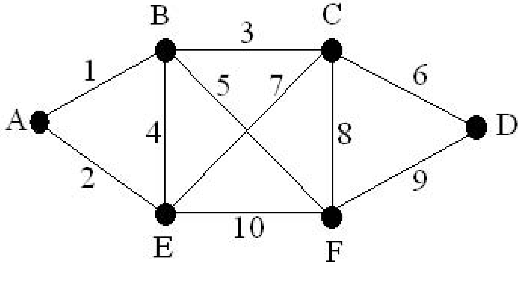


Fig. 1. Subnet

$S = (U, C \cup d, V, f)$ is an information system, where 'U' expressing a finite non-null set of link objects, 'C' expressing a finite non-null set of link's QOS attributes. Here, $U = \{1, 2, 3, \cdots, 10\}$; $C = \{c_1, c_2, c_3, c_4, c_5\}$; $D = \{I, II, III, IV, V, VI\}$; $V_1 = \{1, 2, 3, 4, 5, 6, 7, 8\}$; $V_2 = \{1, 2, 3, 4, 5\}$; $V_3 = \{0, 1, 2, 4, 5\}$; $V_4 = \{1, 2, 3, 5, 6, 8\}$; $V_5 = \{1, 2, 3\}$. The Table. 1 expresses the function of the information 'f'.

Table 1. Information System of Link

Link num.	Available bandwidth (C_1)	Propagation delay (C_2)	Link jitter (C_3)	Bit error ratio (C_4)	Correction possibility (C_5)	Link rank (d)
1	4	4	1	5	3	IV
2	5	3	1	8	2	III
3	1	4	3	5	3	IV
4	1	4	3	5	1	VI
5	6	2	2	3	2	II
6	3	4	3	5	1	IV
7	5	1	1	3	1	V
8	3	1	0	3	3	I
9	7	5	4	1	3	III
10	8	1	1	2	3	I

4.2 Reduction Information System

From Table 1, the discernibility matrix $M(C_D(i,j))$ can be given as follow:

$$\begin{bmatrix}
0 & c_1c_2c_4c_5 & 0 & c_1c_3c_5 & \Omega & 0 & c_1c_2c_4c_5 & c_1c_2c_3c_4 & c_1c_2c_3c_4 & c_1c_2c_4 \\
 & 0 & \Omega & \Omega & c_1c_2c_3c_4 & \Omega & c_1c_2c_4c_5 & \Omega & \Omega & c_1c_2c_4c_5 \\
 & & 0 & c_5 & \Omega & c_1c_5 & c_2c_3c_4c_5 & c_1c_2c_3c_4 & c_1c_2c_3c_4 & c_1c_2c_3c_4 \\
 & & & 0 & \Omega & c_1 & c_2c_3c_4 & \Omega & \Omega & c_1c_2c_4c_5 \\
 & & & & 0 & \Omega & c_1c_2c_3c_5 & c_1c_2c_3c_5 & \Omega & \Omega \\
 & & & & & 0 & c_1c_2c_4 & c_2c_3c_4c_5 & \Omega & c_1c_2c_4c_5 \\
 & & & & & & 0 & c_1c_2c_3c_5 & c_1c_3c_4c_5 & c_1c_2c_4c_5 \\
 & & & & & & & 0 & c_1c_2c_3c_4 & c_1c_3c_4 \\
 & & & & & & & & 0 & c_1c_2c_3c_4 \\
 & & & & & & & & & 0
\end{bmatrix},$$

where $\Omega = c_1c_2c_3c_4c_5$. In the above matrix, $|C_D(3,4)| = 1$, $|C_D(4,6)| = 1$, we can get $Core(A) = \{c_1, c_5\}$. This means that a_1 and a_4 are the most important routing attributes. When $C_D(i,j)$ contains a_1 or a_4, then we set $C_D(i,j) = 0$. A new simple matrix can be obtained:

$$\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & 0 & 0 & 0 & c_2c_3c_4 & 0 & 0 & 0 \\
 & & & & 0 & 0 & 0 & 0 & 0 & 0 \\
 & & & & & 0 & 0 & 0 & 0 & 0 \\
 & & & & & & 0 & 0 & 0 & 0 \\
 & & & & & & & 0 & 0 & 0 \\
 & & & & & & & & 0 & 0 \\
 & & & & & & & & & 0
\end{bmatrix}$$

The reduced attribute set can be $(c_1 \wedge c_2 \wedge c_5) \vee (c_1 \wedge c_3 \wedge c_5) \vee (c_1 \wedge c_2 \wedge c_4)$. It means the reduction attributions can be $c_1 \wedge c_2 \wedge c_5$, $c_1 \wedge c_3 \wedge c_5$or $c_1 \wedge c_4 \wedge c_5$. We take $c_1 \wedge c_2 \wedge c_5$ as an example, then get the table 2.

Table 2. Reduction table

Link num.	Available bandwidth (C_1)	Propagation delay (C_2)	Connection possibility (C_5)	Link rank (d)
1	4	4	3	IV
2	5	3	2	III
3	1	4	3	IV
4	1	4	1	VI
5	6	2	2	II
6	3	4	1	IV
7	1	5	1	V
8	3	1	3	I
9	7	5	3	III
10	8	1	3	I

4.3 Logical Rules

We can mine ten rules though this sample.

$(1)(c_1,3) \wedge (c_2,1) \wedge (c_5,3) \rightarrow (d,I)$; $(2)(c_1,8) \wedge (c_2,1) \wedge (c_5,3) \rightarrow (d,I)$;
$(3)(c_1,6) \wedge (c_2,2) \wedge (c_5,2) \rightarrow (d,II)$; $(4)(c_1,5) \wedge (c_2,3) \wedge (c_5,2) \rightarrow (d,III)$;
$(5)(c_1,7) \wedge (c_2,5) \wedge (c_5,3) \rightarrow (d,III)$; $(6)(c_1,4) \wedge (c_2,4) \wedge (c_5,3) \rightarrow (d,IV)$;
$(7)(c_1,1) \wedge (c_2,4) \wedge (c_5,3) \rightarrow (d,IV)$; $(8)(c_1,3) \wedge (c_2,4) \wedge (c_5,1) \rightarrow (d,IV)$;
$(9)(c_1,1) \wedge (c_2,5) \wedge (c_5,1) \rightarrow (d,V)$; $(10)(c_1,1) \wedge (c_2,4) \wedge (c_5,1) \rightarrow (d,VI)$.

The mined rules can be applied to a great deal of data to distinguish the link into six ranks. The simplification of routing- reduction attributes needn't to select all routing-reduction attributes in the condition of keeping the consistence of routing-reduction table. That is to say, after some of the link attributes aren't selected,we found that if the link attribute with same row could decide the same decision as before. For compatible data table,the most excellent attribute set is mined by rough set theory.The links of the subnet are classified to six ranks.

5 Conclusion

The complication of traditional solving schemes to QoS Routing is NP-completeness. A data mining method for QoS routing based on rough set theory has been presented in this paper. Based on QOS routing concepts and rough set theory, we study data Mining algorithms for routing attribute and routing data reduction. The instance shown that the method is good at selecting the best route in network.

Acknowledgement

The authors would like to thank Dr. Y.Y. Yao and Dr. J.T. Yao for their help during my visit in Canada. The work is supported by the Natural Science Foundation of CSTC and CQUPT under Grant No. A200469.

References

1. Huishan Liu, Mingwei Xu, Ke Xu and Yong Cui, Research on Internetwork Routing Protocol: a Survey, Science of Telecommunications.2003.019: 28-32
2. A. Anthony Bruno, CCIE #2738, CCIE Routing and Switching Exam Certification Guide. 2003, 8
3. Pan Li, Hong Zheng and Saeid Nahavandi, The Application of Rough Set and Kohonen Network to Feature Selection for Object Extraction. Proceedings of the second International Conference on Machine Learning and Cybernetics. November 2003:1185-1189
4. E. Crawley, Argon Networks, R. Nair et al. RFC 2386, IETF, August 1998
5. Pawlak. Rough Set. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, Dordrecht, Boston, London, 1991
6. Pawlak. Rough Set Approach to Knowledge-Based Decision Support, European Journal of Operational Research,l: 48-57, 1999
7. G. Apostolopoulos, D. Williams, S. Kamat, R. Guerin et al. RFC 2676,IETF, August 1999
8. Yong Li, Classification of clients in client relationship management base on rough set theory, Proceedings of the 2th International Conference on Machine Learning and Cybernetics. November 2003:242-246
9. Zhiliang Wang, Wenbo Meng, Xuejing Gu et al. The Research of the Police GIS Spatial Data Classification Technology Based on Rough Set. Proceedings of the 4th Congress on Intelligent Control and Automation. June 2002:10-14
10. G. Y. Wang, H.Yu and D.C.Yang, Decision table reduction based on information entropy (in Chinese), Chinese Journal computers, Vol. 2, No.7,pp.759-766,2002
11. http://www.dataconnection.com/
12. Ying Zhang, Rough Set and Genetic Glgorithms in Path Planning of Robot.Proceeding of Second International Conference on Machine Learning and Cybernetics, Xi'an,Nov.2003:698-701

A Novel Data Mining Method Based on Ant Colony Algorithm

Weijin Jiang[1], Yusheng Xu[2], and Yuhui Xu[1]

[1] Department of computer, Hunan University of industry,
Zhuzhou 412008, P.R. China
jwjnudt@163.com
[2] College of Mechanical Engineering and Applied Electronics,
Beijing University of Technology, Beijing 100022, P.R. China
yshxu520@163.com

Abstract. Data mining has become of great importance owing to ever-increasing amounts of data collected by large organizations. This paper propose an data mining algorithm called Ant-Miner(I),which is based on an improvement of Ant Colony System(ACS) algorithm. Experimental results show that Ant-Miner(I) has a higher predictive accuracy and much smaller rule list than the original Ant-Miner algorithm.

Keywords: Data mining algorithm, Ant-Miner(I), Ant Colony System(ACS) algorithm.

1 Introduction

Mining information and knowledge from large data-base has been recognized by many researchers as a key research topic in database system and machine learning, and by many industrial companies as an important area with an opportunity of major revenues. One of the data mining tasks gaining significant attention is the classification rules extraction from databases. The goal of this task is to assign each case (object, record, and instance) to one class, out of a set of predefined classes, based on the values of some attributes for the case[1,2,3]. There are different classification algorithms used to extract relevant relationship in the data as decision trees which operate performing a successive partitioning of cases until all subsets belong to single class (Quin-lan,1986).This operating way is impracticable except for trivial data sets. There are have been many other approaches for data classification, such as statistical and roughest approaches (Ziarko,1994) and neural networks(Lu et al.,1995).Though these classification techniques are algorithmically strong they require significant expertise to work effectively and do not provide intelligible rules[4-6].

The classification problem becomes very hard when the number of possible different combinations of parameters is so high that algorithms based on exhaustive searches of the parameter space rapidly become computationally infeasible. The met heuristics algorithms based on nature, such as genetic algorithms(GA),neural networks, immune algorithm, are extremely appealing for the tasks of data mining. Thus it is natural to

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 284 – 291, 2005.

devote attention to a heuristic approach to find a "good-enough" solution to the classification problem. In recent years, Ant Colony System (ACS) algorithm (Dorigo & Maniezzo,1996) has emerged as a promising technique to discover useful and interesting knowledge from database. But the use of the algorithms for mining classification rule, in the context of data mining, is a research area where few people explored[7,8].

In this paper the objective is to investigate the capability of ACS algorithm to discover classification rule with higher predictive accuracy and much smaller rule list. The remainder of the paper is organized as follows. In the next section, we present the basic idea of the ACS algorithm. In section 3, the Ant-Miner algorithm is analyzed. In section 4,a novel improved algorithm called Ant-Miner(I) is introduced. Section 5 reports computation results when comparing with Ant-Miner across six data sets. Finally, the paper ends with conclusions and directions for future research.

2 The ACS Algorithm

The ACS algorithm is the recently developed, population-based approach. One of its main ideas is the indirect communication among the individuals of a colony of agents, called(artificial)ants, based on an analogy with trails of a chemical substance, called pheromone which real ants use for communication. The(artificial) pheromone trails are a kind of distributed numerical information which is modified by the ants to reflect their experience accumulated while solving a particular problem[9].

The ACS algorithm is basically a multi-agent system where low level interactions between single agents (i.e. artificial ants) result in a complex behavior of the whole ant colony, which has been shown to be both robust and versatile- in the sense that it has been applied successfully to a range of different NP-hard combinatorial optimization problems(Dorigo&Maniezzo,1996).But the use of algorithm for mining classification rule, in the context of data mining, is a research area where few people explored. Parpinelli is the first to propose Ant algorithm for mining classification rules, with the systems Ant-Miner(Parpinelli et al.,2002).Liu presented a modified version of Ant-Miner (i.e. Ant-Miner2)(Liu et al.,2002), where the core heuristic value was based on a simple density estimation heuristic. In addition, Liu further introduced another ant-based algorithm, which uses a different pheromone updating strategy and state transition rule(Liu et al.,2003). In the next section, we briefly analyze Ant-Miner and their key steps[10].

3 Ant-Miner

Algorithm I: Overview of Ant-Miner

```
Training set=all training cases;
 WHILE (No.of uncovered cases in the Training
           set>max_uncovered_cases)
   i=0;
   REPEAT
     i=i+1;
```

```
      Anti incrementally constructs a classification rule;
      Prune the just constructed rule;
      Update the pheromone of the trail followed by Anti ;
    UNTIL(i≥No_of_Ants) or (Anti constructed the same rule
                            as        the        previous
                            No_Rules_Converg-1 Ants);
   Select the best rule among all constructed rule;
   Remove the cases correctly covered by the selected rule
     from the training set;
 END WHILE.
```

A high-level description of Ant-Miner is shown in Algorithm I. Ant-Miner follows a sequential covering approach to discover a list of classification rules covering all, or almost all, the training cases. At first, the list of discovered rules is empty and the training set consists of all the training cases. Each iteration of the WHILE loop of Algorithm I, corresponding to a number of executions of the REPEAT-UNTIL loop, discovers one classification rule. This rule is added to the list of discovered rules and the training cases that are covered correctly by this rule (i.e. cases satisfying the rule antecedent and having the class predicted by the rule consequent) are re-moved from the training set. This process is per-formed iteratively while the number of uncovered training cases is greater than a user-specified thresh-old, called max_uncovered_cases.

3.1 Pheromone Initialization

The initial amount of pheromone deposited at each path is inversely proportional to the number of values of all attributes, and is defined by the following equation:

$$r_{ij}(t=0) = \frac{1}{\sum_{i=1}^{a} b_i} \tag{1}$$

where a is the total number of attributes; b_{i} is the number of possible values that can be taken on by attribute A_1.

3.2 State Transition Rule Construction

Let $term_{ij}$ be a rule condition of form $A_{\mathrm{i}}=V_{ji}$, where A_{i} is the ith attribute and V_{ij} is jth value of domain of A_{i}. The probability that $term_{ij}$ is chosen to be added to the current partial rule is given by the following equation:

$$p_{ij}(t) = \frac{\tau_{ij}(t)\cdot\eta_{ij}}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b_i}(\tau_{ij}(t)\cdot\eta_{ij})} \tag{2}$$

Where τ_{ij} (t) is the amount of pheromone associated with $term_{ij}$ at iteration t ; η_{ij} is the value of a problem-dependent heuristic function for $term_{ij}$, the function is based on information theory, and its definition is shown in the next section; a is the total number of attributes ; x_i is set to 1 if the attribute A_i was not yet used by the current ant, or to 0 otherwise; b_i is the number of values in the domain of the ith attribute.

3.3 Heuristic Function

For each *term*$_{ij}$ that can be added to the current rule, Ant-Miner computes the value η_{ij} of a heuristic function that is an estimate of the quality of this term, with respect to its ability to improve the predictive accuracy of the rule. This heuristic function is based on information theory and normalized. The proposed normalized, information-theoretic heuristic function is as follows:

$$\eta_{ij} = \frac{iog_2 k - H(W \mid A_i = V_{ij})}{\sum_{i=1}^{a} x_i \cdot \sum_{j=1}^{b_i} \log_2 k - H(W \mid A_i = V_{ij})} \tag{3}$$

where a, x_i and b_i have the same meaning as above illustration; k is the number of classes; W is the class attribute(i.e. the attribute whose domain consists of the classes to be predicted).

3.4 Rule Pruning

The main goal of rule pruning is to remove irrelevant terms that might have been unduly included in the rule. The basic idea of rule pruning is to iteratively remove one-term at a time from the rule while this process improves the quality of the rule, and the quality of the resulting rule is computed by the following equation:

$$Q = \frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN} \tag{4}$$

where TP is the number of cases covered by the rule that have the class predicted by the rule; FP is the number of cases covered by the rule that have a class different from the class predicted by the rule; FN is the number of cases that are not covered by the rule but have the class predicted by the rule; TN is the number of cases that are not covered by the rule and that do not have the class predicted by the rule.

3.5 Pheromone Updating Rule

Whenever an ant constructs its rule and rule is pruned, the amount of pheromone in all segments of all paths must be updated. Pheromone updating for a *term*$_{ij}$ performed according to the following equation:

$$\tau_{ij}(t+1) = \tau_{ij}(t) + \tau_{ij}(t) \cdot Q, \forall i, j \in R \tag{5}$$

where R is the set of terms occurring in the rule constructed by the ant at iteration t. Meanwhile, the effect of pheromone evaporation for unused terms is achieved by dividing the value of each $\tau_{ij}(t)$ by the summation of all $\tau_{ij}(t)$.

4 Our Proposed Mining Algorithm: Ant-Miner(I)

Although Ant-Miner is a flexible and robust classification mining method, the system has some drawbacks:1)state transition rule computation is very complex, and lacks of balancing between exploration and exploitation;2) if rule quality measure *Q* is very small, evolutionary process will become stagnant;3) the $H(W|A_i{=}V_{ij})$ of *term*$_{ij}$ is always the same when computing heuristic function η_{ij}, regard-less of the contents of the rule in which the term occurs. Which is impossible in real application. In order to overcome above drawbacks, we propose an improved classification rule mining algorithm, called Ant-Miner(I). In the following we discuss improved state transition rule, pheromone updating rule and heuristic function.

4.1 Simple State Transition Rule

In fact, state transition rule is determined by heuristic function η_{ij} and pheromone $\tau_{ij}(t)$, if we use addition instead of multiplication, and use relative weight to adjust their roles, then we may provide a simple way to compute state transition rule. In addition, in order to provide a direct way to balance between exploration of new terms and exploitation of a priori and accumulated knowledge about the classification, we introduce a parameter $q_0 \in [0,1]$ q and a random number q which is uniformly distributed in [0,1].Then, our proposed state transition rule in choosing *term*$_{ij}$ is as follows:

```
IF q≦q_0
   THEN choose term_ij
   ELSE choose term_ij according to probability
```

$$P_{IJ}(t) = \frac{\beta \cdot \tau_{ij}(t) + (1-\beta)\cdot \eta_{ij}}{\sum_{i=1}^{a}\sum_{j=1}^{b_i} x_i \cdot (\beta \cdot \tau_{ij}(t) + (1-\beta)\cdot \eta_{ij})} \tag{6}$$

```
END IF.
```

where a, x_i, b_i, $\tau_{ij}(t)$ and η_{ij} have the same meaning as above illustration; is parameter that controls the relative importance of trail versus visibility. Therefore the transition probability is a trade-off between visibility (which says that close terms should be chosen with high probability, thus implementing a greed constructive heuristic) and trail intensity at time t (that says that if on *term*$_{ij}$ there has been a lot of traffic then it is highly desirable, thus implementing the autocatalytic process).

4.2 Our Pheromone Updating Rule

When an ant constructs its rule and rule is pruned, the amount of pheromone in all segments of all paths must be updated according to pheromone updating rule in Ant-Miner system. But, if rule quality measure *Q* is very small (near to zero), evolutionary process will become stagnant. In order to overcome the above shortcoming, we introduce the following pheromone updating rule:

$$\tau_{ij}(t+1) = (1-\rho)\cdot\tau_{ij}(t) + (\lambda e^{-\rho} + Q)\cdot\tau_{ij}(t), \forall i, j \in R \tag{7}$$

where Q is the quality measure of constructed rule; ρ is an evaporation rate, its value in (0,1); λ is an adjusting parameter, which value is in (0.8,1).In this paper, we set λ =0.85.

Our proposed pheromone updating rule concurrently realizes pheromone accumulation and evaporation, and avoid stagnation.

4.3 Our Heuristic Function

Heuristic function computation of Ant-Miner is based on information entropy and normalization, but authors of Ant-Miner(Parpinelli et al.,2002) consider the $H(W|A_i{=}V_{ij})$ of *term*$_{ij}$ is always the same when computing heuristic function η_{ij} , regardless of the contents of the rule in which the term occurs. Consequently, the ants likely converge to a single constructed rule too quickly. This leads premature and a failure to produce alternative potential rules. In order to overcome above shortcomings, we introduce a simple Heuristic function, which definition is as follows:

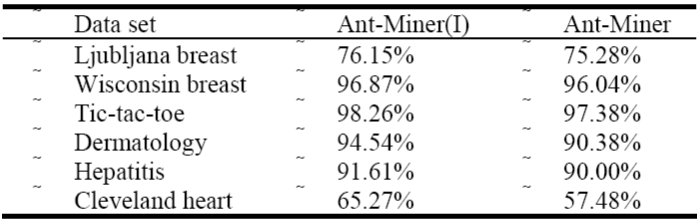
$$\eta_{ij} = \frac{-\sum_{w=1}^{k} freqT_{ij}^{w} \cdot \log_2 freqT_{ij}^{w}}{|T_{ij}|} \tag{8}$$

where k is the number of classes; $|T_{ij}|$ is the total number of cases in partition T_{ij};$freqT_{ij}^{w}$ is the number of cases in partition T_{ij} with class w.

5 Experimental Results Comparison

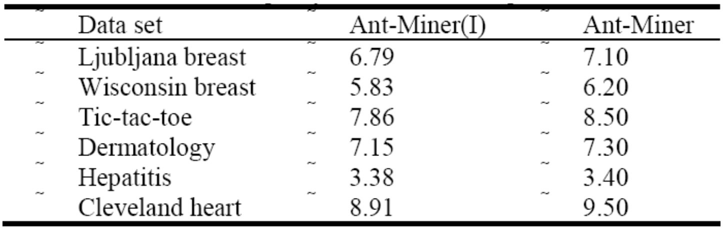
To evaluate performance of our proposed Ant-Miner (I) system, we have conducted experiment with it on a number of datasets taken from the UCI repository (Hettich&Bay,1999).These data sets have been widely used in other comparative studies. To have a quick turn-around time and comparison for our experiments, we used the same subset of data sets as Parpinelli(2002).Because Ant-Miner(I) mines rules referring only to categorical attributes, we have discredited continuous attributes by the C4.5-Disc discretization method(Weiss&KulIkowski,1991) in a preprocessing step. All the results of the comparison were obtained using a Pentium 4 PC (CPU 2.2 G HZ,RAM 256MB).

Table 1. Predictive accuracy comparison



Data set	Ant-Miner(I)	Ant-Miner
Ljubljana breast	76.15%	75.28%
Wisconsin breast	96.87%	96.04%
Tic-tac-toe	98.26%	97.38%
Dermatology	94.54%	90.38%
Hepatitis	91.61%	90.00%
Cleveland heart	65.27%	57.48%

Table 2. Simplicity of the rule list comparison



Data set	Ant-Miner(I)	Ant-Miner
Ljubljana breast	6.79	7.10
Wisconsin breast	5.83	6.20
Tic-tac-toe	7.86	8.50
Dermatology	7.15	7.30
Hepatitis	3.38	3.40
Cleveland heart	8.91	9.50

We have evaluated the performance of Ant-Miner (I) by comparing it with Ant-Miner. The first experiment was carried out to compare predictive accuracy of discovered rule lists by well-known ten-fold cross-validation procedure (Kohavi & Sa-hami,1996).Each data set is divided into ten partitions, each method is run ten times, using a different partition as test set and the other nine partitions as the training set each time. The predictive accuracies of the ten runs are averaged as the predictive accuracy of the discovered rule list. Table 1 shows the results comparing the predictive accuracy of Ant-Miner(I) and Ant-Miner. It can be seen that predictive accuracy of Ant-Miner(I) is higher than that of Ant-Miner[11,12,13].

In addition, we compared the simplicity of the discovered rule list by the number of discovered rules and the average number of terms (conditions) per rule. The results comparing the simplicity of the rule lists discovered by Ant-Miner (I) and Ant-Miner are shown in Table 2. As shown in this table, Ant-Miner(I) mined rule lists a little smaller than the rule list mined by Ant-Miner.

In summary, although Ant-Miner(I) need to set more parameter than Ant-Miner, taking into account both the predictive accuracy and rule list simplicity criteria, our proposed Ant-Miner(I) is rather competitive.

6 Conclusions

We have presented Ant-Miner(I) for data mining, a new method for mining classification rule based on an improvement of Ant-Miner. We have compared the performance of Ant-Miner(I) and Ant-Miner in public domain data sets. Experimental results show that Ant-Miner(I) has a higher predictive accuracy and a little smaller rule list than Ant-Miner. Because the application of ACS algorithm in classification rule mining is still in infant periods, in future works, our further research directions are as follows. We plan to make further experiments to understand sys-tem parameters influence on the performance of Ant-Miner(I), so that, we can set appropriate parameter combinations in term of different classification problems.

Acknowledgments

The paper was supported by the National Natural Science Foundation of China (No. 60373062), and the Natural Science Foundation of Hunan Province of China (No. 04JJ3052).

References

1. Dorigo, M.& Maniezzo,V.: The ant system: optimization by a colony of cooperating agents. IEEE Transactions on Sys-tem, Man, and Cybernetics (1996)26(1):1-13
2. Kohavi, R.& Sahami,M.: Error-based and entropy-based discretization of continuous features. Proceeding of second international conference knowledge discovery and data mining,2-4 August 1996,Menlo Park, CA,USA
3. Hettich, S.& Bay,S.D.: The UCI KDD ar-chive.Url:http://kdd.ics.uci.edu(1999)
4. Lu,H.,R.Setiono,R.& Liu, H. : NeuroRule: a connectionist approach to data mining. Proceeding of the 21st international conference on very large data bases, Zurich, Switzer-land,11-15 September 1995. San Mateo, CA: Morgan Kaufmann
5. Liu,B.,Abbass,H.A.& Mckay,B.: Classification rule discovery with ant colony optimization. Proceeding of the IEEE/WIC international conference on intelligent agent technology, 13-17 October 2003, Beijing, China
6. Liu, B., Abbass, H.A.& Mckay,B.: Density_based heuristic for rule discovery with Ant-Miner. The 6th Australia-Japan joint workshop on intelligent and evolutionary system,30 November-1 December 2002, University House, Australia
7. Quinlan,J.R.: Induction of decision trees. Machine Learning (1986) (1):81-106
8. Parpinelli, R.S., Lopes, H.S.& Freitas A. A. : Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computing (2002)6(4):321-332
9. Ziarko,W.: Rough Sets, Fuzzy Set and Knowledge Discovery. Berlin: Springer-Verlag(1994)
10. Weiss,S.M.& KulIkowski, C. A.: Computer Systems that Learn . San Mateo, CA: Morgan Kaufmann(1991)
11. Weijin Jiang: Hybird Genetic algorithm research and its application in problem optimization. Proceedings of 2004 International Conference on Manchine Learning and Cybernetics, (2004)222-227
12. Weijin Jiang: Research on Extracting Medical Diagnosis Rules Based on Rough Sets Theory. Journal of computer science, (2004)31(11): 93-96
13. Weijin Jiang: Research on Optimize Prediction Model and Algorithm about Chaotic Time Series, Wuhan University Journal of Natural Sciences,(2004) 9(5): 735-740

Context-Sensitive Regression Analysis for Distributed Data*

Yan Xing[1], Michael G. Madden[2], Jim Duggan[2], and Gerard J. Lyons[2]

[1] Faculty of Automation, Guangdong University of Technology,
Guangzhou, 510090, China
[2] IT Department, National University of Ireland, Galway, Ireland
{michael.madden, jim.duggan, gerard.lyons}@nuigalway.ie

Abstract. A precondition of existing ensemble-based distributed data mining techniques is the assumption that contributing data are identically and independently distributed. However, this assumption is not valid in many virtual organization contexts because contextual heterogeneity exists. Focusing on regression tasks, this paper proposes a context-based meta-learning technique for horizontally partitioned data with contextual heterogeneity. The predictive performance of our new approach and the state of the art techniques are evaluated and compared on both simulated and real-world data sets.

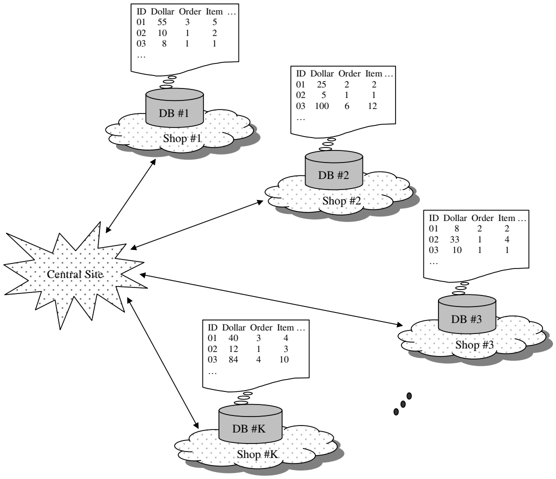
1 Introduction

A virtual organization is “a temporary network of companies that come together quickly to exploit fast changing opportunities...” [1]. As shown in Fig. 1, assume there is a commercial virtual organization consisting of several independent but cooperating shops. Each shop stores its customer and transaction data locally. The data mining task in the organization is to model customer purchase behavior, e.g., estimate the f in $Dollar = f\,(order, item, \ldots)$, and then use it to predict customers’ future purchase patterns.

In this scenario, the detailed business data are not allowed to be shared for reasons of commercial confidentiality. Moreover, there may be differences among the individual shops. For example, the individual shops may sell different products, adopt different price strategies, be located at different areas, employ different strategies of advertisement, etc. This kind of difference is termed “[illegible]”. Therefore from the point of view of data mining, this scenario illustrates the two main characteristics of a virtual organization: local data storage and potential contextual heterogeneity. The former requires that data mining has to be performed in a distributed environment. The latter implies that the data across the partners’ sites are not identically and independently distributed (IID). In essence, it is required that data mining algorithms can deal with non-IID dispersed data.

* The support of the Informatics Research Initiative of Enterprise Ireland is gratefully acknowledged.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 292–299, 2005.



ID	Dollar	Order	Item …
01	55	3	5
02	10	1	2
03	8	1	1
…			

ID	Dollar	Order	Item …
01	25	2	2
02	5	1	1
03	100	6	12
…			

ID	Dollar	Order	Item …
01	8	2	2
02	33	1	4
03	10	1	1
…			

ID	Dollar	Order	Item …
01	40	3	4
02	12	1	3
03	84	4	10
…			

Fig. 1. Diagram of a commercial virtual organization

Focusing on regression tasks, this paper proposes a context-based Meta-Learning approach for horizontally partitioned data in virtual organizations. In the following sections, we first review the state of the art techniques and highlight their limitations. Secondly, we introduce a two-level hierarchical model to model distributed data with contextual heterogeneity. Thirdly, we propose an ensemble-based algorithm within the framework of the two-level model. Fourthly, we evaluate and compare the predictive performance of the new algorithm and the state of the art techniques empirically. Finally, we summarize this work and discuss future research directions.

2 The State of the Art

Distributed data mining (DDM) is concerned with data mining algorithms in distributed environments [2]. In DDM, the dominant technique for horizontally partitioned data is the traditional meta-learning (TML) framework and follows three main steps: 1) Local analysis: local models are generated at individual sites in parallel using the same learning algorithm. 2) Communication: all the local models are collected at a central site (some approaches also require a separate validation set of data to be collected). Then meta-level data are generated. 3) Meta-level analysis: the final model is created by an ensemble scheme at the central site. Approaches within the TML framework include the meta-learning approach of Chan and colleagues [3], distributed learning with knowledge probing (DLKP) [4], the approach of a data fusion system [5] and a distributed bagging-like approach [6], etc.

The two main advantages of TML are that parallelism is used and communication is minimized [2]. However, in the communication and meta-level analysis steps of TML, information about the sites themselves is not considered. For those approaches that need validation sets, information about the sites where the validation data originated is not considered either. This means that TML assumes that the contributing data are IID [7, 8, 9] and contextual heterogeneity does not exist or is negligible. Therefore, in a virtual organization such as the commercial network introduced in Section 1, TML can only deal with the characteristic of local data storage. It is unable to solve the problem of contextual heterogeneity. In the following sections, we will propose a context-based meta-learning technique that can address both of these important issues.

3 Model of Horizontally Partitioned Data with Contextual Heterogeneity

Consider the commercial virtual organization shown in Fig. 1 consisting of K local sites. The data set stored at the kth site consists of data $\{(\mathbf{x}_{ik}, y_{ik}), i = 1, 2, \ldots, N_k\}$, where the ys are numeric response variables, the $\mathbf{x}$s are vectors of explanatory variables where $\mathbf{x}_{ik} = \langle x_{1ik}, x_{2ik}, \ldots, x_{Mik} \rangle$ and M is the number of explanatory variables, N_k is the sample size of the data and $k = 1, 2, \ldots, K$. Assuming that contextual heterogeneity derives from essentially random differences among the sites, data within a local site can be regarded as conditional IID. Accordingly a two-level hierarchy, as shown in Fig. 2, can be obtained. Thus, an existing statistical technique termed hierarchical modelling (HM) [10] can be used to model the non-IID contributing data.

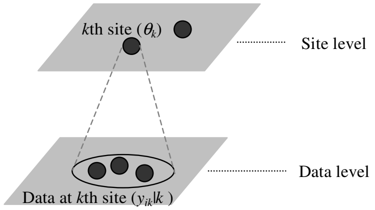


Fig. 2. Hierarchy of non-IID dispersed data

Assuming that, at the bottom level, data y_{ik} at the kth site has a Normal distribution[1] with mean θ_k and variance σ^2; at the top level, the group means of the sites $\theta_1, \ldots, \theta_K$, which are used to represent contextual information, are also random variables having a Normal distribution[2] with mean θ ($\theta = f(\mathbf{x}_{ik})$) and variance τ^2, we can model the non-IID contributing data as:

[1] The assumption of Normal distribution is required in statistical regression analysis. However, for some non-parametric techniques of data mining such as regression trees, this assumption is not strictly required.

[2] The assumption of the contextual heterogeneity to be Normally distributed is a conventional choice in statistical hierarchical modelling [11].

$$\begin{cases} \text{site level}: \ \theta_k \overset{IID}{\sim} N\left(\theta, \tau^2\right) \\ \text{data level}: (y_{ik}|k) \overset{IID}{\sim} N\left(\theta_k, \sigma^2\right) \end{cases} \tag{1}$$

or

$$\begin{cases} y_{ik} = \theta_k + e_{ik} = \theta + t_k + e_{ik} = f\left(\mathbf{x}_{ik}\right) + t_k + e_{ik} \\ e_{ik} \overset{IID}{\sim} N\left(0, \sigma^2\right), t_k \overset{IID}{\sim} N\left(0, \tau^2\right) \end{cases} \tag{2}$$

where e_{ik} denotes the residual of the data at the kth site, t_k denotes the residual of the contextual information across the local sites, t_k and e_{ik} are independent, and τ^2 represents the contextual heterogeneity. When $\tau^2 = 0$, (1) and (2) can be rewritten as the models of IID distributed data that TML deals with [9]. Intra-Class Correlation (ICC) is a measure of how much of the total variance of a model is caused by contextual heterogeneity and calculated by $ICC = \tau^2/\left(\tau^2 + \sigma^2\right)$.

For distributed data modelled as (1) or (2), the main task of regression is to estimate not only the global regression model $f\left(\mathbf{x}_{ik}\right)$, but also the quantity of contextual heterogeneity τ^2. In the following section, we propose a context-based meta-learning (CML) algorithm that has the capability of estimating both of these.

4 Context-Based Meta-learning (CML)

The principal idea of CML is that, based on the two-level hierarchical model for non-IID dispersed data formulated as (1) and (2), CML retains the core idea of ensemble learning from TML, and simultaneously adds the capability of dealing with contextual heterogeneity explicitly.

Since local models are generated only from local data, the local model $f_k\left(\mathbf{x}_{ik}\right)$ generated at the kth site can be regarded as the estimate of θ_k. Given the kth site and $\mathbf{x}_{ik}$, we have:

$$f_k\left(\mathbf{x}_{ik}\right) \approx \theta_k = f\left(\mathbf{x}_{ik}\right) + t_k \tag{3}$$

If E_k denotes expectation over k, then the ensemble model $f_A\left(\mathbf{x}_{ik}\right)$ is:

$$f_A\left(\mathbf{x}_{ik}\right) = E_k\left[f_k\left(\mathbf{x}_{ik}\right)\right] \approx f\left(\mathbf{x}_{ik}\right) + E_k\left[t_k\right] \tag{4}$$

Since $E_k\left[t_k\right] = 0$, the ensemble model $f_A\left(\mathbf{x}_{ik}\right)$ is the estimation of the real global model $f\left(\mathbf{x}_{ik}\right)$. Using $f_A\left(\mathbf{x}_{ik}\right) \approx f\left(\mathbf{x}_{ik}\right)$, we obtain:

$$\hat{t}_k = E_{within-k}\left[f_k\left(\mathbf{x}_{ik}\right) - f_A\left(\mathbf{x}_{ik}\right)\right] = \frac{1}{N_k}\sum_{i=1}^{N_k}\left[f_k\left(\mathbf{x}_{ik}\right) - f_A\left(\mathbf{x}_{ik}\right)\right] \tag{5}$$

where $E_{within-k}$ denotes the expectation over all the data at the kth site.

With (5), $\hat{t}_k$ will never be exactly equal to zero even though the real contextual residual t_k is zero. Therefore, we use a two-tailed , -test to check the null hypothesis $H_0 : t_k = 0$, and estimate the contextual residual of the kth site by:

$$\hat{t}_k = \{\begin{matrix} 0, \text{if } H_0 \text{ accepted given } \alpha; \\ \hat{t}_k, \text{if } H_0 \text{ rejected given } \alpha. \end{matrix} \tag{6}$$

where α is the level of significance. The quantity of contextual heterogeneity can be estimated with $\hat{\tau}^2 = E_k\left[\hat{t}_k^2\right]$.

The procedure of CML algorithm follows seven key steps:

1. At local sites, generate local models (i.e., the data-level models) in parallel using the same learning algorithm
2. At a central site, collect all the local models and then broadcast them to all the local sites
3. At each local site, generate meta-level data with the local models
4. At each local site, generate the ensemble model from the meta-level data using an ensemble scheme
5. At each local site, calculate its contextual residual with (5) and (6)
6. At each local site, generate the final regression model of this site by: $f_{prediction}\left(\mathbf{x}_{ik}\right) = \hat{f}\left(\mathbf{x}_{ik}\right) + \hat{t}_k = f_A\left(\mathbf{x}_{ik}\right) + \hat{t}_k$
7. At the cental site, collect all the contextual residuals and then calculate the contextual variance

In the procedure of CML, the computational complexity of local model induction dominates the entire computation. The network traffic among all the sites is proportional to K^2 for local model sharing and K for transferring information of contextual heterogeneity. So the total traffic is proportional to K^2.

5 Experimental Evaluation

In this section, we empirically evaluate and compare the predictive performance of TML and CML.

5.1 Data Sets

For our experiments, we use three simulated and one real-world data sets. The three artificial data sets are used to simulate real-world distributed scenarios, where the range of ICC is $[0, 1)$. They are generated by the following models:

Non-linear #1:
$$\begin{cases} y_{ik} = 10\sin\left(\pi x_{1ik}x_{2ik}\right) + 20\left(x_{3ik} - 0.5\right)^2 + 10x_{4ik} + 5x_{5ik} + e_{ik} + t_k \\ e_{ik} \overset{IID}{\sim} N\left(0,1\right), t_k \overset{IID}{\sim} N\left(0,\tau_1^2\right) \end{cases}$$

Nonlinear #2:
$$\begin{cases} y_{ik} = \left(x_{1ik}^2 + \left(x_{2ik}x_{3ik} - \frac{1}{x_{2ik}x_{4ik}}\right)^2\right)^{\frac{1}{2}} + e_{ik} + t_k \\ e_{ik} \overset{IID}{\sim} N\left(0,\sigma_2^2\right), t_k \overset{IID}{\sim} N\left(0,\tau_2^2\right) \end{cases}$$

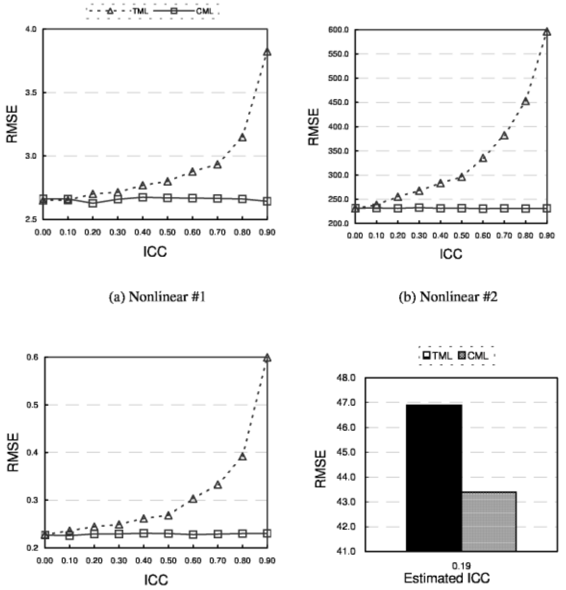
Nonlinear #3:
$$\begin{cases} y_{ik} = \tan^{-1}\left(\frac{x_{2ik}x_{3ik} - \frac{1}{x_{2ik}x_{4ik}}}{x_{1ik}}\right) + e_{ik} + t_k \\ e_{ik} \overset{IID}{\sim} N\left(0,\sigma_3^2\right), t_k \overset{IID}{\sim} N\left(0,\tau_3^2\right) \end{cases}$$

For nonlinear #1, each of $x_1, x_2, \ldots, x_{10}$ is uniformly distributed over $[0, 1]$. For the others, x_1, x_2, x_3, x_4 are uniformly distributed as $0 \leq x_1 \leq 10$, $20 \leq \frac{x_2}{2\pi} \leq 280$, $0 \leq x_3 \leq 1$, $1 \leq x_4 \leq 11$ and the parameter σ_2^2, σ_3^2 is selected to give a signal/noise ratio of $3:1$. For all of the simulated data sets, the total number of sites is set as $K = 10$. At each site, the sample size is $N_k = 1200$. The values of $\tau_1^2, \tau_2^2, \tau_3^2$ are adjusted so that $ICC = 0.0, 0.1, \ldots, 0.9$ can be obtained.

The real-world data set stores 36 months of business information from a catalog-based retail organization [12], which consists of $K = 9$ shops and $N = 88,002$ customers in total. The number of customers in each shop N_k ranges from 767 to 55,065. The information in the data set includes life-to-date orders, money spent, items bought, recency of the first and latest purchases, payment methods and very minimal demographics for each consumer. The organization would like to build customer behavior models (how much money a customer spends on average per purchase) to predict customers' future buying patterns.

5.2 Experimental Results

For our experiments, we have implemented TML and CML in Java, using a regression trees induction algorithm for the local learning and un-weighted averaging as the ensemble scheme. For the , -tests in CML, $\alpha = 0.05$. At each site, one sixth (for the artificial data) or two thirds (for the commercial data) of the data is used for training and the rest for testing. The process of learning and testing is repeated 20 times and the final results are the average.



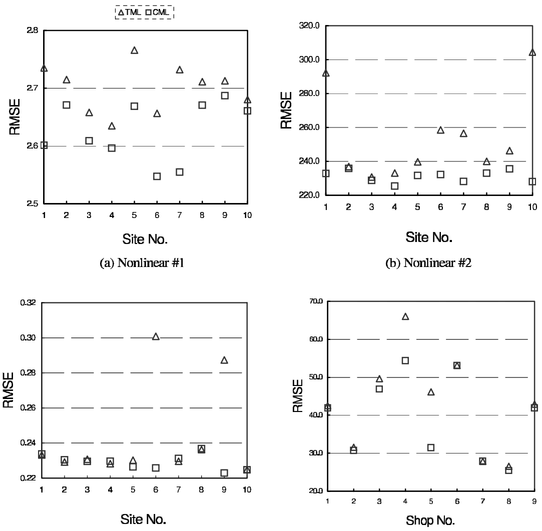
(a) Nonlinear #1

(b) Nonlinear #2

(c) Nonlinear #3

(d) Commercial data

Fig. 3. Global prediction errors



(a) Nonlinear #1

(b) Nonlinear #2

(c) Nonlinear #3

(d) Commercial data

Fig. 4. Local prediction errors ($ICC = 0.20$ for simulated data sets)

For non-IID dispersed data, there are two aspects of predictive performance: global predictive performance (measured with global prediction error) and Local predictive performance (measured with local prediction error). The former refers to the total predictive performance in the entire virtual organization, while the latter refers to the predictive performance at an individual site. Fig. 3 and Fig. 4 show the global prediction errors (Root Mean Squared Error, RMSE) and the local prediction errors on the four data sets respectively.

From Fig. 3, it can be seen that, when the contextual heterogeneity is not negligible, the global predictive performance of CML is better than that of TML. From Fig. 4, it can be seen that, for those sites with larger contextual residuals (for the simulated data, contextual residuals are generated randomly with the simulation models; for the commercial data, contextual residuals are estimated by CML) such as 1^{st}, 5^{th}, 6^{th} and 7^{th} sites in (a); 1^{st}, 6^{th}, 7^{th} and 10^{th} sites in (b); 6^{th} and 9^{th} sites in (c); and 3^{rd}, 4^{th} and 5^{th} shops in (d), TML behaves worse than CML. For those sites with smaller contextual residuals, both the algorithms behave comparably well. Therefore, we can conclude that CML outperforms the traditional meta-learning techniques on non-IID dispersed data.

6 Summary and Future Work

In the domain of virtual organizations, contributing data are non-IID because contextual heterogeneity exists across different sites. Conventional ensemble-based DDM approaches such as TML cannot be directly adopted because its predictive performance worsens as the quantity of contextual heterogeneity increases. By explicitly modelling non-IID dispersed data with a two-level hierarchical model, this paper proposes a context-based meta-learning approach (CML) for distributed regression. The experimental results demonstrate that CML can successfully address both of the characteristics of a virtual organization introduced in Section 1.

At present we only use a two-level hierarchical model where σ^2 and τ^2 are constant. In some applications, more complex models may be required. Another limitation of CML is that we neglect the variability of the ensemble models built at local sites. If the variability is significant, the estimation of the contextual residuals will suffer. Future versions of CML will address these two aspects.

References

1. Byrne, J.: The virtual corporation. Business Week (1993) 36–40
2. Park, B.H., Kargupta, H. In: Distributed Data Mining: Algorithms, Systems, and Applications. IEA (2002) 341–358
3. Chan, P.K., Fan, W., Prodromidis, A.L., Stolfo, S.J.: Distributed data mining in credit card fraud detection. IEEE Intelligent Systems **14** (1999) 67–74
4. Guo, Y., Sutiwaraphun, J.: Distributed learning with knowledge probing: A new framework for distributed data mining. In Kargupa, H., Chan, P., eds.: Advances in Distributed and Parallel Knowledge Discovery. MIT/AAAI Press (2000) 113–131
5. Gorodetski, V., Skormin, V., L.Popyack, O.Karsaev: Distributed learning in a data fusion system. In: Proceedings of Conference of the World Computer Congress (WCC-2000) and Intelligent Information Processing (IIP2000), Beijing (2000) 147–154
6. Chawla, N.V., Moore, T.E., Hall, L.O., Bowyer, K.W., Kegelmeyer, W.P., Springer, C.: Distributed learning with bagging-like performance. Pattern Recognition Letters **24** (2003) 455–471
7. Wirth, R., Borth, M., Hipp, J.: When distribution is part of the semantics: A new problem class for distributed knowledge discovery. In: Proceedings of PKDD'01 Workshop on Ubiquitous Data Mining for Mobile and Distributed Environments, Freiburg, Germany (2001) 56–64
8. Xing, Y., Madden, M.G., Duggan, J., Lyons, G.J.: Distributed regression for heterogeneous data sets. In: Advances in Intelligent data Analysis V (Lecture Notes in Computer Science 2810), Berlin, Germany, Springer (2003) 544–553
9. Xing, Y.: Context-based Numeric Prediction for Distributed Data with Contextual Heterogeneity. PhD thesis, National University of Ireland, Galway, Ireland (2004)
10. Goldstein, H.: Multilevel Statistical Models. 2nd edn. ARNOLD (1995)
11. Draper, D.: Bayesian hierarchical modeling.
    Online: http://citeseer.nj.nec.com/draper00bayesian.html (2000)
12. DMEF: DMEF academic data sets. The Direct Marketing Educational Foundation INC. New York, USA. (2002)

Customer Churn Prediction Using Improved One-Class Support Vector Machine

Yu Zhao, Bing Li, Xiu Li, Wenhuang Liu, and Shouju Ren

Cims Research Center, Automation Department,
Tsinghua University, Beijing 100084, China
zhaoyu01@mails.tsinghua.edu.cn

Abstract. Customer Churn Prediction is an increasingly pressing issue in today's ever-competitive commercial arena. Although there are several researches in churn prediction, but the accuracy rate, which is very important to business, is not high enough. Recently, Support Vector Machines (SVMs), based on statistical learning theory, are gaining applications in the areas of data mining, machine learning, computer vision and pattern recognition because of high accuracy and good generalization capability. But there has no report about using SVM to Customer Churn Prediction. According to churn data set characteristic, the number of negative examples is very small, we introduce an improved one-class SVM. And we have tested our method on the wireless industry customer churn data set. Our method has been shown to perform very well compared with other traditional methods, ANN, Decision Tree, and Naïve Bays.

1 Introduction

According to Don Peppers and Martha Rogers, the marketing experts, most efficiency, and the incapacity of searching a specific item, enterprises average lost 25% customers annually. However, the cost of obtaining a new customer is five times higher than maintaining an existing customer [1].

In many industry fields, churn - that can be looked as the customer's decision to end the relationship and switch to another company - has become a major concern.

The churn rate for U.S. mobile carriers is 2 % to 3 % monthly, a major expense for the companies, which spend $400 to $500 to sign a single customer who typically generates about $50 in monthly revenue. Companies are now beginning to realize just how important customer retention is. In fact, one study finds that "the top six US wireless carriers would have saved $207 million if they had retained an additional 5% of customers open to incentives but who switched plans in the past year" [2]. Over the next few years, the industry's biggest marketing challenge will be to control churn rates by identifying those customers who are most likely to leave and then taking appropriate steps to retain them. The first step therefore is predicting churn likelihood at the customer level.

The Customer Churn Prediction problem has two major characteristics:

The first is that the number of churn customers (the negative examples) is small (2% in the total examples);

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 300–306, 2005.

The second is accuracy. Consequently, for a carrier with 1.5 million subscribers, improving the monthly prediction accuracy rate 1% would yield an increase in annual earnings of at least $54 million.

Customer Churn Prediction generally can be considered as a binary classification problem, distinguishing between normal and churn. The standard support vector machine (SVM) is a classifier that finds a maximal margin separating two classes of data. There have been a lot of successful applications about that. But the data of Customer Churn Prediction are very special: the normal dataset is much larger than the abnormal. Therefore, the standard SVM does not work well on our task. We present an improving SVM method, which is based on one-class SVM described in [3] by Bernhrd Scholkopf et al. We used dataset provided by a wireless telecom company and included more than 150 variables describing more than 100,000 customers. We have performed experiments on the improved one-class SVM with the various kernel functions, and have compared the performance of SVM and other normal methods (such as ANN, Decision Tree, and Naïve Bays).

The rest of this paper is organized as follows. A brief description of the Customer Churn Prediction and SVM model will be described in Section 2. The improved one-class SVM will be introduced in Section 3. The dataset preparation and various experiments of improved one-class SVM and their results are presented in Section 4. Some concluding remarks and future work are given in Section 5.

2 Customer Churn and SVM Model

Customer churn – the propensity of customers to cease doing business with a company in a given time period – has become a significant problem for many firms. These include publishing industry, investment services, insurance, electric utilities, health care providers, credit card providers, banking, Internet service providers, telephone service providers, online services, and cable services operators.

There are numerous predictive modeling techniques for predicting customer churn. These vary in terms of statistical technique (e.g., neural nets versus logistic regression), variable selection method (e.g., theory versus stepwise selection), number of variables included in the model, and time spent in total on the modeling exercise as well as how a given time budget is allocated across various tasks in the model-building process [4].

SVM algorithm developed by Vapnik [5] is based on statistical learning theory. In some classification cases, we try to find an optimal hyper-plane that separates two classes. When the two classes of points in the training set can be separated by a linear hyper-plane, it is natural to use the hyper-plane that separates the two groups of points in the training set by the largest margin. In order to find an optimal hyper-plane, we need to minimize the norm of the vector w, which defines the separating hyper-plane. This is equivalent to maximizing the margin between two classes. [6]

Customer Churn is a problem of classification between "churn" and "no churn". But when the number of the negative examples is too small, the generalization performance of SVM classifier must be weak, and the error rates is proved unsatisfactory.

3 Improved One-Class SVM

One-class SVM: Bernhard Scholkopf [7] et al. suggested a method of adapting the SVM methodology to the one-class classification problem. Essentially, after transforming the feature via a kernel, they treated the origin as the only member of the second class. By introducing "relaxation parameters", they separate the image of the one class from the origin.

Li present a One-Class SVM for anomaly detection [8]. The basic idea is to work first in the feature space, and assume that not only is the origin in the second class, but also that all data points "close enough" to the origin are to be considered as outliers or anomaly data points. If the input data match the selected samples, then they are regarded as anomaly data, i.e., that belongs to the anomaly class.

Here we introduce an improved One-Class SVM to predict customer churn:

Suppose we are given the training data:

$\{(x_1, y_1), (x_1, y_1), \cdots (x_1, y_1)\}$, where $x \in R^N, y \in \{-1, +1\}$, and R^N is the feature space. This leads to the following quadratic programming problem:

$$\begin{aligned} &\min(R^2 + \sum_{y_i=1} C_+\xi_i + \sum_{y_i=-1} C_-\xi_i) \\ &s.t. \quad y_i(\|\Phi(x_i) - \alpha\|^2 - R^2) \le \xi_i \\ &\xi_i \ge 0, 1 \le i \le l \end{aligned} \tag{1}$$

Where ξ_i are slack variables that are penalized in the objective function. The goal of introducing the slack variables is to allow some error during the training, where α and R are the center and radius of the hyper-sphere respectively, and $C_+ = \frac{l_-}{l_+ + l_-}C,\ C_- = \frac{l_+}{l_+ + l_-}C$ are penalty parameter. l_+ is the number of the positive examples. l_- is the number of the negative examples.

The corresponding dual is:

$$\begin{aligned} &\min(\sum_{i=1}^{l}\sum_{j=1}^{l}\alpha_i\alpha_j y_i y_j(\Phi(x_i)\cdot\Phi(x_j)) - \sum_{i=1}^{l}\alpha_i y_i(\Phi(x_i)\cdot\Phi(x_j))) \\ &s.t. \ \sum_{i=1}^{l}\alpha_i y_i = 1, \\ &\quad 0 \le \alpha_i \le C_+, \quad y_i = 1, \\ &\quad 0 \le \alpha_i \le C_-, \quad y_i = -1, \end{aligned} \tag{2}$$

We can use some appropriate kernel $K(x_i, x_j)$ representing the inner product $\Phi(x_i)\cdot\Phi(x_j)$. The choice of the kernel functions depends on the experience and experiment.

For any input x, first we calculate the distance between the data point and the center of the hyper-sphere, if the following condition is true,

$$\|\Phi(x)-x\| \leq R \tag{3}$$

The data point x belongs to the hyper-sphere and regard it belongs to +1 class, otherwise it belongs to -1 class.

$$R^2 = 1 - \frac{2}{n}\sum_{k,i} a_i y_i K(x_k, x_i) + \sum_{i,j} a_i a_j y_i y_j K(x_i, x_j) \tag{4}$$

Where x_k are the bounded vectors and n is the number of the bounded vectors.

The decision function can be written as:

$$\begin{gathered} f(x) = \operatorname{sgn}(\sum_{i=1}^{l} \alpha_i y_i K(x, x_i) + b) \\ where\ b = -\frac{1}{n}\sum_{k,i} \alpha_i y_i K(x_k, x_i) \end{gathered} \tag{5}$$

4 Data and Experiment Results

4.1 Characteristics of Input Data

Ultimately, churn occurs because subscribers are dissatisfied with the price or quality of service, usually as compared to the offerings of competing carriers. The main reasons for subscriber dissatisfaction vary by region and over time. We categorize our input variables as follows [9].

- Demographics: Geographic and population data of a given region.
- Usage level. Call detail records (date, time, duration, and location of all calls), peak / off-peak minutes used, additional minutes beyond monthly prepaid limit etc.
- Quality of Service (QOS): Dropped calls (calls lost due to lack of coverage or available bandwidth), and quality of service data (interference, poor coverage).
- Features / Marketing: Details of service bundle such as email, instant messaging, paging, rate plans offered by carrier and its competitors, recent entry of competitors into market, advertising campaigns, etc.

4.2 Data

The subscriber database provided by the carrier is stored in an Oracle database. It contains three relations which are listed in Table 1.

Table 1. Relations in the subscriber database

Relation	Description
Demographics	Demographic records (Geographic and population data)
Billing	Billing records (fee, additional charges, etc)
CDR	Call detail records (date, time, duration, location, etc)

Table 2. Distribution of the data used in training and test in the simulation

Training data set		**Testing data set**	
Number of normal examples	Number of churn examples	Number of normal examples	Number of churn examples
2134	152	824	67

The simulation data bases are summarized in Table 2. We select 2958 examples (2134 examples for training data set and 824 examples for testing data set) from the data consisted of 100,000 customers for whom there were 171 potential predictor variables. The data were compiled for a three month period and then whether or not the customers churned in the fifth month was recorded.

4.3 Experiment Results

We apply our improved one-class SVM to a set of Customer Churn data as described above. We use different Kernel function in SVM and we get different accuracy rate. The result of comparison of different Kernel function is shown in Table 3. Gaussian Kernel function shows the best performance, and the result also indicates that the separating hyper-plane is non-liner.

Table 3. Comparison of different Kernel function

SVM Kernel	Linear	Polynomial	Gaussian
Accuracy rate	72.28%	77.65%	87.15%

We perform the experiment over the abstract data, by using ANN, Decision Tree, Naïve Bays, and compare them with the improved one-class SVM (Gaussian Kernel) given in this paper. The neural networks used in our experiments are multilayer perceptrons with a single hidden layer which contains 20 nodes and they were trained by the back propagation algorithm with the learning rate was set to 0.3 and the momentum term was set to 0.7. The result is given in Table 4. The experience shows that the improved one-SVM method has the best performance in detecting the churn.

Table 4. Comparison of different Algorithm

Algorithm	ANN	Decision Tree	**Naïve** Bays	SVM (Gaussian)
Accuracy rate	78.12%	62%	83.24%	87.15%

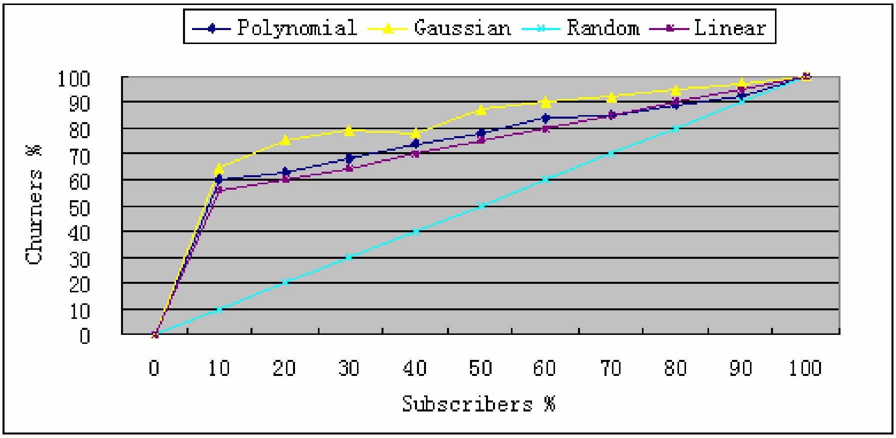


Fig. 1. Lift curve of different kernel function (Polynomial, Gaussian and Linear)

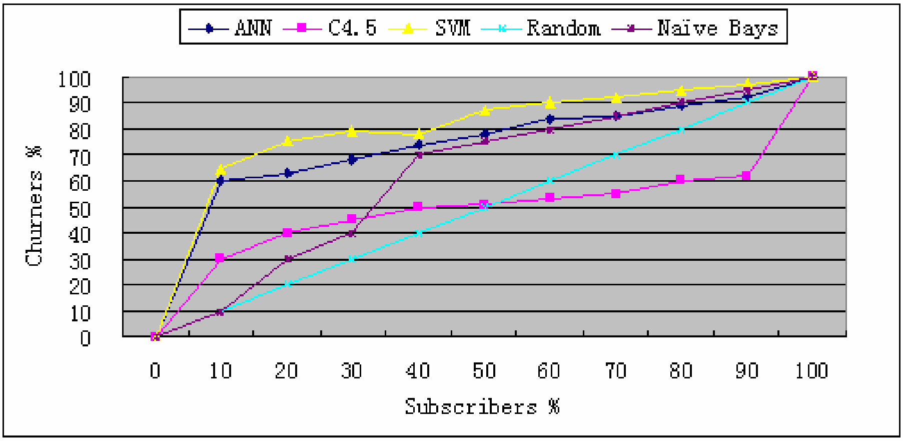


Fig. 2. Lift curve of different algorithm: ANN, C4.5, Naïve Bays and SVM

In the telecommunications industry, the "churn" and "no churn" prediction is usually expressed as a lift curve. The lift curve plots the fraction of all churners having churn probability above the threshold against the fraction of all subscribers having churn probability above the threshold. The lift curve indicates the fraction of all churners can be caught if a certain fraction of all subscribers were contacted.

Fig 1 shows the lift curves of different kernel function of improved one-class SVM. SVM with Gaussian kernel function can detect more churners than which with Polynomial and Linear kernel. Fig 2 shows the lift curves of different algorithm.

5 Conclusion and Future Work

In this paper, we introduce Customer Churn Prediction and use an improved one-class SVM method to wireless industry data set. The performance of different kernel functions in the improved one-class SVM has been investigated, and the result shows that RBF kennel function get highest accuracy. The classification accuracy of SVM, 87.15%, is better than of ANN, Decision Tree, and Naïve Bays. Support vector machines hold high potential against traditional approaches due to their scalability, faster training and running times. Application of support vector machines to the task of customer churn prediction shows promising results and this work is a contribution to the researches done in the field.

Some more research should be done in how to choose appropriate kernel parameters and input features for better accuracy.

Acknowledgement

Foundation item: Project supported by the National Natural Science Foundation of China (Grant No.70202008) and the National High-Tech. R&D Program for CIMS, China (Grant No.2003AA414021).

References

1. Ding-An Chiang, Yi-Fan Wang, Shao-Lun Lee, Cheng-Jung Lin, Goal-oriented sequential pattern for network banking churn, Expert Systems with Applications 25 (2003) 293-302
2. Duke Teradata, Teradata Center for Customer Relationship Management. Retrieved on: Nov 7, (2002).
3. Bernhard scholkopf et al., Estimating the support of a High-Dimensional Distribution, Technical Report, Department of Computer Science, University of Haifa, Haifa, (2001)
4. Scott A. Neslin, Sunil Gupta Wagner Kamakura Junxiang Lu Charlotte Mason, Defection Detection: Improving Predictive Accuracy of Customer Churn Models
5. V. N. Vapnik, The Nature of Statistical Learning Theory. New York: Springer, (1995)
6. Trafalis, Theodore B. Support vector machine for regression and applications to financial forecasting, Proceedings of the International Joint Conference on Neural Networks, v 6, (2000) 348-353
7. B. Scholkopf, J. C. Platt, J. T. Shawe, A. J. Smola, R. C. Williamson, "Estimation the support of a high-dimensional Distribution", Technical Report MSR-TR-99-87, Microsoft Research
8. Kunlun Li, Houkuan Huang, Shengfeng Tian, Wei Xu, Improving one-class SVM for Anomaly detection, Proceedings of the second international conference on machine learning and cybernetics, Xi'an, 2-5 November, (2003)
9. Nath, Shyam V., Behara, Ravi S.Customer churn analysis in the wireless industry A data mining approach, Proceedings - Annual Meeting of the Decision Sciences Institute, (2003) 505-510

The Application of Adaptive Partitioned Random Search in Feature Selection Problem

Xiaoyan Liu[1], Huaiqing Wang[1], and Dongming Xu[2]

[1] Department of Information systems,
City University of Hong Kong, Hong Kong
50007212@student.cityu.edu.hk, iswang@cityu.edu.hk
[2] Business School, The University of Queensland, Australia
dxu@business.uq.edu.au

Abstract. Feature selection is one of important and frequently used techniques in data preprocessing. It can improve the efficiency and the effectiveness of data mining by reducing the dimensions of feature space and removing the irrelevant and redundant information. Feature selection can be viewed as a global optimization problem of finding a minimum set of M relevant features that describes the dataset as well as the original N attributes. In this paper, we apply the adaptive partitioned random search strategy into our feature selection algorithm. Under this search strategy, the partition structure and evaluation function is proposed for feature selection problem. This algorithm ensures the global optimal solution in theory and avoids complete randomness in search direction. The good property of our algorithm is shown through the theoretical analysis.

1 Introduction

Data mining is the process of finding patterns and relations in large data base [10]. When data becomes increasingly larger in both numbers of features and instances, a severe challenge is posed in terms of efficiency and effectiveness due to higher dimensions of the feature space or irrelevant information. Feature selection is one of important and frequently used techniques in data preprocessing to solve this kind of problems [3, 14].

Feature selection problem can be defined as finding m relevant features among the original n attributes, where $1 \leq m \leq n$ to define the data in order to minimize the error probability or some other reasonable selection criteria. It is proved to be NP-hard in nature [4]. Many methods have been proposed. In general, they can be classified to two categories (1) the filter approach [6, 9, 16, 22], i.e., the feature selector is independent of a learning algorithm and serves as a filter to sieve the irrelevant and redundant features, (2) and the wrapper approach [5, 8, 11, 12], i.e., the feature selector is like a wrapper around a learning algorithm relying on which the relevant features are determined. In this paper, we are focused in the filter approach.

Researchers have studied various aspects of feature selection algorithms [15]. Major aspects of feature selection [13] include feature subset generation, search strategies, goodness evaluation, stopping criteria, etc. Feature subset generation studies

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 307 – 314, 2005.

how a subset is generated following search directions. Search strategies cover exhaustive and complete search, random search, and heuristic search, etc [7, 16, 19]. The goodness of a feature subset can be evaluated using various measures: consistency, distance, information, dependency, accuracy, etc [1, 2, 9, 15]. The stopping criterion determines when the feature selection process should stop, such as minimum number of features or maximum number of iterations, the increment of improvement.

In this paper, we applied adaptive partitioned random search [20, 21] into our feature selection algorithm. In the previous research [18], exhaustive search is the optimal search in the sense that the best solution guaranteed. But its cost is $O(2^n)$. Sequential research is done in an iterative manner and once the state is selected it is not possible to go back. Its cost is $O(n^{k+1})$, where *k* is the number of evaluated subsets in each state change. However, these methods do not guarantee an optimal result since the optimal solution could be in a region of the search space that is not visited. Random search use its randomness to avoid to stay on a local minimum and to allow temporarily moving to other state with worse solutions. Since the subset generation is completely random, it is also called pure random search, PRS for short. Its low efficiency is caused by their passive character since they do not use the previously obtained information. Unlike them, adaptive partitioned random search partitions the feasible solution space to several regions and concentrates the computational effort in the most promising region which is determined by random sampling in these regions.

The rest of the paper is organized as follows. In Section 2, a brief introduction of adaptive partitioned random search is presented. Based on the idea of adaptive partitioned random search, the partition structure and promising index are proposed in Section 3, and then our adaptive partitioned random search algorithm for feature selection problem is given, which is abbreviated to APRSF. In section 4, we compare our algorithm with PRS algorithm theoretically. The last section concludes the paper with a brief discussion and future extension of this study.

2 Adaptive Partitioned Random Search

Unlike the pure random search, the adaptive partitioned random search strategy always searches in the most promising region. The basic idea is as follows. Let the search region *Q* be partitioned into certain number of sub-regions. If the sampled information of each sub-region can be utilized to determine which sub-region is more “promising,” we can avoid spending too many function evaluations in those unpromising sub-regions. Specifically, the problem is formulated as follows.

First of all, a partition structure on search region should be constructed. Suppose partition the search region r into $s \ge 2$ nonempty sub-regions, r_i, such that $r_i \ne \varnothing$, $r_i \bigcap r_j = \varnothing$, $1 \le i \le s$. An independent and identically distributed (i.i.d.) random sampling scheme is employed for each sub-region. Then the observed function value in sub-region r_i can be considered as a random variable. Through the sampling, the promising index is obtained for each sub-region. The one with the best promising index is taken as the current most promising region. It is further partitioned into a certain number of smaller sub-regions. The remaining sub-regions are annexed into a single region, named as surrounding region. Then an i.i.d. random sampling scheme is employed for the new surrounding region and sub-regions of the current most promising region.

3 Adaptive Partitioned Random Search Method for Feature Selection Problem

3.1 Partition Structure

Suppose a feature selection problem with n features $f_1, f_2, \cdots, f_n$, the set of all the feasible solutions is composed of all the possible subsets of the full feature set $F = \{f_1, f_2, \cdots, f_n\}$. We denote it by Q, then

$$Q = \{\boldsymbol{v} \mid \boldsymbol{v} \in \{0,1\}^n\},$$

where $\boldsymbol{v}$ is an n dimensional 0-1 vector. The 0-1 value appearing in the ith position of $\boldsymbol{v}$ indicates whether the subset contains the feature f_i or not. It's easy to see the cardinality of Q, denoting by $|Q|$, is $\sum_{i=0}^{n} C_n^i = 2^n$. (For convenience, $(0,0,\cdots,0)$ is considered as a feasible solution.) To partition Q, we randomly select k features. There are 2^k possible values for those k features. Fixed the values of these features, the remained features can take 0 or 1 arbitrarily. Thus Q is partitioned to 2^k regions each of which contains 2^{n-k} feature subsets. Then constantly randomly select k features from the remained features to partition each sub-region until all the features are used. This kind of partition process is called k-partition scheme. The detail of the 1-partition process when n=3 is illustrated in Fig. 1. The nodes of 1-partition tree denotes the different partitioned regions. The leaf node can be taken either a feasible solution or a region containing only one element. The root node symbols the initial region r_0 composed of all the subsets. Each middle node is a region composed of the subsets in the leaf nodes belonging to it.

3.2 Promising Index

The promising index guides the search direction in the given partition structure. At fact, it is the evaluation function. How promising one region should be determined by the samples in this region. We use the best sampled subset to indicate it. Consequently, it is reduced to how to evaluate the goodness of a feature subset. In this paper, we propose an evaluation function similar to consistency measure.

Inconsistence rate is a kind of often used consistency measure. It is calculated as follows. Two instances are considered inconsistent if they match except for their class labels. For all the matching instances (without considering their class labels), the inconsistency count is the number of the instances minus the largest number of instances of class labels. The inconsistency rate is the sum of all the inconsistency counts divided by the total number of instances.

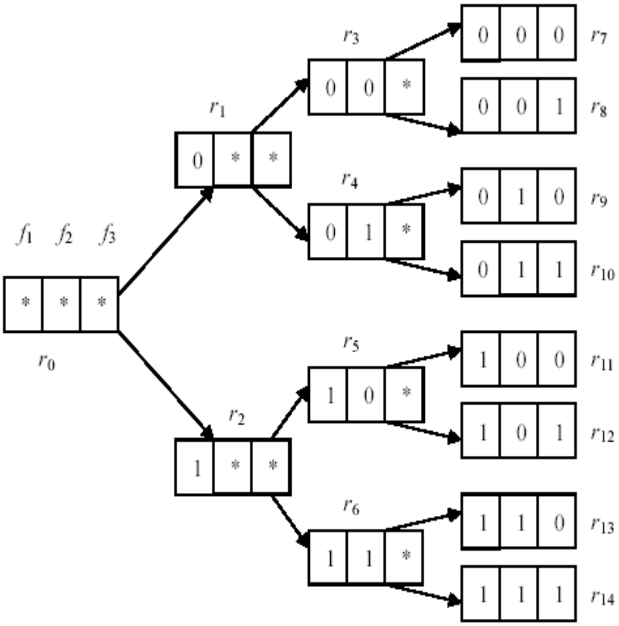


Fig. 1. 1-partition scheme when *n*=3

Suppose there are *s* classes in the classification problem and each class is denoted by $c_i, i = 1,2,\cdots,s$. $P(c_i)$ is the priori probability for class *i* in the original data set. Among the *n* matching instances in a given feature subset *G*, denote $n_i(i = 1,2,\cdots,s)$ as the number of instances belonging to c_i. We assume that all the *n* matching instances belong to the class with the largest n_i. Then a new class probability for all the instances, $P_G(c_i)$, $i = 1,2,\cdots,s$, is obtained. Here the Chi-Square test is used to examine the difference between $P_G(c_i)$ and $P(c_i)$, whose degree of freedom is *s*-1. If the two class probabilities have no significant difference under the Chi-Square hypothesis test, the subset is good enough to substitute the full feature set *F*.

The promising index of the region not only relies on the Chi-Square statistics of sampled subsets but also depends on the number of features in the subset. For any sampled subset *G*, the evaluation function is defined as follows,

$$h(G) = \begin{cases} 1, & \chi_G^2 > C \\ \dfrac{|n|}{|G|}, & otherwise \end{cases} \tag{1}$$

where *C* is the critical value to determine whether the subset is similar to the full feature set *F*. If there exists two subsets such that $h(G_1) = h(G_2)$, then the one with the smaller Chi-Square value is preferred. Let G^* be the best among all *N* sampled subsets in a region *r*, the promising index of this region, $I(r) = h(G^*)$.

3.3 Adaptive Partitioned Random Search Algorithm for Feature Selection Problem

Under the frame of the adaptive portioned random search strategy, we give the algorithm as follows.

Algorithm 1. Adaptive Partitioned Random Search Algorithm for Feature Selection Problem (APRSF)

```
Input: $F = \{f_1, f_2, \cdots, f_n\}$
Output: A subset of $F$
Step 1: Initial $r_0 = r = Q$, $s = \varnothing$, $G_{best} = F$, $k$=0, $C$, $K$ are specified
Step 2: Partition $r_0$,
Step 3: Find the most promising index:
        $\sigma = \{\eta \mid \eta \in \text{children}(r)\mid\}$, $m = \mid\sigma\mid$, $\sigma_{m+1} = s$, Index=0;
        For $i$=1: $m+1$
            Randomly sample $N$ subsets in $\sigma_i$, find the best $G_i^*$
        End;
        Index= $\arg\max_{1\le i\le m+1}(I(\sigma_i))$,
        If $G_{index}^* \ge G_{best}$ then $G_{best} = G_{index}^*$
        If $\mid\sigma_{index}\mid = 1$ then goto Step 5
Step 4: If $1 \le \text{index} \le m$ then $r = \sigma_{index}$, $s = r_0 \setminus r$ else $r = \sup(r)$; $k = k + 1$;
        If $k > K$ then goto Step5 else goto Step3
Step 5: Output the $G_{best}$
```

In the algorithm, Children(*r*) and Sup(*r*) respectively symbol the set of children nodes and the parent node of *r*. *C* and *K* are the predefined critical value and maximal transition steps, respectively. *C* is determined by the confidence level α. Step 1 initials the most promising region, surrounding region and the optimal solution, critical value, the maximal transition steps. In Step 2, the partition structure of the whole search space is generated. Step 3 is to find the most promising index and record the current best solution in G_{best}. In Step 4, if one sub-region of the current most promising region is most promising, it becomes the new current most promising region, and all the remained regions are annexed to a new surrounding region. Otherwise, if the index indicates the surrounding region is most promising, return to the parent node of the current most promising region to resample. The algorithm is stopped when the predefined maximal transition step is exceeded or the current most promising region is a leaf node. At last, the recorded G_{best} is output as the final solution.

4 Algorithm Analyses

In this section, we analyze the behavior of APRSF. First, the output of our algorithm is one subset $G \subseteq F$ with smaller $|G|$, such that $\chi_G^2 < C$. The search process for optimal subset is realized by transition between different regions. It can be written as a sequence of regions $\{r_i\}\Big|_{i=0}^{\infty}$, which apparently is a Markov Chain because the next transition is only related with the current state. Its state space is the set of all partitioned regions. According to [20], the adaptive partitioned random search converges almost surely to a global optimal solution in finite time. That means if the transition steps is long enough, our algorithm always gets the optimal subset.

Let $F(y) = \Pr(h(G) \le y)$ and $F_i(y), 1 \le i \le m+1,$ be the probability distribution function of $h(G)$ induced from the independent and uniformly distributed sampling scheme applied to entire region *Q* and its sub-region *i*, we have the following equation [21],

$$F(y) = \frac{1}{m+1}\sum_{i=1}^{m+1} F_i(y) \tag{1}$$

The PRS generates the next subset from the whole region randomly, equal to the case when *m*+1=1 in APRS. Suppose *m*+1 subsets be sampled by APRS and PRS. $h(G_1)$ and $h(G_2)$ are the corresponding evaluation values of the sampled optimal subsets respectively. Though the probabilities that the optimal subset is chosen by both methods are $(m+1)/2^n$, the expectations of $h(G_1)$ and $h(G_2)$ are different.

Because according to [21], for pure random search,

$$\Pr(h(G_2) \le y \mid \text{PRS}) = (F(y))^{m+1} = \left(\frac{1}{m+1}\sum_{i=1}^{m+1} F_i(y)\right)^{m+1}.$$

For adaptive partitioned random search,

$$\Pr(h(G_1) \le y \mid \text{APRS}) = \prod_{i=1}^{m+1} F_i(y)\cdot$$

Since arithmetic average exceeds geometric average, it follows that

$$\Pr(h(G_2) > y \mid \text{PRS}) \le \Pr(h(G_1) > y \mid \text{APRS}),$$

then

$$E(h(G_1)) \ge E(h(G_2)) \tag{2}$$

This is the case when *N*=1. We discuss the case when $N \ne 1$ below.

In each iteration of our algorithm, we find the most promising region by sampling *N* subsets in each of *m*+1 regions. It is equivalent to the process that each round we sample *m*+1 subsets from the *m*+1 regions, namely one subset in one region, then get the optimal subset among them. The sampling process is operated *N* rounds, at last compare among *N* optimal subsets. If the optimal value in each round is denoted by $h(G_{1i}^*)$, and the optimal value among $(m+1)*N$ sampled subsets in pure random search is $h(G_2^*)$. It is easy to see that $h(G_{1i}^*), i = 1,\cdots,N$ are independent and have the identical distribution (i.i.d.). Thus, we have

$$\Pr\left(\max_{i=1,\cdots,N}(h(G_{1i}^*) \le y)\right) \mid \text{APRS}) = \left(\prod_{i=1}^{m+1} F_i(y)\right)^N \cdot$$

$$\Pr(h(G_2^*) \le y \mid \text{PRS}) = (F(y))^{N(m+1)} = \left(\frac{1}{m+1}\sum_{i=1}^{m+1} F_i(y)\right)^{N(m+1)}$$

$$= \left[\left(\frac{1}{m+1}\sum_{i=1}^{m+1} F_i(y)\right)^{(m+1)}\right]^N$$

In a similar way, the following inequation holds,

$$E(\max_{i=1,\cdots,N}(h(G_{1i}^*)) \ge E(h(G_2^*)) \tag{3}$$

It means in the same sampling times, our adaptive partitioned random search algorithm is better than pure random search in the sense of expectation.

5 Conclusions

In this paper, we apply the adaptive partitioned random search strategy in our feature selection algorithm. Under the adaptive partitioned random search strategy, we defined the partition structure and promising index. To test whether subset is approximate to the full feature set, we propose a new criterion function based on Chi-Square test. Though we analyzed the behavior of our algorithm theoretically, our algorithm indeed has its own advantage. On the one hand, compared with sequential search which doesn't guarantee the optimal solution, our algorithm has the global convergence in theory. On the other hand, compared with the pure random search, under the same sampling times, our result always is better than pure random search in the sense of expectation.

This work still has some opened topics worth to deep study. In our future work, we will conduct numeric experiments to test our algorithm's practical efficiency and accuracy. Since the partition structure also influences the algorithm's efficiency, another future study will be on how to reasonably partition the whole solution space to improve the algorithm.

References

1. Almuallim, H., Dietterich, T. G.: Learning with Many Irrelevant Features, in Proceedings of the Ninth National Conference on Artificial Intelligence (1992) 547-552
2. Ben-Bassat, M.: Pattern Recognition and Reduction of Dimensionality, in Krishnaiah, P. R., Kanal, L. N. (eds.): Handbook of Statistics-II, North Holland (1982) 773-791
3. Blum, A. L., Langley, P.: Selection of relevant features and example in machine learning, vol. 97, Artificial Intelligence (1997) 245-271

4. Blum, A. L., Rivest, R. L.: Training a 3-node Neural Networks in NP-complete, vol. 5, Neural Networks (1992) 117-127
5. Caruana, R., Freitag, D.: Greedy Attribute Selection, in Proceedings of the Eleventh International Conference on Machine Learning (2002) 153-172
6. Dash, M., Choi, K., Scheuermannm, P., Liu, H: Feature Selection for Clustering-a Filter Solution, in Proceedings of the 2nd International Conference on Data Mining (2002)115-122
7. Doak, J.: An Evaluation of Feature Selection Methods and Their Application to Computer Security,Technical report, University of California, Department of Computer Science (1992)
8. Dy, J. G., Brodley, C. E.: Feature Subset Selection and Order Identification for Unsupervised Learning, in Proceedings of the Seventeenth International Conference on Machine Learning (2000) 247-254
9. Hall, M.A.: Correlation-based Feature Selection for Discrete and Numeric Class Machine Learning, in Proceedings of the Seventeenth International Conference on Machine learning, (2000) 359-366
10. Kerber, R., Livezy, B., Simoudis, E.: A hybrid System for Data Mining, in Goonatilake, S., Khebbal, S. (eds.), Intelligent Hybrid Systems, John Wiley (1995)
11. Kim, Y., Street, W., Menczer, F.: Feature Selection for Unsupervised Learning via Evolutionary Search, in Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2000) 365-369
12. Kohavi, R., John, G. H.: Wrappers for Feature Subset Selection, vol. 97, no.1-2, Artificial Intelligence (1997) 273-324
13. Langley, P.: Selection of Relevant Features in Machine Learning, in Proceedings of the AAAI Fall Symposium on Relevance, AAAI Press (1994) 140-144
14. Liu, H., Motoda, H.: Feature Extraction, Construction and Selection: a Data Mining Perspective, Boston: Kluwer academic publishers (1998)
15. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining, Kluwer academic publishers, Boston (1998)
16. Liu, H., Setiono, R.: A Probabilistic Approach to Feature Selection- a Filter Solution, in Proceedings of the 13th International Conference on Machine Learning, (1996) 319-327
17. Liu, H., Yu, L.: Toward Integrating Future Selection Algorithms for Classification and Clustering, vol. 17, no. 3, IEEE Trans. on Knowledge and Data Engineering (2005) 1-12
18. Molina, L.C., Belanche, L., Nebot, A.,: Feature Selection Algorithms: A Survey and Experimental Evaluation, in Proceedings of IEEE International Conference on Data Mining (2002) 306-313
19. Narendra, P.M., Fukunaga, K.: A Branch and Bound Algorithm for Feature Selection Subset Selection, vol. C-26, no. 9, IEEE Trans. On Computing (1977) 917-922
20. Shi, L.Y., Olasfsson, S.: Nested Partitions Method for Global Optimization, vol. 48, no. 3, Operation research (2000) 390-407
21. Tang, Z.B.: Adaptive Partitioned Random Search to Global Optimization, vol. 39, no. 11, IEEE Trans. On Automatic Control (1994) 2235-2244
22. Yu, L., Liu, H.: Feature Selection for High Dimensional Data: a Fast Correlation-based Filter Solution, in Proceedings of the Twentieth International Conference on Machine Learning (2003) 856-863

Heuristic Scheduling of Concurrent Data Mining Queries*

Marek Wojciechowski and Maciej Zakrzewicz

Poznan University of Technology,
Institute of Computing Science,
ul. Piotrowo 3a, 60-965 Poznan, Poland
{marek, mzakrz}@cs.put.poznan.pl

Abstract. Execution cost of batched data mining queries can be reduced by integrating their I/O steps. Due to memory limitations, not all data mining queries in a batch can be executed together. In this paper we introduce a heuristic algorithm called CCFull, which suboptimally schedules the data mining queries into a number of execution phases. The algorithm significantly outperforms the optimal approach while providing a very good accuracy.

1 Introduction

Multiple Query Optimization (MQO) [8] is a database research area that focuses on optimizing sets of queries together by executing their common expressions only once in order to save query execution time. Many exhaustive and heuristic algorithms have been proposed for traditional MQO [3][6][7]. A specific type of a database query is a Data Mining Query (DMQ) [5], which describes a data mining task. It defines constraints on the data to be mined and constraints on the patterns to be discovered. DMQs are submitted for execution to a Knowledge Discovery Management System KDDMS [5], which is a Database Management System (DBMS) extended with data mining functionality. Traditional KDDMSs execute DMQs serially and do not try to share any common expressions between different DMQs.

DMQs are often processed in batches of dozens of queries, executed during low user activity time. Queries in a batch may show many similarities to each other, e.g., their source data sets may overlap. If such queries were executed serially, then it would be likely that many I/O operations were wasted because the same database blocks were required by multiple DMQs. If I/O steps of different DMQs were integrated and performed once, then it would be possible to decrease the overall execution cost and time of the whole batch. One of the methods to process batches of DMQs is Apriori Common Counting (ACC) [9], focused on frequent itemset discovery queries [1]. ACC is based on Apriori algorithm [2] and it integrates the steps of candidate support counting – all candidate hash trees for multiple DMQs are

* This work was partially supported by the grant no. 4T11C01923 from the State Committee for Scientific Research (KBN), Poland.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 315–322, 2005.

loaded into memory and the database is scanned only once. Basic ACC [9] assumes that all DMQs fit in memory, which is not a common case, at least for initial Apriori iterations. If the memory can hold only a subset of all DMQs, then it is necessary to divide/schedule the DMQs into subsets called phases [10]. The way such scheduling is done determines the overall cost of batched DMQs execution. To solve the scheduling problem, in [10] we proposed an "initial" heuristic algorithm, called *CCRecursive*. According to our experiments *CCRecursive* offers acceptable accuracy and on average outperforms the optimal scheduling algorithm. However, in particular situations its execution time could increase significantly due to its recursive nature, which is the motivation for seeking novel, more predictable solutions. In this paper we present and evaluate another heuristic algorithm for scheduling data mining queries to be executed by ACC, called *CCFull.*

1.1 Related Work

Multiple-query optimization has been extensively studied in the context of database systems (see e.g. [8]), however very little work has been done on optimizing sets of data mining queries. To the best of our knowledge, apart from the ACC method discussed in this paper, the only other multiple query processing scheme for data mining queries is Mine Merge, presented in one of our previous papers [11]. In contrast to ACC, Mine Merge is independent of a particular frequent itemset mining algorithm. However, it was proven very sensitive to data distribution and less predictable than ACC.

2 Preliminaries and Problem Statement

Data mining query. A *data mining query* is a tuple $DMQ = (\mathcal{R}, a, \Sigma, \Phi, \beta)$, where $\mathcal{R}$ is a relation, a is an attribute of $\mathcal{R}$, Σ is a condition involving the attributes of the relation $\mathcal{R}$, Φ is a condition involving discovered patterns, and β is the minimum support threshold. The result of the data mining query is a set of patterns discovered in $\pi_a \sigma_\Sigma \mathcal{R}$ and satisfying Φ.

Problem statement. Given is a set of data mining queries $DMQ = \{dmq_1, dmq_2, ..., dmq_n\}$, where $dmq_i = (\mathcal{R}, a, \Sigma_i, \Phi_i, \beta_i)$, Σ_i has the form "$(l^i{}_{1min} < a < l^i{}_{1max}) \vee (l^i{}_{2min} < a < l^i{}_{2max}) \vee .. \vee (l^i{}_{kmin} < a < l^i{}_{kmax})$", $l^i{}_* \in dom(a)$ and there exist at least two data mining queries $dmq_i = (\mathcal{R}, a, \Sigma_i, \Phi_i, \beta_i)$ and $dmq_j = (\mathcal{R}, a, \Sigma_j, \Phi_j, \beta_j)$ such that $\sigma_{\Sigma i}\mathcal{R} \cap \cap \sigma_{\Sigma j}\mathcal{R} \neq \emptyset$. The problem of *multiple query optimization* of *DMQ* consists in generating such an algorithm to execute *DMQ* that has the lowest I/O cost.

Apriori Common Counting (ACC). If the set of data mining queries was executed serially, i.e. one data mining query at a time, then the total execution cost would be the sum of execution costs of data selection formulas for each data mining query separately. ACC executes a set of data mining queries by integrating their I/O operations. It is based on the traditional Apriori approach to discover frequent itemsets. In the first step, for each data mining query we build a separate hash tree for

1-candidates. Next, for each distinct data selection formula we scan its corresponding database partition and we count candidates for all the queries that contain the formula. Such a step is performed for 2-candidates, 3-candidates, etc. Notice that if a given distinct data selection formula is shared by many queries, then its corresponding database partition is read only once. An overview of ACC is shown in Fig. 1.

```
for ($i$=1; $i$<=$n$; $i$++)                    /* $n$ = number of data mining queries */
  $\mathcal{C}_1^i$ = {all 1-itemsets from $\sigma_{s1 \cup s2 \cup .. \cup sk}\mathcal{R}$, $\forall s_j \in S$: $(dmq_i, s_j) \in E$}  /* generate 1-candidates */
for ($k$=1; $\mathcal{C}_k^1 \cup \mathcal{C}_k^2 \cup .. \cup \mathcal{C}_k^n \neq \emptyset$; $k$++) do begin
   for each $s_j \in S$ do begin
      $\mathcal{CC} = \bigcup \mathcal{C}_k^l$: $(dmq_l, s_j) \in E$;          /* select the candidates to count now */
      if $\mathcal{CC} \neq \emptyset$ then count($\mathcal{CC}$, $\sigma_{sj}\mathcal{R}$);
   end
   for ($i$=1; $i$<=$n$; $i$++) do begin
      $\mathcal{F}_k^i$ = {$C \in \mathcal{C}_k^i$ | $C.count \geq minsup^i$};      /* identify frequent itemsets */
      $\mathcal{C}_{k+1}^i$ = generate_candidates($\mathcal{F}_k^i$);
  end
end
for ($i$=1; $i$<=$n$; $i$++) do
   $Answer^i = \bigcup_k \mathcal{F}_k^i$;                  /* generate responses */
```

Fig. 1. Apriori Common Counting

3 Heuristic Scheduling of Concurrent Data Mining Queries

The basic ACC assumes unlimited memory and therefore the candidate hash trees for all DMQs can completely fit in memory. If, however, the memory is limited, then ACC execution must be divided into multiple *phases*, so that in each phase only a subset of DMQs is processed. In such a case, the key question to answer is: which data mining queries from the set should be executed together in one phase and which data mining queries can be executed in different phases? We refer to the task of data mining queries partitioning as to *data mining query scheduling*.

The problem of data mining query scheduling is a combinatorial problem which can be solved by generating all possible schedules and then choosing the best one. Such approach can be used for a small number of data mining queries, however, for a realistic case it is infeasible. The number of all possible schedules is determined by the *Bell number*, e.g., for 13 queries we get over 4 million schedules. Therefore, we propose a heuristic algorithm *CCFull*, which quickly finds a suboptimal schedule.

3.1 Algorithm CCFull

In the first step we generate a *gain graph* for the set of data mining queries. The gain graph is a full hypergraph, in which vertices represent the data mining queries while edges are described with weights which represent the amount of I/O cost reduction to be achieved if data mining queries connected with the edge were executed together (in the same phase). If common execution of given data mining queries results in no reduction of I/O cost, the weight of the connecting edge is zero. A sample gain graph is shown in Fig. 2. For example, it can be noticed that common execution of the data mining queries *dmq*$_0$, *dmq*$_2$, and *dmq*$_3$ would reduce the total I/O cost by 16 units (the weight of the connecting hyperedge) compared with the sequential execution, since for *dmq*$_0$ and *dmq*$_2$ the cost of redundant I/O operations is 5 units, for *dmq*$_2$ and *dmq*$_3$ the cost of redundant I/O operations is 8 units, and for *dmq*$_0$ and *dmq*$_3$ the cost of redundant I/O operations is 3 units. Using the same example, it can be also noticed, that common execution of only the data mining queries *dmq*$_1$ and *dmq*$_2$ provides no cost reduction (the weight of the connecting hyperedge is zero).

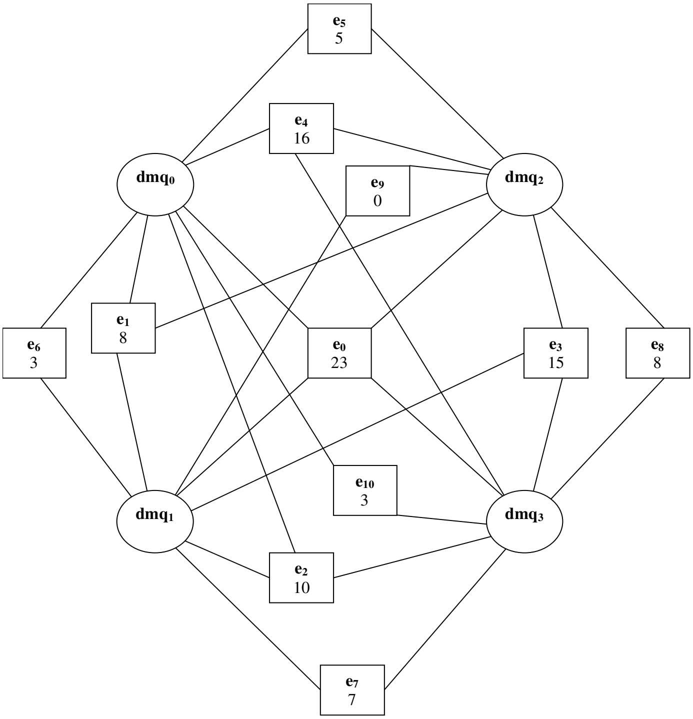


Fig. 2. Sample gain graph

The gain graph can be generated using the algorithm *GenerateGainGraph* shown in Fig. 3. The algorithm takes two arguments: the set of all distinct data selection formulas and the set of all data mining queries. First, the algorithm builds a full hypergraph whose nodes are the data mining queries (line 1). Each hyperedge receives the weight of zero, initially (line 3). Then, for each hyperedge *e*, we create a set *P* of distinct data selection formulas involved in all data mining queries connected with the hyperedge *e* (line 4). I/O costs for executing the distinct data selection formulas from *P* are then summarized and the result is assigned to the hyperedge *e* weight (line 5 and 6).

```
GenerateGainGraph(S, DMQ):
begin
  generate a full hypergraph $G=\{V,E\}$, $V=DMQ$
  for each $e \in E$ do begin
    $e.gain = 0;$
    $P = \{s_i \in S \mid \exists dmq_j \in e, dmq_j = (\mathcal{R}, a, \Sigma_j, \Phi_j, \beta_j), s_i \subseteq \Sigma_j \}$
    for each $s \in P$ do begin
      $e.gain \mathrel{+}= cost(s) * (|\{ dmq_j: dmq_j \in e, dmq_j = (\mathcal{R}, a, \Sigma_j, \Phi_j, \beta_j), s_i \subseteq \Sigma_j \}| - 1)$
    end
  end
  return G
end
```

Fig. 3. Gain graph generation algorithm

After having created the gain graph, *CCFull* performs the following steps. All hyperedges are sorted in descending order according to their weights. Next, *CCFull* iterates over the hyperedges and checks if data mining queries connected with the current hyperedge have been already scheduled. If none of the data mining queries has been scheduled so far, and if their hash trees fit in memory, then a new phase is generated and the data mining queries are assigned to it. Otherwise, if only some of the data mining queries have been already scheduled to different phases, then *CCFull* tries to combine all those phases together with the unscheduled data mining queries. If such combined phase does not fit in memory, then the current hyperedge is ignored and *CCFull* continues with the next one. The algorithm ends when all hyperedges are processed. The algorithm *CCFull* is shown in Fig. 4.

The detailed steps of the algorithm from Fig. 4 are the following. In line (1) we initialize the set of scheduled phases – we start with the empty set. In line (2) we sort the list *E* of hyperedges from the gain graph. Hyperedges with weights equal to zero are removed from the list. In line (3) a loop starts, which iterates over the list of hyperedges. In line (4) we select all data mining queries which are connected with the current hyperedge (*tmpV*). In line (5) we test if any of the selected data mining queries belongs to any of the phases scheduled so far. If not, then in line (7) we create a new candidate phase containing all the data mining queries from *tmpV*. Otherwise, in line (9) we create a new candidate phase containing both all the data mining queries from *tmpV* and data mining queries from earlier scheduled phases, to which any of the *tmpV* data mining queries was also scheduled. In line (10) we check if hash trees of all the data mining queries from the new candidate phase fit in memory (*MEMSIZE* is the available memory size). If this condition is satisfied, then in lines (11) and (12) we append the new candidate phase to the current set of scheduled phases *Phases*, possibly replacing some of the existing phases (when multiple phases are combined). In line (13), for each data mining query which has not been scheduled we create a new phase. In step (14) we return the generated phases.

```
CCFull(G=(V,E)):
begin
  $Phases \leftarrow \{\varnothing\}$
  sort $E = <e_i, e_2, ..., e_k>$ in desc. order w.r.t. to $e_i$.gain, ignore edges with zero gains
  for each $e_i$ in $E$ do begin
    $tmpV \leftarrow \{v \in V \mid v \in e_i\}$
    if $(|\{p \in Phases \mid p \cap tmpV \neq \varnothing\}| = 0)$ then
      $commonPhases \leftarrow \varnothing$
      $newPhase \leftarrow tmpV$
    else
      $commonPhases \leftarrow \{p \in Phases \mid p \cap tmpV \neq \varnothing\}$
      $newPhase \leftarrow tmpV \cup \bigcup p \mid p \in commonPhases$
    end if
    if $(treesize(newPhase) \leq MEMSIZE)$ then
      $Phases \leftarrow Phases - commonPhases$
      $Phases \leftarrow Phases \cup newPhase$
    end if
  end
  add phase for each unscheduled query
  return Phases
end
```

Fig. 4. *CCFull* algorithm

3.2 Example

Consider scheduling of data mining queries from Fig. 2. For the sake of simplicity, assume that hash tree sizes are 10MB for each data mining query and the available memory is 20MB.

Hyperedges of the gain graph are sorted according to their weights (skipping zero-weighted hyperedges): $<e_0, e_4, e_3, e_2, e_1, e_8, e_7, e_5, e_6, e_{10}>$. In the first iteration we select the hyperedge e_0, which is connecting the data mining queries dmq_0, dmq_1, dmq_2 and dmq_3. None of the data mining queries has been scheduled so far, and total size of their hash trees is 40MB, exceeding the available memory. Therefore, the algorithm ignores the hyperedge and starts another iteration.

In the second iteration we select the hyperedge e_4, which is connecting the data mining queries dmq_0, dmq_2 and dmq_3. None of the data mining queries has been scheduled so far, and total size of their hash trees is 30MB, exceeding the available memory again. Therefore, the algorithm ignores the hyperedge and starts another iteration. In a similar way the iterations over the hyperedges e_3, e_2 and e_1 are performed – total sizes of hash trees exceed the available memory.

Yet in the sixth iteration the algorithm will behave in a different way. We select the hyperedge e_8, which is connecting the data mining queries dmq_2 and dmq_3. The total size of their hash trees is 20MB, so a new phase is created: $\{dmq_2, dmq_3\}$. In the next iteration we select the hyperedge e_7, which is connecting the data mining queries *dmq*$_1$ and *dmq*$_3$. Since *dmq*$_3$ already belongs to a scheduled phase, we try to replace the existing phase {*dmq*$_2$, *dmq*$_3$} with a new one: {*dmq*$_1$, *dmq*$_2$, *dmq*$_3$}. We are unsuccessful because the total size of hash trees for the data mining queries is 30MB, what exceeds the available memory. In the next iteration we select the hyperedge e_5, and again we are unsuccessful when trying to replace the existing phase {*dmq*$_2$, *dmq*$_3$} with a new phase {*dmq*$_0$, *dmq*$_2$, *dmq*$_3$}. The next iteration operates on the hyperedge e_6, which is connecting the data mining queries *dmq*$_0$ and *dmq*$_1$. These data mining queries do not belong to any of the existing phases and total size of their hash trees is 20MB. Therefore, a new phase is created: {*dmq*$_0$, *dmq*$_1$}.

In the last iteration we select the hyperedge e_{10}, which is connecting the data mining queries *dmq*$_0$ and *dmq*$_3$. Since both data mining queries have already been scheduled to some phases, the algorithm tries to combine the existing phases {*dmq*$_2$, *dmq*$_3$} and {*dmq*$_0$, *dmq*$_1$}. However, the phases are not merged since the total size of hash trees of their data mining queries is 40MB and exceeds the available memory. The algorithm has completed. The constructed scheduling of the four data mining queries consists of 2 phases: {*dmq*$_2$, *dmq*$_3$} and {*dmq*$_0$, *dmq*$_1$}.

4 Experimental Evaluation

In order to evaluate performance and accuracy of the *CCFull* scheduling algorithm we performed several experiments using the MSWeb dataset from the UCI KDD Archive [4]. The experiments were conducted on a PC with AMD Duron 1.2 GHz processor and 256 MB of main memory. The datasets used in all experiments resided in flat files on a local disk. Memory was intentionally restricted to 10kB-50kB. Each experiment was repeated 100 times. The queries were randomly generated.

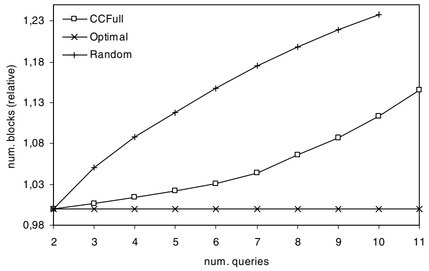


Fig. 5. Accuracy of data mining query scheduling algorithms

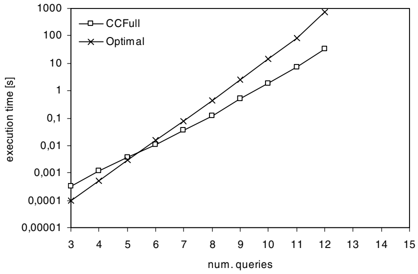


Fig. 6. Execution time of data mining query scheduling algorithms

Fig. 5 shows disk I/O costs of schedules generated by the optimal scheduling algorithm, by the *CCFull* algorithm, and by a random algorithm (which randomly builds phases from queries). For example, for the set of 10 data mining queries, the *CCFull* algorithm misses the optimal solution by only 6%.

Fig. 6 illustrates execution times for the optimal scheduling algorithm and for *CCFull*. Notice that the optimal algorithm needs ca. 1000s to schedule 12 data mining queries while *CCFull* executes in 30s.

Comparing accuracy and performance of *CCFull* with the experimental results obtained for *CCRecursive* (reported in [10]), we have to admit that on average *CCRecursive* outperforms *CCFull*, while offering a slightly better accuracy. However, in particular, rare situations *CCRecursive* took more time to complete than the optimal scheduling algorithm due to its recursive nature. We have not observed such problems with *CCFull*, which makes it a more predictable solution.

5 Conclusions and Future Work

In this paper we have introduced a new heuristic algorithm *CCFull* to schedule data mining queries for Apriori Common Counting. The algorithm offers a significant reduction of execution time over the optimal algorithm while providing a very good accuracy, and is predictable compared to our previous heuristics *CCRecursive*.

CCFull assumes that the set of data mining queries is static. However, in a real system, new queries may arrive while other queries are being executed. In the future we plan to investigate methods allowing for dynamic scheduling of the arriving queries.

References

1. Agrawal R., Imielinski T., Swami A: Mining Association Rules Between Sets of Items in Large Databases. Proc. of the 1993 ACM SIGMOD Conf. on Management of Data (1993)
2. Agrawal R., Srikant R.: Fast Algorithms for Mining Association Rules. Proc. of the 20th Int'l Conf. on Very Large Data Bases (1994)
3. Alsabbagh J.R., Raghavan V.V.: Analysis of common subexpression exploitation models in multiple-query processing. Proc. of the 10th ICDE Conference (1994)
4. Hettich S., Bay S. D.: The UCI KDD Archive [http://kdd.ics.uci.edu]. Irvine, CA: University of California, Department of Information and Computer Science (1999)
5. Imielinski T., Mannila H.: A Database Perspective on Knowledge Discovery. Communications of the ACM, Vol. 39, No. 11 (1996)
6. Jarke M.: Common subexpression isolation in multiple query optimization. Query Processing in Database Systems, Kim W., Reiner D.S. (Eds.), Springer (1985)
7. Roy P., Seshadri S., Sundarshan S., Bhobe S.: Efficient and Extensible Algorithms for Multi Query Optimization. ACM SIGMOD Intl. Conference on Management of Data (2000)
8. Sellis T.: Multiple query optimization. ACM Transactions on Database Systems, Vol. 13, No. 1 (1988)
9. Wojciechowski M., Zakrzewicz M.: Evaluation of Common Counting Method for Concurrent Data Mining Queries. Proc. of the 7th ADBIS Conference (2003)
10. Wojciechowski M., Zakrzewicz M.: Data Mining Query Scheduling for Apriori Common Counting. Proc. of the 6th Int'l Baltic Conf. on Databases and Information Systems (2004)
11. Wojciechowski M., Zakrzewicz M.: Evaluation of the Mine Merge Method for Data Mining Query Processing. Proc. of the 8th ADBIS Conference (2004)

Using Gap-Insensitive String Kernel to Detect Masquerading

Chuanhuan Yin, Shengfeng Tian, and Shaomin Mu

School of Computer and Information Technology,
Beijing Jiaotong University, Beijing, 100044, China
{chhyin, sftian}@center.njtu.edu.cn
msm@sdau.edu.cn

Abstract. Masquerade attacks may be one of the most serious attacks in computer security context. To avoid being detected, masqueraders sometimes insert some common commands such as “ls” into their command sequences intentionally for concealing their actual purpose. This causes the masquerade attacks difficult to be detected. We refer to these command sequences mixed with confusable commands as gap-insensitive. To eliminate the effects on the insertion, we present a string kernel called gap-insensitive kernel without regard to the gaps in the command sequences, and use it to detect masquerade attacks. We test it and other kernels on the dataset from keyboard commands on a UNIX platform. We find that many users’ attacks against other users can be easily detected by our gap-insensitive kernel, which means that the command sequences of these attackers are gap-insensitive. The results reveal that gap-insensitive kernel can determine gap-insensitivity in command sequences, and efface the gaps in the sequences.

1 Introduction

Computer attacks are an important security problem while the masquerade attack may be one of the most serious attacks [1]. Literally masquerading is the attempt of substituting oneself for another. It can be a very serious menace to the computer system security. Some other intrusions may be less severe than Masquerading. Let’s take Denial of Service (DoS) for example, DoS is also a kind of usual and serious attack, but its purpose is just to stop prevalent services. Different from the DoS, masquerading may obtain secure information or destroy the whole computer system after getting super privilege. Moreover, the detection of Masquerade attacks is more difficulty than that of other attacks. By mimicking the legitimate user’s behavior, the masquerade attack won’t be detected by most intrusion detection systems. In order to elude detection, some smart masqueraders insert some common or meaningless commands such as “ls” into their command sequences intentionally for concealing their actual purpose, resulting in the miss alarms of intrusion detection system.

Support vector machines, or SVMs, are learning machines that map the training vectors into high-dimensional feature space, labeling each vector by its class [2]. SVMs consider the classification problem as a quadratic optimization problem. They

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 323 – 330, 2005.

combine generalization control with a technique to avoid the “curse of dimensionality” by placing an upper bound on the margin between the different classes, making it a practical tool for large and dynamic data sets. SVMs are widely used in pattern recognition problems such as text classification and speech recognition.

For the purpose of masquerade detecting, Schoonlau et al. provided a dataset collected from keyboard commands on a UNIX platform [1]. The dataset is available at http://www.schonlau.net/. To the best of our knowledge, the dataset above mentioned is the best in masquerade detection context [3]. Therefore, we use this dataset to test our approach.

There are a number of string kernels [4], including subsequences kernels, gappy kernels, and mismatch kernels, etc. But all these kernels value will be changed when the meaningless commands are inserted in original strings, which can be seen by their definition. In this paper, we propose a string kernel called gap-insensitive kernel, and test the string kernel on the dataset above mentioned. We also present an algorithm to compute the gap-insensitive kernel. The results show that this kernel can detect many attack commands sequence with intended commands insertion in $O(|x||y|)$ time.

The rest of the paper is organized as follows. In the next section we discuss the string kernel we presented. In section 3 we describe the dataset, the experiments on it, and our results. Section 4 concludes with a discussion.

2 String Kernels

Common learning systems are designed to operate on input data after they have been converted into feature vectors. So it is necessary for these systems to extract features from the input data. However, many kinds of input data such as strings can’t be explicitly extracted to feature vectors with facility. Kernel methods [2, 4] provide an effective method which is an alternative to explicit feature extraction through the use of a kernel function. The kernel function is an inner product of two feature vectors.

To compute the feature vectors extracted from strings, a family of kernel functions is proposed, called string kernels [5]. These kernels concern about the occurrence of subsequences in string. The subsequences can be defined variously, resulting in the variety of string kernels. They can be contiguous or non-contiguous, and they can be limited to the fixed-length or not, etc.

Given a number $k \geq 1$, the spectrum kernel concerns the k-length continuous substrings shared in two strings [6]. The (k,m)-mismatch kernel is similar to spectrum kernel, but at most m mismatches are allowed [7]. In the kernel developed by Lodhi et al. [8], the gaps in the occurrence of the k-length subsequences are allowed.

In view of the command insertion of masquerade attacks, there are four criteria to string kernels proposed for detecting masquerading [9]. Firstly, string kernels are similarity measures between strings which are assessed by number of (possibly non-continuous) matching subsequences shared by two strings. Secondly, the feature map is indexed not only by k-length subsequences but all possible subsequences from alphabet Σ. Thirdly, the kernel $K(x,y)$ is stable despite the effect of the purposive insertion of some meaningless commands. Finally the kernel computation is efficient.

To follow the above criteria, we propose a new string kernel. Let ∑ be a finite set which we call the alphabet, $x\in\sum^*$ denote string defined over the alphabet ∑, |x| the length of x. The neighborhood N(x) generated by x is the set of all subsequences that x contains. Given a number $q\geq 0$, the feature map $\Phi(x)$ is defined as

$$\Phi(x) = (\varphi_s(x))_{s\in\Sigma^*}, \tag{1}$$

where

$$\varphi_s(x) = \begin{cases} \sqrt{q} & \text{s is empty string} \\ 2^{|s|-|x|} & \text{s belongs to N(x)}, \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

and q is a parameter playing the same role as σ in the RBF kernel. In the expression of $\varphi_s(x)$, we use the notation $2^{-|x|}$ to weight the kernel by the length of string s, and the square root of q to evaluate the contribution of empty string in the kernel.

The kernel K(x,y) is then defined as

$$K(x,y) = \langle \Phi(x), \Phi(y) \rangle = q + \sum_{s\in\Sigma^*:s=x(\mathbf{i})=u(\mathbf{j})} 2^{2|s|-|x|-|y|}. \tag{3}$$

In order to compute the kernel value, we present an algorithm based on dynamic programming. For strings x and y, the kernel value k(x, y) is needed to compute. We denote the length of x and y by n and m, and without lost of generality, assume $n\geq m$. Then we consider matrices $U_{i,j}$ and $DP_{i,j}$ for $i\in[0, n]$ and $j\in[0, m]$, where $U_{i,j}$ is the set of all common subsequences between strings x[1:i] and y[1:j], $DP_{i,j} = U_{i,j} - U_{i-1,j}$ is the set of common subsequences which are just appeared on *ith* line.

Figure 1 depicts the computation of common subsequences for strings 'gatta' and 'cata'.

In the process of computation, it is not necessary to store $DP_{i,j}$ for each column. Therefore, we just count the number of k-length subsequences in *ith* line, denoted by $D_{i,k}$.

DP	g	a	t	t	a
c					
a		a	a	a	a
t			t, at	t, at	t, at
a					ta, aa, ata

Fig. 1. Computations for the dynamic programming tables of the common subsequences in strings 'gatta' and 'cata'

Moreover, to speed up the computation of kernel, we also adopt a fast bit-vector algorithm [10]. M_i is used to represent the array $D_{i,k}$ for $k\in[0, m]$ because the coefficient of $D_{i,k}$ is a multiple of 2. For more details about bit-vector algorithm, paper [10] can be referenced.

According to the above analysis, we can compute the kernel K(x,y) by below procedure:

```
procedure compute_kernel
begin
  for i = 1 to n
    $E_i = 0^{2m}$;
  for i = 1 to n
    $M_i = 0^{2m}$;
  for c ∈ $\Sigma$
    Sig[c] = 1;
  for j = 1 to m
    $G = 0^{2m}$;
    for i = 1 to n
      if $x_i = y_j$
        $B = G - E_i$; $E_i = G$; $G = M_i$;
        $M_i = (B<<2) + M_i + Sig[y_j]$
        if $Sig[y_j]$ == 1
          $Sig[y_j] = 0$;
      else
        $G = G + M_i$;
  $K(x,y) = (M_1 + M_2 + \dots + M_n)\cdot 2^{2-m-n}+q$;
End.
```

In the above procedure n and m are the length of x and y respectively. Obviously this procedure gives rise to an O(mn) expected time algorithm for the computing of kernel between strings x and y.

To ensure the kernel in the interval [0, 1], the normalized kernel is defined as

$$K^s(x,y) = [K(x,x)K(y,y)]^{-1/2}K(x,y), \tag{4}$$

which we called gap-insensitive kernel. Because $\varphi_s(x)$ only takes the length of subsequences s and string x other than the gaps in x into account, we deem that the gap-insensitive kernel can eliminate the negative effect of command insertion on masquerade detecting. For example, the normalized kernel value of string “abcbc” and string “abcb” is equal to that of string “abcbc” and string “adbcb”, invalidating the insertion of command d. Therefore, this kernel can determine gap-insensitivity in command sequences, and efface the gaps in the sequences.

For comparison, we also employ the spectrum kernel to detect masquerading [6]. The spectrum kernel concerns the number of k-length common continuous substrings in two strings, leading to overlook their non-contiguous common subsequences.

3 Dataset, Experiments, and Results

3.1 Dataset

As described by Schonlau et al. [1], the dataset consists of 15,000 commands, from each of seventy different users, recoded over a time span of several months. Some users generated these many commands in a few days, others in a few months. Fifty of the seventy users were chosen as intrusion target, and the other twenty users as masqueraders. For the victims, the first 5,000 commands are contiguous and can be used for training a detector. The next 10,000 commands of the victims were randomly injected with commands issued by the twenty masqueraders. For simplicity the commands are grouped into blocks, with 100 commands per block. Each block of the last 10,000 commands of the intrusion target is either pure clean or pure masquerader. Schonlau et al. pointed that each of the fifty target users contained different numbers of masquerader blocks. Some users had not been attacked, whereas other users had as many as 24 attack blocks. Maxion et al. refer to this data scheme as the SEA (Schonlau Et Al.) configuration [3].

According to the SEA configuration, different masqueraders were injected into different users and not all users were injected with masquerader. The computer in real world may encounter similar masqueraders. However, in test scene, failure to run a consistent set of masqueraders against all users makes it difficult to draw useful conclusions from the hit and miss rates. Once some algorithms took a poor performance in detecting masquerade, we can’t know whether the failure is due to the victim, the masquerader or these algorithms. For this reason, Maxion et al. reconfigured the data provided by Schonlau et al. [3]. In the new configuration, the training data were remained the same as in the SEA configuration, i.e., the first 5,000 commands of a given target. However, the testing set was altered. The testing data consist of 245,000 commands for each user, which made up of the first 5,000 commands of other 49 users. This configuration regarded all 49 of the others as masqueraders against a given user. It is therefore referred to as the 1v49 (1 versus 49) configuration.

In view of the limitation of the SEA configuration, we use 1v49 configuration data, i.e. we just consider the first 5,000 commands of each user. For each user, his first 5,000 commands are treated as normal training data while other 49 users’ first 5,000 commands is considered as attack commands, amounting to 245,000 commands. In 1v49 configuration, there is no self command to detect for each user; consequently we just consider the numbers of true positive (detected intrusions) and false negative (undetected intrusions), neglecting that of true negative (self commands) and false positive (self commands mistakenly identified as intrusions).

3.2 Experiments

There are two stages to the masquerade detection: in the training stage, we scan the first 5,000 commands of each user to build up a normal command sequences mode for each user; after that we use the normal mode of each user to test the command sequences of other 49 users, which is called testing stage. Like Schonlau et al. [1], the

Table 1. The number of detected and successful intrusions for each attackers

User	Using gap-insensitive kernel		Using spectrum kernel	
	Detected intrusions	Successful intrusions	Detected intrusions	Successful intrusions
1	1766	684	1414	1036
2	1991	459	1861	589
3	1972	478	1799	651
4	1253	1197	1143	1307
5	1341	1109	1494	956
6	1926	524	1819	631
7	1422	1028	1326	1124
8	2027	423	1888	562
9	1241	1209	1418	1032
10	*2072*	*378*	*1578*	*872*
11	1631	819	1554	896
12	1478	972	1414	1036
13	1542	908	1318	1132
14	1123	1327	1186	1264
15	1363	1087	1316	1134
16	1516	934	1585	865
17	1953	497	1881	569
18	2058	392	1967	483
19	*652*	*1798*	*1297*	*1153*
20	1840	610	1758	692
21	2037	413	1586	864
22	1251	1199	1171	1279
23	2012	438	2085	365
24	*2127*	*323*	*1793*	*657*
25	827	1623	678	1772
26	*738*	*1712*	*584*	*1866*
27	2023	427	1838	612
28	1714	736	1623	827
29	1757	693	1817	633
30	2058	392	1904	546
31	1305	1145	1412	1038
32	*2264*	*186*	*1516*	*934*
33	2051	399	1985	465
34	1378	1072	1206	1244
35	1216	1234	1538	912
36	1871	579	1809	641
37	1271	1179	1103	1347
38	*354*	*2096*	*643*	*1807*
39	1823	627	1626	824
40	1514	936	1094	1356

41	1373	1077	1017	1433
42	1877	573	1678	772
43	1921	529	1701	749
44	1607	843	1489	961
45	1799	651	1245	1205
46	1819	631	1917	533
47	1179	1271	993	1457
48	1645	805	1516	934
49	1351	1099	1292	1158
50	1826	624	1574	876
Total	*80155*	*42345*	*74449*	*48051*

commands are grouped into blocks, with 100 commands per block in the latter stage. We then test each block to determine its abnormity.

There are a total of 856 unique commands in the dataset above described. Each user command is assigned an identifying number ranging from 1 to 856.

We designate the parameter q as 7 because it achieves best detection performance, the parameter k of spectrum kernel as 3, and SMO [11] as the learning procedure of SVMs.

3.3 Results

Different from Schonlau et al. [1] and Maxion et al. [3], we depict the results from the perspective of attacker, listing the number of their successful and failed attacks, as shown in the following table. We also list the performance of SVMs using spectrum kernel for comparison.

As shown in the above table, the attack sequences of user 32, 24, and 10 are insensitive to gaps. Using gap-insensitive kernel we can easily detect their attackers from normal command sequences. The spectrum kernel used the sequence information but ignored the command insertion, resulting in worse performance of detecting these attackers. However, we can see that the gap-insensitive kernel is not applicable to some users. We attribute the poor performance to as follows. The whole 5,000 commands of some users had been generated in a few months, implying that there is no sequence information in their command sets. Therefore, the string kernels including gap-insensitive and spectrum kernel both achieved inevitably poor performance, like user 26. The second reason for poor performance is that some attackers such as user 38 shared many command tuples with the victims, hiding their purpose. Moreover, some attackers' command sequences are sensitive to the gaps in their sequences, resulting in the poor performance on gap-insensitive kernel but better performance on spectrum kernel, like user 19. Nevertheless, the gap-insensitive kernel outperforms spectrum kernel on the whole, as depicted by the above table.

4 Conclusions

We addressed in this paper the problem of "How to detect masquerading that common or meaningless commands are inserted purposely in order to conceal intruding?" For the purpose of detecting masquerader, we presented a string kernel, named gap-insensitive kernel. The experiment results showed that this kernel could overlap the ordinary commands inserted by intruder so as to detect the real intention of intruder. However, the disadvantage of this kernel is also evident. Sometimes, the gap-insensitive kernel failed to distinguish wicked behavior from normal behaviors due to its insensitive nature.

In our future work, we plan to present a string kernel considering not only the length of string x and its subsequence s, but the gaps in s, and use a parameter to determine the trade off between the length and the gaps.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China (No. 60442002) and the Science and Technology Foundation of Beijing Jiaotong University (No. 2004SM010).

References

1. Schonlau, M., DuMouchel, W., Ju, W.-H., Karr, A.F., Theus, M., Vardi, Y.: Computer intrusion: Detecting masquerades. Statistical Science **16** (2001) 58–74
2. Vapnik, V. N.: The nature of statistical learning theory. Springer, Berlin Heidelberg New York (2000)
3. Maxion, R.A., Townsend, T.N.: Masquerade detection augmented with error analysis. IEEE Transactions on Reliability **53** (2004)
4. Shawe-Taylor, C., Cristianini, N.: Kernel methods for pattern analysis. Cambridge University Press (2004)
5. Haussler, D.: Convolution kernels on discrete structures. Technical report, UC Santa Cruz (1999)
6. Leslie, C., Eskin, E., Noble, W.S.: The spectrum kernel: a string kernel for SVM protein classification. In: Proceedings of the pacific biocomputing Symposium (2002)
7. Leslie, C., Eskin, E., Weston, J., Noble, W.S.: Mismatch string kernels for SVM protein classification. In: Proceedings of Neural Information Processing Systems (2002)
8. Lodhi, H., Saunders, C., Shawe-Taylor, C., Cristianini, N., Watkins, C.: Text classification using string kernels. Journal of Machine Learning Research **2** (2002) 419-444
9. Tian, S.F., Yu, J., Yin, C.H.: Anomaly detection using support vector machines. In: Proceedings of Advances in Neural Networks (2004) 592-597
10. Myers, G.: A fast bit-vector algorithm for approximate string matching based on dynamic programming. Journal of the ACM **3** (1999) 395-415
11. Platt, J.C.: Sequential minimal optimization: A fast algorithm for training support vector machines. Technical Report MSR-TR-98-14, Microsoft Research (1998)

A New Method for Linear Ill-Posed Problems: Iteration Method by Rectifying Eigenvalue

Yugang Tian[1], Peijun Shi[1], Xinzhou Wang[2], and Kun Qin[3]

[1] College of Resources Science & Technology,Beijing Normal University,
Beijing,100875 , China
ygangtian@ires.cn, spj@bnu.edu.cn

[2] College of Geodesy and Geomatics, Wuhan University, Wuhan, 430079,China
xzhwang@sgg.whu.edu.cn

[3] College of Remote Sensing Information Engineering, Wuhan
University,Wuhan,430079,China
qqqkkk@263.net

Abstract. In order to overcome the weaknesses of Regularization Method for linear ill-posed problem, the authors suggest a new method named Iteration Method by Rectifying Eigenvalue (IMRE) in this paper. Firstly, the rigorous theoretical proofs that IMRE can achieve convergent and unbiased solution are given. Then an effective method called L-Curve method is introduced to determine parameter α in IMRE. Thirdly, a computing program is designed. Finally an example is given to testify the advantages of IMRE by the above program.

1 Introduction

With the fast development in data collection and storage in the recent decade, we can obtain large quantity of data, but it is hard to get knowledge we want. Data mining technology is studied to deal with this problem. Classification and prediction are two kinds of data analysis methods. They can be used to extract the models of important data classes or to predict the trend of future data. Least Square Regression is applied widely in classification and prediction. Ill-posed problems usually appear in linear Least Square Regression (LSR) because of approximate linear relationship among parameters selected and computations techniques. Even if a tiny disturbance in observation data, there may cause enormously fluctuate in parameters estimation results [1]. Tiknonov proposed the Regularization Method to solve ill-posed problem [2]. The method can meliorate ill-posed problem in some extent, but two difficult problems remain unresolved. 1) The method breaks equivalence relation of Least Square Estimation of ill-posed problems and leads to biased solution. 2) Regularization Parameter α is determined optionally [3]. In this paper, the authors suggest a new method named Iteration Method by Rectifying Eigenvalue (IMRE), in which a proper parameter α selection method called L-Curve method is introduced to overcome the shortage of regularization method.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 331 – 338, 2005.

2 Ill-Posed Problems and Regularization Method

Let's see the following equation：

$$\mathrm{A}f_* = F \quad f_* \in \mathbf{X}, F_* \in \mathbf{B} \tag{1}$$

Where $\boldsymbol{X}$ is a function set. ρ_X and ρ_B are distance measure defined on space $\boldsymbol{X}$ and $\boldsymbol{B}$ respectively.

If $\forall \varepsilon > 0$ and $\exists \delta(\varepsilon) > 0$， then for $\forall F_1, F_2 \in \boldsymbol{B}$， when $\rho_B(F_1, F_2) \le \delta(\varepsilon)$ and $\rho_X(f_1, f_2) \le \varepsilon$ ($f_1, f_2 \in \boldsymbol{X}$), the solution of $\mathrm{A}f_* = F_*$ on $(\boldsymbol{X}, \boldsymbol{B})$ is stable, where $f_* \in \boldsymbol{X}$.

If the solution of $\mathrm{A}f_* = F_*$ on $(\boldsymbol{X}, \boldsymbol{B})$ satisfies two conditions.1) $\forall F_* \in \boldsymbol{B}$, there exists an exclusive $f_* \in \boldsymbol{X}$, let $\mathrm{A}f_* = F_*$. 2) The solution of $\mathrm{A}f_* = F_*$ on $(\boldsymbol{X}, \boldsymbol{B})$ is stable. Then the problem to solve the solution of $\mathrm{A}f_* = F_*$ on $(\boldsymbol{X}, \boldsymbol{B})$ is well-posed, otherwise, ill-posed [1].

In this paper, we will discuss the ill-posed problem that there exists an exclusive solution but not stable.

The main idea of regularization method is to minimize the following functional:

$$R^*(f_*) = \| \mathrm{A}f_* - F_* \|^2 + r(\delta)\Omega(f_*) \tag{2}$$

where $\Omega(f_*)$ is a certain kind of functional，and $r(\delta)$ is a proper constant at certain noise level. If substituting $R(f_*)$ with minimum $R^*(f_*)$,then when $\delta \to 0$, $f_\delta \to f$. The above method is called Regularization Method.

Take Least Squares Regression for example：we assume that sample set D_n ={ $(\vec{x}_i, y_i)$ ：(i=1，2，…， n)， $\vec{x}_i \in R^m$ ， $y_i \in R$ }，the regression equation is given in Eq. (3)：

$$f(\vec{x}_*) = W\vec{x}_* + b \tag{3}$$

where $W \in R^m$ ， $b \in R$ and they denote weight vector and threshold respectively. According to Tikhonov Regularization Method [2], we can get object function :

$$E_r(W,b) = \frac{1}{2}(y - AW - bu)^T(y - AW - bu) + \frac{\alpha}{2}(W^TW + b^2)$$

$$\text{Let} \quad \begin{cases} \dfrac{\partial E_r(W,b)}{\partial W} = 0 \\ \dfrac{\partial E_r(W,b)}{\partial b} = 0 \end{cases} \Rightarrow \begin{bmatrix} A^TA+\alpha & A^Tu \\ u^TA & u^Tu+\alpha \end{bmatrix} \begin{bmatrix} W \\ b \end{bmatrix} = \begin{bmatrix} A^Ty \\ u^Ty \end{bmatrix} \tag{4}$$

Where $A=\begin{pmatrix}\vec{x}_1\\ \vdots\\ \vec{x}_n\end{pmatrix}_{n\times1}=\begin{pmatrix}a_{11}&\cdots&a_{1m}\\ \vdots&\ddots&\vdots\\ a_{n1}&\cdots&a_{nm}\end{pmatrix}_{n\times m}$, $y=\begin{pmatrix}y_1\\ \vdots\\ y_n\end{pmatrix}_{n\times1}$, $u=\begin{pmatrix}1\\ \vdots\\ 1\end{pmatrix}_{n\times1}$ .Where α is called Regularization Parameter and if it is an positive number large enough and the coefficient matrix of Eq. (4) is not singular, then we can obtain the exclusive solution $\begin{bmatrix}W\\ b\end{bmatrix}$. And W^TW+b^2 is called Regularizing Term.

It is obvious that adding of regularizing term leads to a biased solution compared with Least Square Regression. It still needs further study on how to select Regularization Parameter α, so we will present a new method named IMRE in this paper.

3 Iteration Method by Rectifying Eigenvalue (IMRE)

3.1 Basic Idea of IMRE

If adding $\alpha\begin{bmatrix}\mathbf{w}\\ b\end{bmatrix}$ to the right side of Eq.(4), then:

$$\begin{bmatrix}A^TA & A^Tu\\ u^TA & u^Tu\end{bmatrix}\begin{bmatrix}W\\ b\end{bmatrix}+\alpha\begin{bmatrix}W\\ b\end{bmatrix}=\begin{bmatrix}A^Ty\\ u^Ty\end{bmatrix}+\alpha\begin{bmatrix}W\\ b\end{bmatrix} \tag{5}$$

Let $\begin{bmatrix}A^TA & A^Tu\\ u^TA & u^Tu\end{bmatrix}=H^TH=\underset{t\times t}{N}$, $\begin{bmatrix}W\\ b\end{bmatrix}=\underset{t\times1}{\hat{X}}$, $\begin{bmatrix}A^Ty\\ u^Ty\end{bmatrix}=H^Ty=\underset{t\times1}{U}$, then:

$$(N+\alpha E)\hat{X}=U+\alpha\hat{X} \tag{6}$$

Where $H=[A\quad u]$, $t=m+1$, and E is a t-order unit matrix.

For both sides contain parameter $\hat{X}$, iteration method is used to solve Eq.(6).

$$\hat{X}^{(k)}=(N+\alpha E)^{-1}(U+\alpha\hat{X}^{(k-1)}) \tag{7}$$

$$\text{Let } p=(N+\alpha E)^{-1} \tag{8}$$

Then Eq.(7) will be:

$$\hat{X}^{(k)}=(p+\alpha p^2+\cdots+\alpha^{k-1}p^k)U+\alpha^kp^k\hat{X}^{(0)}$$

Let $q=\alpha p=(\alpha^{-1}N+E)^{-1}$, then

$$\hat{X}^{(k)}=\alpha^{-1}(q+q^2+\cdots+q^k)U+q^k\hat{X}^{(0)} \tag{9}$$

Where $\hat{X}^{(0)}$ is the initial value of unknown parameter $\hat{X}$. Eq.(7) or Eq.(9) is called Iteration Method by Rectifying Eigenvalue (IMRE).

3.2 Proofs for Convergent and Unbiased Estimation of IMRE

For Eq.(9), we have three theorems in the following [3].

Theorem 1. Whatever pose N is, well-posed or ill-posed, the following equation is identical.

$$Rank(\alpha^{-1}N + E) = t \tag{10}$$

Therefore $\alpha^{-1}N + E$ is a matrix with full rank.

Theorem 2. If $Rank(N) = t$, then $\forall \hat{X}^{(0)}$,we always have the following conclusion.

$$\lim_{k\to\infty} \hat{X}^{(k)} = N^{-1}U \tag{11}$$

Theorem 3. If $Rank(N) = r < t$ and $\hat{X}^{(0)}$=0, then:

$$\lim_{k\to\infty} \hat{X}^{(k)} = N^{-}U \tag{12}$$

In order to testify Theorem 1~3, we should give Lemma 1~7 in advance [3].

Lemma 1. If N is t-order positive definite matrix, then its eigenvalue satisfies:

$$\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_t > 0 .$$

If $Rank(H) = r < t$ and N is t -order semi-positive matrix, then its eigenvalue satisfies: $\lambda_1 \geq \cdots \geq \lambda_r > \lambda_{r+1} = \cdots = \lambda_t = 0$.

Lemma 2. If $Rank(N) = t$, then $\|q\|_2 < 1$; if $Rank(N) = r < t$, then $\|q\|_2 < 1$.

Lemma 3. If $\|q\|_2 < 1$, then matrix power series

$$E + q + q^2 + \cdots + q^k + \cdots$$

is of absolutely convergence, and its summation is $(E - q)^{-1}$.

Lemma 4. If $\|q\|_2 < 1$, then q is of power convergence.

$$\lim_{k\to+\infty} q^k = 0 \tag{13}$$

Lemma 5. If H is a t-order constant matrix, $\{B_k\}_{k\geq 1}$ is a t-order matrix series, and $\lim_{k\to+\infty} B_k = B$, then:

$$\lim_{k\to+\infty} (HB_kH) = HBH \tag{14}$$

Lemma 6. If $Rank(N) = r < t$, then:

$$\lim_{k\to+\infty}(q^{k+1} - q^{k+2}) = 0 \tag{15}$$

Lemma 7. If $Rank(N) = r < t$, then:

$$E + q + q^2 + \cdots + q^k + \cdots = (E - q)^{-} \tag{16}$$

Proof of Theorem 1

$$\because \left|\alpha^{-1}N + E\right| = |Q||D + E|\left|Q^T\right| = |Q||D_1|\left|Q^T\right|$$

$$= (\alpha^{-1}\lambda_1 + 1)(\alpha^{-1}\lambda_2 + 1)\cdots(\alpha^{-1}\lambda_t + 1) > 0$$

$\therefore Rank(\alpha^{-1}N + E) = t$.

Proof of Theorem 2

According to **Lemma 4** and Eq. (9),we have:

$$\hat{X}^{(k)} = \alpha^{-1}(q + q^2 + \cdots + q^k)U + q^k\hat{X}^{(0)}$$

$$= \alpha^{-1}(E + q + q^2 + \cdots + q^k)U - \alpha^{-1}U$$

and by **Lemma 1~3,** we have:

$$\lim_{k\to\infty}\hat{X}^{(k)} = \alpha^{-1}(E + q + \cdots + q^k + \cdots)U - \alpha^{-1}U = \alpha^{-1}[(E - q)^{-1} - E]U$$

$$= \alpha^{-1}[(E - q)^{-1}(E - (E - q))]U = \alpha^{-1}[q^{-1}(E - q)]^{-1}U = \alpha^{-1}(q^{-1} - E)^{-1}U$$

$$= (q^{-1}\alpha - \alpha E)^{-1}U = N^{-1}U$$

Proof of Theorem 3

If $\hat{X}^{(0)} = 0$, let $B_k = E + q + \cdots + q^k$, Then Eq.(9) changes into:

$$\hat{X}^{(k)} = \alpha^{-1}(q + q^2 + \cdots + q^k)U = \alpha^{-1}qB_{k-1}U$$

by **Lemma 7**, we have:

$$\lim_{k\to\infty}\hat{X}^{(k)} = \alpha^{-1}\lim_{k\to\infty}qB_{k-1}U = \alpha^{-1}q(\lim_{k\to\infty}B_{k-1})U = \alpha^{-1}q(E - q)^{-}U$$

$$= \alpha^{-1}[(E - q)q^{-1}]^{-}U = \alpha^{-1}(q^{-1} - E)^{-}U$$

$$= N^{-}U$$

From theorem 1~3, we can draw a conclusion that Eq. (9) is convergent to Least Square Estimation solution of Eq. (3), so we can get a unbiased solution for linear ill-posed problems by IMRE.

3.3 Selection of Parameter α of IMRE with the L-Curve Method

In this section we introduce the L-Curve method to select parameter α of IMRE. $\left\|\mathbf{y} - \mathbf{A}\mathbf{w} - b\mathbf{u}\right\| = \left\|\mathbf{y} - H\hat{X}\right\|$ and $\left\|\hat{X}\right\|$ are both functions of parameter α.

If take $\left\|\mathbf{y}-H\hat{X}\right\|$ as horizontal coordinate and $\left\|\hat{X}\right\|$ as vertical coordinate, for different α, we can obtain many points ($\left\|\mathbf{y}-H\hat{X}\right\|,\left\|\hat{X}\right\|$), thus a curve is obtained by curve fitting with these points, whose shape is similar to “L”. Therefore, this method of selecting parameter α is named as L-curve method.

The key of this method is to get the point of maximum curvature of the L-curve and the corresponding parameter α is what we want.

Let $\eta=\left\|\hat{X}\right\|^2, \rho=\left\|\mathbf{y}-H\hat{X}\right\|^2$. If make a logarithm transform to both sides of Eq.(17), then

$$\eta=\left\|\hat{X}\right\|^2, \rho=\left\|\mathbf{y}-H\hat{X}\right\|^2 \tag{17}$$

$$\hat{\eta}=\log_{10}\eta=2\log_{10}\left\|\hat{X}\right\|, \hat{\rho}=\log_{10}\rho=2\log_{10}\left\|\mathbf{y}-H\hat{X}\right\| \tag{18}$$

the L-curve is fitted by points $(\hat{\rho}/2,\hat{\eta}/2)$. We can compute curvature κ at any point of the L-curve according to Eq. (19).

$$\kappa=2\frac{\hat{\rho}'\hat{\eta}''-\hat{\rho}''\hat{\eta}'}{((\hat{\rho}')^2+(\hat{\eta}')^2)^{3/2}} \tag{19}$$

Where $\hat{\rho}', \hat{\eta}', \hat{\rho}'', \hat{\eta}''$ are the first order and second order derivative of $\hat{\rho}$ and $\hat{\eta}$ separately. We can obtain $\kappa_{\max}$ by maximizing Eq. (19). The corresponding point with $\kappa_{\max}$ is what we want, and the corresponding α with this point is what we look for by the L-curve method[4][5].

3.4 Computing Program of IMRE

IMRE can be implemented easily and its computing program is as follows:

Step 1. Comput α according to the L-Curve methods and input $\delta=\ldots$;

Step 2. Comput $p=(N+\alpha E)^{-1}$, let $M=p$;

Step 3. Comput $M=p+\alpha Mp$;

Step 4. Comput $\hat{X}=MU$;

Step 5. If $\left|\hat{X}^{(k+1)}-\hat{X}^{(k)}\right|>\delta \quad (k=1,2,\ldots)$, where $\delta(\delta>0)$ is error threshold, then turn to step 3; else output $\hat{X}$.

4 An Example

$$H = \begin{bmatrix} 2.00 & -5.00 & 1.00 & 1.00 & -9.50 \\ -2.00 & 4.00 & 1.00 & -1.05 & 8.50 \\ -2.00 & 1.00 & 1.00 & -1.00 & 2.40 \\ -1.00 & 2.50 & 4.00 & -0.50 & 7.00 \\ -1.00 & 3.20 & 4.00 & -0.50 & 8.40 \\ 1.00 & 1.00 & -3.00 & 0.40 & 0.49 \\ 3.00 & 7.00 & -3.00 & 1.50 & 12.70 \\ 5.00 & -1.00 & -2.00 & 2.50 & -3.00 \\ 4.00 & 2.00 & -2.00 & 2.01 & 3.00 \\ 4.00 & 3.00 & -2.00 & 2.00 & 5.00 \end{bmatrix} \qquad \mathbf{y} = \begin{bmatrix} -10.50 \\ 10.45 \\ 1.40 \\ 12.00 \\ 14.10 \\ -0.11 \\ 21.20 \\ 1.50 \\ 9.01 \\ 12.00 \end{bmatrix}$$

$N = H^T H$, $U = H^T y$, $cond(N) = 1.2892 \times 10^5 \gg 1000$, it is a highly ill-posed problem. It is evident that their true value is $X = [1,1,1,1,1]^T$.

We can get $\alpha = 0.6343$ for this sample[5] by the L-curve method. Regularization Method and IMRE are applied for this sample with $\alpha = 0.6343$ respectively. The solutions are as follow:

$$\hat{X}_R = [\ 1.1746, 0.4330, 0.8370, 0.6021, 1.2807]^T$$

$$\left\| \hat{X}_R - X \right\| = 0.7847$$

$$\hat{X}_I = [\ 1.1922, 0.4326, 0.8624, 0.6153, 1.2819]^T$$

$$\left\| \hat{X}_I - X \right\| = 0.7780$$

where R and I denote the first letter of Regularization Method and IMRE respectively. From this sample, we can see that the L-curve method is an effective method for parameter α selection.

5 Conclusions

In this paper, we suggested a new method named IMRE to solve the linear ill-posed problem in classification and prediction of Data Mining. The solution of IMRE is proved to be convergent and unbiased theoretically in this paper. The L-curve method proposed in this study for selecting parameter α of IMRE can position the exact α and it is more applicable than other methods with rigorous mathematical deduction. An example of linear ill-posed problem is also given in this study. The major conclusions are as follows:

- IMRE is a better method than Regularization Method because it can achieve convergent and unbiased solution.
- IMRE does not require setting initial value of $\hat{X}^{(0)}$. To any initial value of $\hat{X}^{(0)}$, if iteration numbers is big enough, then the value of $q^{k}\hat{X}^{(0)}$ in Eq. (9) always comes to zero.
- IMRE can be easily carried out according to section 3.4 by regular mathematical software.

Acknowledgements

This paper is supported by the National Natural Science Foundation of China for Outstanding Youth Scholars under Grant No. 40425008 of National Science Foundation of China and Chinese-France Cooperation Project under Grant No.P2R0303 of the Ministry of Science and Technology of China.

References

1. Tikhonov A N, Arsenin V Y.:Solution of Ill-posed Problem. Winston and Sons, Washington DC(1977)
2. Tikhonov A N, Goncharsky A V.:Ill-posed Problems in the Natural Sciences. Translated from Russian by Bloch M, MIR publishers, Moscow(1987)
3. X.Z.,Wang,D.Y.,Liu,Q.Y.,Zhang,H.N.,Huang.:The Iteration by Correcting Characteristic Value and Its Application in Surveying Data Processing.Journal of Heilongjiang Institute of Technology, vol.15.Heilongjiang(2001)3-6(In Chinese)
4. Hansen,P.C.:Analysis of Discrete Ill-posed Problems by means of the L-curve.SIAM Review,vol.34.No.4.(1992)561-580
5. Z.J.,Wang.:Research on the Regularization Solutions of Ill-posed Problems in Geodesy [dissertation].Institute of Geodesy and Geophysics, Chinese Academy of Sciences, Wuhan,China(2003)21-27 (In Chinese)

A Non-VSM kNN Algorithm for Text Classification

Zhi-Hong Deng and Shi-Wei Tang

National Laboratory on Machine Perception,
School of Electronics Engineering and Computer Science,
Peking University, Beijing 100871, China
zhdeng@cis.pku.edu.cn, tsw@pku.edu.cn

Abstract. The text classification problem, which is the task of assigning natural language texts to predefined categories based on their content, has been widely studied. Traditional text classification use VSM (Vector Space Model), which views documents as vectors in high dimensional spaces, to represent documents. In this paper, we propose a non-VSM kNN algorithm for text classification. Based on correlations between categories and features, the algorithms first get k F-C tuples, which are the first k tuples in term of correlation value, from an unlabeled document. Then the algorithm predicts the category of the unlabeled documents via these tuples. We have evaluated the algorithm on two document collections and compared it against traditional kNN. Experimental results show that our algorithm outperforms traditional kNN in both efficiency and effectivity.

1 Introduction

In recent years, we have seen a great growth in the volume of online text documents available on the World Wide Web. It has been forecasted that these documents with other unstructured data will become the predominant data type stored online [1]. These Web documents contain rich textual information, but the rapid growth of the Internet has made it increasingly difficult for users to locate the relevant information on the Web. This provides a huge opportunity to make more effective use of these document galleries and there is a growing need for developing efficient techniques to help users find useful information. Automatic text classification, which is the task of assigning natural language texts to predefined categories based on their context, can help organize and search information in the tremendous gallery of Web documents.

A growing number of statistical classification methods and pattern recognition techniques have been applied to text categorization in recent years. As mentioned in [2] [3] [4], kNN is one of the best algorithms in terms of classification performance (effectivity). In spite of its advantage, kNN have an obvious drawback that spends a lot of time for classifying unknown documents. The reason results from two aspects. The first one is that kNN must calculate the similarities of unlabeled documents with each training (labeled) documents. The second one is that documents are represented by VSM [5], where a document is regarded as a vector in feature spaces, and similarities are quantified by *cosine* of document vectors. In text categorization, tens of thousands training documents are often available and the feature spaces usually contain more than ten thousand features. All these make traditional kNN time consuming.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 339 – 346, 2005.

In this paper, we will present a new algorithm originated from kNN. We call the algorithm non-VSM kNN because it doesn't adopt VSM for the representation model of documents. In non-VSM kNN, a document is regarded as set of elements, each of which includes a feature and the weigh of the feature in the document. Further, non-VSM kNN use the correlations between features and categories instead of all training documents for predicting unknown documents. These characteristics make non-VSM kNN more efficient than kNN. In addition, non-VSM kNN adopt an adaptive strategy of feature selection during classifying documents. This makes it more effect than kNN.

The reminder of this paper is organized as follows. Section 2 describes VSM and kNN. Section 3 describes some basic definitions and non-VSM kNN algorithm. Section 4 experimentally compares non-VSM kNN with kNN on two text collections according to efficiency and effectivity. Section 5 summarizes our study and points out some future research.

2 VSM and kNN

VSM is a widely used model for representation of documents in text classification. kNN is a famous algorithm applied to classifying documents. In this session, we describe VSM and kNN in brief as background.

The core idea of VSM is that it views documents as vectors in high dimensional geometry space. Dimensions of the space are composed of features, which are terms extracted from documents. We hereby call the space feature space. Let $F = \{f_1, \ldots, f_n\}$ be the set of features[1] selected from all documents. The vector for a document d_i is represented by $V_i = (w_{i1}, \ldots, w_{in})$. Each element in V_i can be best qualified by *tf*idf*. For the definition of *tf*idf*, please refer to [6].

*k*NN is an instance-based learning algorithm that has been applied to text classification since the early days of research. In this classification paradigm, the unlabelled document d is assigned to the category c_i if c_i has the biggest similarity score to d among all categories. The similarity score of a document d to a category c_j is computed as:

$$s(d,c_j) = \sum_{d_i \in kNN} sim(d,d_i)y(d_i,c_j) \tag{1}$$

$sim(d, d_i)$ is the similarity between the document d and the training document d_i. Let $V = (w_1, \ldots, w_n)$ and $V_i = (w_{i1}, \ldots, w_{in})$ are vector representations of d and d_i respectively, $sim(d, d_i)$ can be quantified by the *cosine* of the angle between V and V_i. That is,

$$sim(d,d_i) = \frac{V \bullet V_i}{\|V\|_2 \|V_i\|_2} = \frac{\sum_{k=1}^{n} w_k \times w_{ik}}{\sqrt{\sum_{k=1}^{n} w_k^2}\sqrt{\sum_{k=1}^{n} w_{ik}^2}} \tag{2}$$

[1] In this paper, features are words selected from documents.

$d_i \in k$NN stands for that d_i is one of the k nearest neighbors to d in the light of the function *sim*(). $y(d_i, c_j) \in \{0,1\}$ is the classification for document d_i with respect to category c_j ($y(d_i, c_j)$ =1 for YES, and $y(d_i, c_j) = 0$ for NO). Finally, based on these similarity calculated from formula (1), the category of document is assigned by

$$\arg\max_{j=1,\ldots,m} (s(d, c_j)) \tag{3}$$

where $c_1, \ldots, c_m$ are the predefined categories.

3 Non-VSM kNN

This section explains our algorithm. First, we bring forward some foundational concepts. After that, we describe the non-VSM kNN algorithm in detail. Let $C = \{c_1, \ldots, c_m\}$ be the set of predefined categories, $F = \{f_1, \ldots, f_n\}$ be feature set, and $TD = \cup D_i$ be the set of documents, where D_i is the set of documents labeled category c_i.

3.1 Some Concepts

In our algorithm, a new kind of representation model of documents is adopted. We call this model SFW, which means set of features with weight. The following definition describes the content of SFW.

Definition 1: For the SFW model, a document $d_j \in TD$ is represented by $\{<f_i, w_{ij}>\}$,where $f_i \in F$ is a feature and w_{ij} is the weigh of f_i in d_j. Without loss of generalization, we write $d_j = \{<f_i, w_{ij}>\}$.

w_{ij} stands for the degree that f_i expresses the content of d_j. Intuitionally, the more frequently f_i occurs in d_j, the more f_i expresses the content of d_j. Therefore, the frequency of f_i in d_j, which is the numbers of times the feature f_i is mentioned in the text of the document d_j, is a good estimation of w_{ij}. However, the frequencies of features in long documents are usually bigger than these in short documents. For the sake of impartiality, w_{ij} is quantified by normalized frequency of f_i in d_j in this paper. The formula for measuring w_{ij} is defined as follows:

$$w_{ij} = \frac{freq_{ij}}{\max_k freq_{kj}} \tag{4}$$

where the $freq_{ij}$ stands for the frequency of f_i in d_j and the maximum is computed over all features which are mentioned in the text of the document d_j.

Another import concept in our algorithm is F-C tuples, which include three elements, a feature, a category and a value.

Definition 2: A F-C tuple is a 3-element tuple $<f_i, c_j, v_{ij}>$, where $f_i \in F$ is a feature, $c_j \in C$ is a category, and v_{ij} is a value. v_{ij} stands for the correlation between feature f_i and category c_j.

There are numbers of statistical methods for measuring the correlation between features and categories. These methods include *information gain* (IG) [7], *mutual in-*

formation (MI) [7], χ^2 *statistic* (CHI) [7], *Odds Radio* [8] and *Category Relevance Factor* (CRF) [9]. As mentioned in [9] and [10], we found that *Category Relevance Factor* is the best method for measuring correlation between a feature and a category, especially in text classification. In this paper, v_{ij} is quantified by *Category Relevance Factor* as follows:

$$v_{ij} = CRF(f_i, c_j) = \log \frac{X/Y}{U/V} \tag{5}$$

where X is the number of documents that contain feature f_i and are labeled category c_j, Y is the number of documents that are labeled category c_j, U is the number of documents that contain feature f_i and are not labeled category c_j, V is the number of documents that are not labeled category c_j. For a given feature f_i, we call $\{v_{ij} \mid 1 \le j \le m\}$ category correlation set of f_i,. The advantage of *Category Relevance Factor* is that it synthetically considers the frequencies of a feature in both positive documents ($\in D_i$) and negative documents ($\notin D_i$).

3.2 Description of Classification Algorithm

The core idea of non-VSM kNN is that the similarity of document d_i and category c_j is measured by features in d_i and the correlation valuses between the features and c_j. It is well known that there are lots of noises in the features [7]. If all features are involved in measuring the similarity, there may be some mistakes. These mistakes may make the similarity far from real one. Therefore, we just use the top-k F-C tuples, correlation values of which are the biggest, to quantify the similarity. Given an integer k, the similarity of document d_i and category c_j is given by

$$sim(d_i, c_j) = \sum_{<f_l, c_j, v_{lj}> \in top-k \quad F-C \quad tuples} w_{li} v_{lj} \tag{6}$$

where f_l is a feature occurring in document d_i.

non-VSM kNN includes two components: one for learning classifier and the other for classifying new documents. For the sake of description, we label the former ***Training_phrase*** and the latter ***Classifying_Phase***.

```
Training_Phase:
Input: training document collection $TD = \cup D_i$, $1 \le i \le m$, $D_i = \{$document $d \mid d$ are
labeled category $c_i\}$, feature set $F = \{f_1, f_2, \ldots, f_n\}$.
Output: set of F_C tuples $TUPS = \{<f_i, c_j, v_{ij}>\}$.
Step1. Set $TUPS = \varnothing$.
Step2. For $j$ = 1 to $m$ do
          For $i$ = 1 to $n$ do
            1.  Compute $v_{ij}$ according to formula (5);
            2.  $TUPS = TUPS \cup \{<f_i, c_j, v_{ij}>\}$;
          End_do
       End_do
Step3. Output $TUPS$.
```

```
Classifying_Phase:
Input: $TUPS = \{<f_i, c_j, v_{ij}>\}$, $F = \{f_1, f_2, \ldots, f_n\}$, unlabelled document $d_{new}$, and parameter $k$.
Output: $label$--- the category of $d_{new}$.
Step1. $label = null$, $SFW = \varnothing$, $tups\_of\_d_{new} = \varnothing$, $k\_tups = \varnothing$, $Sim\_Set = \varnothing$.
Step2. For each feature $f_i$ in $d_{new}$ do
        1. Compute $w_{inew}$ According to formula (4);
        2. $SFW = SFW \cup \{<f_i, w_{inew}>\}$;
        3. For $j = 1$ to $m$ do
                $tups\_of\_d_{new} = tups\_of\_d_{new} \cup \{<f_i, c_j, v_{ij}>\}$;
           End_do
    End_do
Step3. Sort tuples in $tups\_of\_d_{new}$ according to $v_{ij}$ in descend order. Use the first $k$ tuples to constitute $k\_tups$. That is, $k\_tups = \{<f_i, c_j, v_{ij}>|(<f_i, c_j, v_{ij}> \in tups\_of\_d_{new}) \wedge (\forall <f_x, c_y, v_{xy}> \in tups\_of\_d_{new} \wedge <f_x, c_y, v_{xy}> \notin k\_tups, v_{xy} \le v_{ij})\}$.
Step4. For $j = 1$ to $m$ do
        1. Based on $SFW$ and $k\_tups$, compute the similarity $sim(d_{new}, c_j)$ according to formula (6);
        2. $Sim\_Set = Sim\_Set \cup \{sim(d_{new}, c_j)\}$;
    End_do
Step5. Select the biggest value from $Sim\_Set$. Supposed it is $sim(d_{new}, c_x)$. Then regard $c_x$ as the category of $d_{new}$. That is, $label = c_x$. Output $label$.
```

4 Experiments

In this section, we present a performance comparison of non-VSM kNN with traditional kNN in both efficiency and effectivity. All the experiments are performed on a 1.4-GHz Pentium notebook PC machine with 512 megabytes main memory, running on Microsoft Windows XP. All programs are written in Microsoft Visual C++ 6.0.

Document Collections used to evaluate the classification algorithms are taken from Newsgroups-18828[2] and Ohscal[3]. Each document of two collections is labeled one category. Newsgroups-18828 contains 18828 documents, and Ohscal contains 11162 documents. We randomly selected 14121 documents from Newsgroups-18828 to construct the training set and regarded the rest as test documents. For Ohscal, 8930 documents were selected to construct the training set and the rest constituted test set. It should be notice that the frequency of document distribution of each category in training set is the same as that in original collection for both partitions. For these two training set, we used a stop-list to remove common words, and the words were stemmed using Porter's suffix-stripping algorithm [11]. Furthermore, we also skip rare frequency words that occur in less than three documents.

Two factors are vital in evaluating the performance of classification algorithms. One is efficiency, which can be quantified by either time spent in learing (constructing) a

[2] http://www.ai.mit.edu/~jrennie/20Newsgroups/20news-18828.tar.gz

[3] http://www.cs.umn.edu/~han/data/tmdata.tar.gz

classifier or time spent in classifying unlabeled documents. The other is effectivity, which represents the capability of a classifier in predicting categories of unknown documents. As for efficiency, it is obvious that time for classifying is much more significant than time for learning. Therefore, we adopt runtime used for classifying test documents to measure efficiency in this paper. In terms of effectivity, we adopt the widely used *micro-averaging* F_1*-measure* [4].

The runtime of non-VSM kNN and traditional kNN as the parameter *k* (number of nearest neighbours) increase from 10 to 100 are shown in Figure 1 and Figure 2. NV_kNN is the abbreviation for non-VSM kNN. Figure 1 shows the runtime on Newsgroups-18828, and Figure 2 shows the runtime on Ohscal.

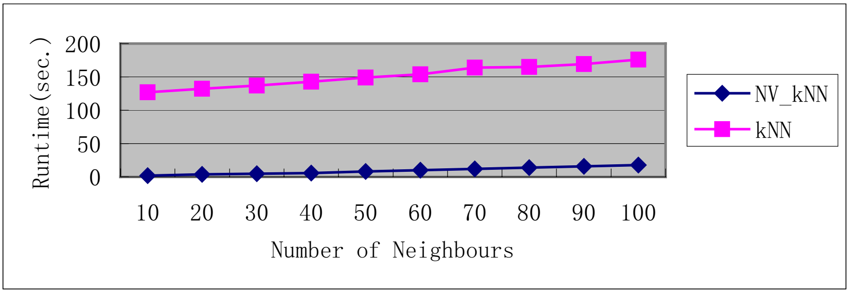


Fig. 1. Runtime of NV_kNN and kNN on Newsgroups-18828

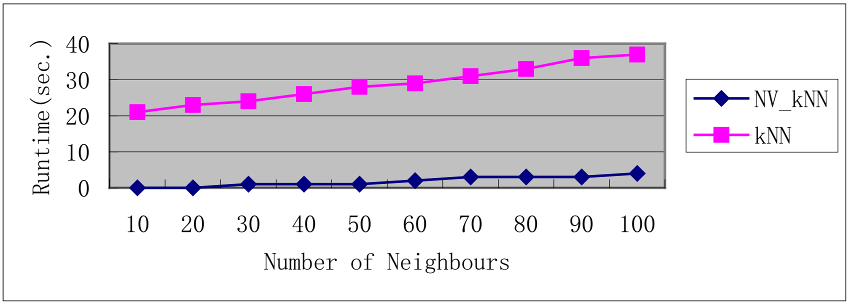


Fig. 2. Runtime of NV_kNN and kNN on Ohscal

In terms of Runtime, NV_kNN is about one or two order of magnitude faster than kNN in each parameter *k*. This is because all training documents are involved in predicting an unlabeled document. If average number of features occurring in a document is *m*, the computation for similarity based on *cosine* of two document vectors is O(*m*). Supposed that the number of training documents is *n*, the computation for all similarities based on *cosine* is O(*mn*). The computation for finding k nearest neighbours from *n* documents is about O(*nlogn*). Hence, the total computation complexity of traditional kNN is about O(*mn* + *nlogn*). In the same way, we can see that the computation complexity of NV_kNN is about O(*lmloglm*), where *l* is the number of categories. It is obvious that *l*<< *n* and *m*<< *n*. Therefore, *lmloglm* is much less than *mn*.

Figure 3 shows that the *micro-averaging* F_1 of NV_kNN and traditional kNN on Newsgroups-18828 as the parameter *k* increase from 10 to 100. It shows that the *micro-averaging* F_1 score of NV_kNN is always better than that of kNN on same parameter *k*. The same result can be derived from Figure 4, where the document collection is Ohscal. The reason may be that not all features but only useful features (at most *k* features) are adopted to predict unknown documents in NV_kNN. This lessens the negative affect of noise features and makes result of classification more effective.

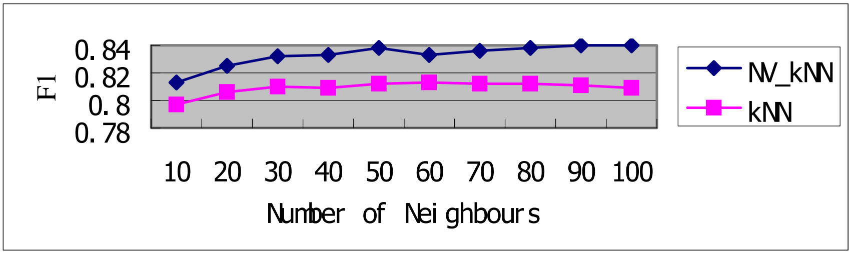


Fig. 3. *Micro-averaging* F_1 of NV_kNN and kNN on Newsgroups-18828

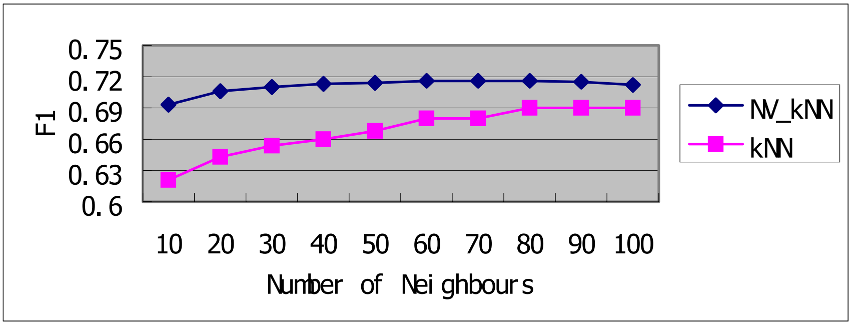


Fig. 4. *Micro-averaging* F_1 of NV_kNN and kNN on Ohscal

5 Conclusions

In this paper we have proposed a novel algorithm, non-VSM kNN originated from kNN, for classifying documents. As our experimental results on two text collections have shown, this algorithm outperforms traditional kNN in both efficiency and effectivity.

There are many ways to further improve the performance of this non-VSM kNN classification algorithm. First, in its current form it is not well suited to handle multi-modal classes. However, support for multi-modality can be easily implemented by using thresholding strategies [2] [12] [13] in automated text categorization. Second.

Much more effective statistic methods for quantifying the correlation between features and categories need to be further investigated in the future.

Acknowledgement. This research is supported by the National Natural Science Foundation of China under grant No. 60473051. We are also grateful to anonymous reviewers for their comments.

References

[1] Forrester Research. Coping with complex data. The Forrester Report, April 1995.

[2] Y. Yang. An evaluation of statistical approaches to text categorization. Journal of Information Retrieval, 1(1/2): 67-88, 1999.

[3] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In Proceedings of the 1998 European of conference on Machine Learning (ECML), pages: 137-142, 1998.

[4] Y. Yang, X. Liu. A re-examination of text categorization methods. In 22nd Annual Interna-tional ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'99), pages: 42-49, 1999.

[5] G. Salton and M. E. Lesk. Computer evaluation of indexing and text processing. *Journal of the ACM*, 1968, 15(1): 8-36.

[6] B. Y. Ricardo, R. N. Berthier. Modern Information Retrieval. ACM press, 1999.

[7] Y. Yang, J.P. Pedersen. A Comparative Study on Feature Selection in Text Categorization. In Proceedings of 14th International Conference on Machine Learning, pages: 412-420, 1997.

[8] D. Mladenic, M. Grobelnik. Feature Selection for Classification Based on Text Hierarchy. In Working notes of Learning from Text and the Web, Conference on Automated Learning and Discovery (CONALD'98), 1998.

[9] Z.H. Deng, S.W. Tang, D.Q. Yang, M. Zhang, X.B. Wu, M. Yang. A Linear Text Classification Algorithm Based on Category Relevance Factors. In Proceedings of 5th International Conference on Asian Digital Library (ICADL2002), Lecture Note Series in Computer Science (LNCS 2555) of Springer-Verlag, pages: 88 – 98, 2002.

[10] Zhi-Hong Deng, Shi-Wei Tang, Dong-Qing Yang, Ming Zhang, Li-Yu Li, Kun-Qing Xie. A Comparative Study on Feature Weight in Text Categorization. In Proceedings of The 6th Asia Pacific Web Conference (APWEB 2004), Lecture Note Series in Computer Science (LNCS 3007) of Springer-Verlag, pages: 588-597, 2004.

[11] M.F. Porter. An algorithm for suffix stripping. Program, 14(3): 130-137, 1980.

[12] D. D. Lewis. An evaluation of phrasal and clustered representations on a text categorization task. In 15th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), pages: 37-50, 1992.

[13] Y. Yang. A study on thresholding strategies for text categorization. In 24th Annual International of ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), pages: 137-145, 2001.

A Study on Text Clustering Algorithms Based on Frequent Term Sets

Xiangwei Liu[1,2] and Pilian He[1]

[1] Dept. of Computer Science, postbox 26#,
Tianjin University, Tianjin, 300072, China
[2] Dept. of Computer Science,
Tianjin Polytechnic University, Tianjin, 300160, China
lxw@eyou.com, plhe@tju.edu.cn

Abstract. In this paper, a new text-clustering algorithm named Frequent Term Set-based Clustering (FTSC) is introduced. It uses frequent term sets to cluster texts. First, it extracts useful information from documents and inserts into databases. Then, it uses the Apriori algorithm based on association rules mining efficiently to discover the frequent items sets. Finally, it clusters the documents according to the frequent words in subsets of the frequent term sets. This algorithm can reduce the dimension of the text data efficiently for very large databases, thus it can improve the accuracy and speed of the clustering algorithm. The results of clustering texts by the FTSC algorithm cannot reflect the overlap of texts' classes. Based on the FTSC algorithm, an improved algorithm—Frequent Term Set-based Hierarchical Clustering algorithm (FTSHC) is given. This algorithm can determine the overlap of texts' classes by the overlap of the frequent words sets, and provide an understandable description of the discovered clusters by the frequent terms sets. The FTSC, FTSHC and K-Means algorithms are evaluated quantitatively by experiments. The results of the experiments prove that FTSC and FTSHC algorithms are more efficient than K-Means algorithm in the performance of clustering.

1 Introduction

With the increment of the text information's contents, the text classifications now available cannot include all the contents, so text clustering has been one of the research hotspots in text mining. Traditional clustering methods, such as partitioning method, hierarchical method, density-based method, grid-based method, and model-based method, are not precise and efficient enough, especially in clustering the text items that have high dimensions or strong incidence relations. These methods have not given rational descriptions about clustering results. To solve these problems, we provide Frequent Term Set-based Clustering (FTSC) and Frequent Term Set-based Hierarchical Clustering (FTSHC) algorithms. These new algorithms extract feature terms, use Apriori algorithm to get frequent feature terms and cluster documents.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 347–354, 2005.

2 Feature Terms Extracting

At the beginning, we use the segmentation and part-of-speech tagging system provided by Beingjing University to segment the terms in the document set $D=\{d_1,d_2,\ldots,d_n\}$ with n documents. Then, we delete the useless terms, merge the terms of name and number, and extract the feature terms of documents. In the end, the document d_j is represented by $d_j=\{t_1,t_2,\ldots t_m\}$.

In order to consider the effect of feature terms in differentiating documents, we weight the discrimination value of feature terms based on the traditional term weighted value, and use the method mentioned in the reference [1] to find terms which can reflect the subject of the documents.

Term Discrimination Value (DV) is the term's ability to discriminate documents with different contents [2] when it is used to identify the content.

Suppose that d_i and d_j are vectors in a document set and they represent two documents in the set: $d_i=(w_{i1}, w_{i2},\ldots,w_{im})$, $d_j=(w_{j1}, w_{j2},\ldots,w_{jm})$. w_{ik} and w_{jk} are the component k of vectors, that is, the weight of the feature item k. The similarity between d_i and d_j can be expressed as $S(d_i, d_j)$:

$$S(d_i,d_j)=\frac{\sum_{k=1}^{m} w_{ik}\times w_{jk}}{\sqrt{\sum_{k=1}^{m}(w_{ik})^2\times\sum_{k=1}^{m}(w_{jk})^2}} \tag{1}$$

If we calculate the similarity of all the couples of the documents in the set, we can get an average similarity $\overline{S}(d_i,d_j)$. It is difficult to discriminate each document when $\overline{S}$ is high, and retrieval efficiency is low. If a term is used as feature term, it can reduce the average similarity of the document sets remarkably, or increase the average similarity value by deleting itself, the term has good ability to discriminate and it is an effective feature term.

Term Discrimination Value can be calculated by Eqn (2):

$$DV_k=\overline{S}_k-\overline{S} \tag{2}$$

where DV_k is the discrimination value of the feature term k, $\overline{S_k}$ is the average similarity after deleting k. The term's weight should be equal to its discrimination value. The weight can be calculated by Eqn (3)

$$w_k=tf_k\times idf_k\times DV_k \tag{3}$$

where w_k is the weight value of the feature term k, tf_k is the frequency of t_k appearing in the document D, df_k is the number of t_k appearing in D, idf_k is the reciprocal of df_k, called inverted document frequency. The feature of the document's subject can be embodied well if we add the term discrimination value to the weight of feature term. Redundant information can be deleted and the dimensions of documents can be reduced by choosing proper terms using Eqn (3).

3 Mining Frequent Feature Terms Based on Association Rules

The Apriori algorithm [3][4][5] based on association rules mining is used to discover the frequent itemsets efficiently. The association rule mining has two steps:

1. Find all the frequent itemsets in the transaction database. If the support of itemset X, support (X) ≥ *min_sup*, then X is a frequent itemset. Otherwise, X is not a frequent itemset.

2. Generate strong association rules from frequent itemsets. For each frequent itemset A, if B⊂A, B≠Ø, and support (A)/support (B) ≥ *min_conf*, then we have association rule B⇒ (A-B).

The processing object is documents in text database but is not transaction in transaction database. Accordingly we use the concept of feature term sets to replace itemsets of association rules and provide a new text clustering algorithm, FTSC.

4 Text Clustering Algorithm Based on Frequent Feature Term Sets

The feature terms are larger than the given threshold. The FTSC algorithm uses the frequent feature terms as candidate set and does not cluster document vectors with high dimensions directly. In this way, the efficiency of clustering is improved. At the same time, the subset of a frequent feature term set corresponds to a document category, which can provide more accurate description for clustering class.

4.1 FTSC Algorithm

Running the FTSC algorithm is a down-top procedure. We choose elements from the surplus frequent feature term sets and put them into an empty set, until all the elements corresponding to the conditions have been chosen. In each step of the calculation, the elements chosen by the FTSC are the frequent feature terms with minimal overlap information entropy.

We find the frequent itemsets based on the Apriori algorithm, and get a frequent feature term set *F* of document set *D* and all the terms in *F* are larger than *min_sup*. Each subset F_i in set *F* forms a clustering category. The terms contained in the subset is regarded as the description of the class in the document set *D*. In the clustering procedure, we hope that the clustering category is the least mutual overlap. Therefore, suppose that f_j is the number of all the frequent feature terms. These terms are mapped by document D_j of document set *D* as in Eqn (4)

$$f_j = \left|\{F_j \in R \mid F_j \subseteq D_j\right| \tag{4}$$

where | | is the number of the sets, *R* is a subset of *F*, F_j is the term set in *R* of document D_j. The smaller the overlap of class C_i with other classes is, the smaller f_j is. In the ideal condition, the documents in C_i only belong to one class, that means f_j =1. The overlap of class C_i with other classes is 0. We define the standard overlap of class C_i as its (f_j -1) average value, expressed by $SO(C_i)$ as in Eqn (5)

$$SO(C_i) = \frac{\sum_{D_j \in C_i} (f_j - 1)}{|C_i|} \tag{5}$$

The $SO(C_i)$ is easy to get, but the frequent itemsets have monotonicity (the subsets of frequent itemset are also frequent itemsets). When the document support *m*—itemset, it must support *m-1*—itemset, *m-2*—itemset…*2*—itemset, *1*—itemset. So a standard overlap of candidate itemset described by many items is usually greater than the one described by a few items.

To reduce this influence, we use information entropy to define the overlap of the class. The information entropy is a ratio distribution of a part of categories' support to the rest support. We use $EO(C_i)$ as entropy overlap of category C_i with Eqn (6).

$$EO(\mathrm{C}_i) = \sum_{D_j \in C_i} -\frac{1}{f_j}.\ln(\frac{1}{f_i}) \tag{6}$$

if f_j=1, entropy overlap is 0, the document D_j only belongs to one class and has not overlap with the other classes.

To find clusters with minimal overlap, we use FTSC algorithm to cluster documents. The time complexity of this algorithm is decided by the internal complexity of frequent feature terms.

Table1 is the first procedure of FTSC in a database including 16 documents.

Table 1. The first procedure of FTSC

Frequent feature term sets	Candidate clustering	EO
{athletics}	{D_1, D_2, D_4, D_5, D_6, D_8, D_9, D_{10}, D_{11}, D_{13}, D_{15}}	2.98
{sport}	{D_1, D_3, D_4, D_6, D_7, D_8, D_{10}, D_{11}, D_{14}, D_{15}, D_{16}}	3.0
{basketball}	{ D_2, D_7, D_8, D_9, D_{10}, D_{12}, D_{13}, D_{14},D_{15}}	2.85
{volleyball}	{D_1, D_2, D_6, D_7, D_{10}, D_{11}, D_{12}, D_{14}, D_{16}}	2.73
{athletics, sport}	{D_1, D_4, D_6, D_8, D_{10}, D_{11}, D_{15}}	1.97
{sport, volleyball}	{D_1, D_6, D_7, D_{10}, D_{11}, D_{16} }	1.72
{ athletics, basketball }	{ D_2, D_8, D_9, D_{10}, D_{11}, D_{15}}	1.72
{ athletics, volleyball }	{D_1, D_2, D_6, D_{10}, D_{11} }	1.34
{ sport, basketball }	{ D_7, D_8, D_{10}, $D_{14}D_{15}$}	1.47
{ basketball, volleyball }	{ D_2, D_7, D_{10}, D_{12}, D_{14} }	1.47
{ athletics, sport, volleyball }	{D_1 D_6, D_{10}, D_{11} }	0.98
{ athletics, basketball, sport }	{ D_8, D_{10}, D_{11}, D_{15}}	0.9

In this step, according to the minimal entropy overlap, we get the cluster described by {athletics, basketball, sport}. Documents D_8, D_{10}, D_{11} and D_{15} belong to this category. They should be moved out from the document database set when the cluster is formed. It can be seen that FTSC algorithm can be used to return the description of the cluster and without the overlap.

The conclusion is that the frequent itemsets have monotonicity: all nonempty subsets must be frequent. According to this property, the FTSC algorithm can be modified to Frequent Term Set-based Hierarchical Clustering algorithm (FTSHC).

4.2 FTSHC Algorithm

FTSC algorithm only uses *k*—itemset and does not use all the frequent feature itemsets. We modify this algorithm and use two levels (*k*—itemset, *k+1*—itemset) to cluster. This new algorithm is called FTSHC.

Frequent feature itemsets is a level structure. An empty feature set including all the documents is the root, frequent *1*—itemset is the first level, frequent *2*—itemset is the second level, and so on. When there is not frequent itemset, the new level is not added by FTSHC.

FTSHC algorithm uses two levels to cluster. The chosen documents are not moved out from the document database set. So, we can get clusters with overlap level.

Figure1 describes the procedure of using FTSHC algorithm to cluster in Table 1. There are only three levels in Figure1, because frequent *4*—itemsets satisfying the given *min_sup* do not exist in 16 documents.

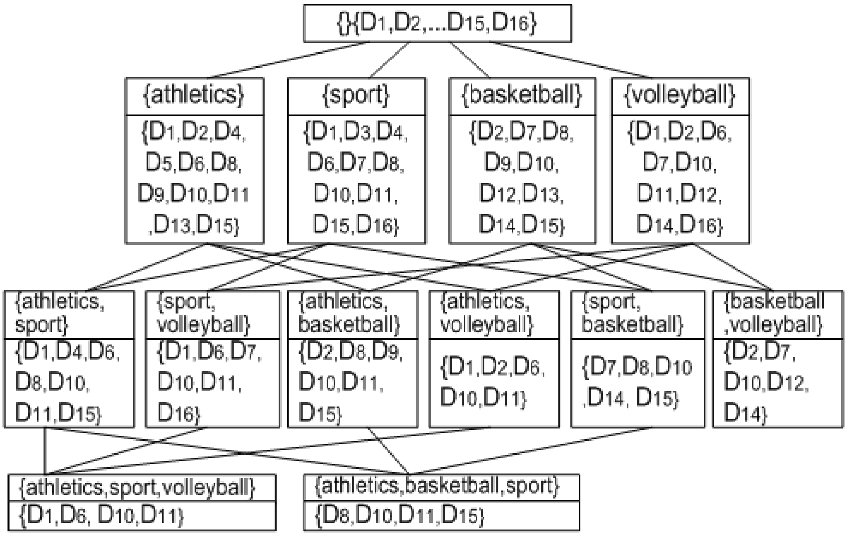


Fig. 1. Procedure of using FTSHC

5 Experimental Analysis

We use NTCIR-3(NII-NACSIS Test Collection for IR System Workshop) document data set to evaluate FTSC algorithm performance qualitatively; we then use NTCIR-3 and Reuters—21578 document data sets to evaluate FTSC, FTSHC algorithms quantitatively. We finally compare the results with that of the K-Means method.

5.1 Qualitative Analysis of FTSC Algorithm

We choose 10500 documents from NTCIR-3 standard testing sets to do the experiment, including sport 1000 pieces, international 500 pieces, politic 500 pieces, art 500 pieces, entertainment 500 pieces, economic 500 pieces, financial 500 pieces, stock 500 pieces, technology 1000 pieces, medical 1000 pieces, education 1000 pieces, service 1000 pieces, navigation 1000 pieces, manufacture 1000 pieces.

Suppose k is a clustering set, $k=\{k_1,...k_k\}$ is the standard category set in the document database. The information entropy of C_j is defined as below:

$$E(C_j) = \sum_{C_j} \frac{n_j}{|D|} \sum_i - p_{ij} \log(p_{ij}) \tag{7}$$

where $|D|$ is the number of documents in the database, n_j is the number of the documents in C_j, p_{ij} is the probability of the document belonging to both C_j and K_j. The value of the information entropy of C_j is in $[0, \log(|K|)]$, which represents the accuracy of clustering category. The smaller the value is, the higher the clustering accuracy is.

Table2 gives the minimal clustering entropy value corresponding to three *min_sup* 0.5, 1.0, and 1.5.

Table 2. Result of clustering the NTCIR-3 document set with FTSC algorithm

min_sup	When the minimal cluster entropy value exists			Final clustering entropy with all documents
	Number of categories	Document recall rate (%)	Minimal entropy	
0.5	14	81.5	0.248	0.381
1.0	10	80.6	0.306	0.403
1.5	5	80.1	0.363	0.486

From the experiment, we know that the number of categories will decrease when the *min_sup* increases. The smaller entropy has big influence on clustering, can provide more information, and is good for the clustering procedure and precision.

5.2 Quantitative Analysis of FTSC and FTSHC Algorithms

Using two standard text data sets NTCIR-3 and Reuters-21578 [6], we evaluate FTSC and FTSHC algorithms by F-Measure method.

Reuters-21578 is often used in text data mining [7][8]. It has 21578 pieces documents and includes 22 files, from reu2-000.sgm to reut2-020.sgm. Each file has 1000 documents. The category is shown in Table 3.

We choose 14 categories, 10500 pieces of documents in NTCIR-3 set and 52 categories, 8654 pieces of documents in Reuters-21578 set to do the experiment.

Precision and recall are two contradictory measure standards. In general condition, the precision will decrease when the recall increases. Both of them cannot be gotten simultaneously. The typical method is F-Measure. It is defined as:

Table 3. Reuters-21578 text category

Category set	Number of categories	Number of categories w/1+ occurrences	Number of categories w/20+ occurrences
EXCHANGES	39	32	7
ORGS	56	32	9
PEOPLE	267	114	15
PLACES	175	147	60
TOPICS	135	120	57

$$F_{\beta}(P,R)=\frac{(\beta^{2}+1)PR}{\beta^{2}P+R} \tag{8}$$

where β is a adjusted parameter. It is used to synthesis the precision and recall. Whenβ=1, the precision and recall are equally handled, at this time F-Measure is also called F_1, and defined as:

$$F_{1}(P,R)=\frac{2PR}{P+R} \tag{9}$$

In the experimental results, we compare K-Means algorithm [9] with FTSC and FTSHC algorithms. We use K-Means algorithm because it is easily programmed and its principal is simple. The complexity of computation is lower. It can process large text data effectively.

Table 4. Result of F1 with K-Mean, FTSC, and FTSHC algorithms

Data Set	FTSC	FTSHC	K-Means
NTCIR-3	0.37	0.34	0.44
Reuters-21578	0.46	0.49	0.57

In Table 4, the clustering results of FTSC and FTSHC algorithms are not much different from that of K-Means algorithm, even though the two algorithms can reduce documents dimension highly. On the other hand, the FTSC and FTSHC algorithms can not only cluster documents but also overlap the clustering results. Compared with the K-Means whose clustering results do not have overlap relations, the FTSC and FTSHC algorithms have remarkable superiority. When we cluster the documents without considering the overlaps of the category, it may cause errors in the following clustering. So the FTSC and FTSHC algorithms are good and practical clustering methods.

6 Conclusions

In this paper, we present FTSC and FTSHC algorithms. These algorithms use Apriori algorithm to get frequent feature terms, reduce the dimensions of the vectors effectively and cluster the documents. The frequent feature terms can descript the clustering results. According to the experiment's results, these algorithms have better clustering performance.

References

1. Lagus, K. and Kaski, S. "Keyword Selection Method for Characterizing Text/Document Maps". Proceeding of ICANN'99.
2. EI-Hamdouchi, A. and Willett, P. "An improved Algorithm for the Calculation of Exact Term Discrimination Values", Information Processing & Management, 24(1):17-22, 1988
3. Agrawal R, Imielimski T, and Swami A. "Mining association rules between sets of items in large databases". In Proc. 1993 ACM-SIGMOD Int. Conf. Management of Data, Washington DC. pp207-216, May 1993
4. Agrawal, R. and Srikant, R. "Fast algorithm for mining association rules in large databases". in Research Report RJ 9839,IBM Almaden Research Center, San Jose, CA, June.1994
5. Agrawal, R. and Srikant, R. "Fast algorithm for mining association rules". Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94), pp487-499, Santiago, Chile, Sep.1994.
6. http://www.daviddlewis.com/resources/testcollections
7. Yang,Y. and Pedersen, J.O. "A comparative study on feature selection in text categorization". Proc. of SIGIR'01, 2001
8. Bekkerman, R. and EI-Yaniv, R. "On feature distributional clustering for text categorization". Proc. Of SIGIR'01,2001
9. Macqueen, J. "Some methods for classification and analysis of multivariate observation". Proc.5th Berkeley Symp. Math. Statist, 1:281-297,1967

An Improvement of Text Association Classification Using Rules Weights

Xiao-Yun Chen [1,2], Yi Chen[2], Rong-Lu Li[2], and Yun-Fa Hu[2]

[1] School of Mathematics and Computer Science,
Fuzhou University, Fuzhou 350002, China
c_xiaoyun@21cn.com
[2] Department of Computer and Information Technology,
Fudan University, Shanghai 200433, China

Abstract. Recently, categorization methods based on association rules have been given much attention. In general, association classification has the higher accuracy and the better performance. However, the classification accuracy drops rapidly when the distribution of feature words in training set is uneven. Therefore, text categorization algorithm Weighted Association Rules Categorization (WARC) is proposed in this paper. In this method, association rules are used to classify training samples and rule intensity is defined according to the number of misclassified training samples. Each strong rule is multiplied by factor less than 1 to reduce its weight while each weak rule is multiplied by factor more than 1 to increase its weight. The result of research shows that this method can remarkably improve the accuracy of association classification algorithms by regulation of rules weights.

Keywords: Data mining; Association classification; Rule intensity; Weight.

1 Introduction

In 1998 Liu, Hsu and Ma proposed the association classification method CBA [1] first. CBA integrates classification process with association rule mining process. CBA gets a better classification result than C4.5 [2] which is a decision tree algorithm also based on rules. After the proposing of CBA, various improvements have been given successively. CMAR [3] and ARC [4] are the typical ones. The basic ideas of these methods are to generate frequent features or frequent feature sets for all classes using existing association rule mining algorithm [5]. These frequent feature sets are used to classify test samples. The more the frequent feature set of a certain class contained in a test sample and the higher the confidence , the greater the probability that the test sample belongs to this class.

Existing text classification methods based on association rules have an obvious deficiency: the classification accuracy is influenced by the distribution of training samples. When the distribution of classification rules is uneven, classifier will prefer the class with more rules. In practice, distribution of frequency of features is often uneven among different classes. If the minimum support is set too high, it may be difficult to find sufficient rules for infrequent classes with low frequency of features.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 355 – 363, 2005.

If the minimum support is set too low, too many useless or over-fit rules will be generated for the frequent classes with high frequency of features. In order to solve this problem, this paper proposes a text categorization method (WARC) based on association rules whose weight can be adjusted. The preference of the classifier to the frequent classes can be reduced by giving different weights to rules pointing to different categories.

2 Disadvantage of Association Classification ABC-BC

2.1 Basic Ideas of Association Classification ARC-BC

Assume that training document set *D* has m classes $y_1, y_2,..., y_m$, where y_i includes n_i texts and text x_j is expressed as a *n*-dimensional vector $x_j=\{w_{j1},w_{j2}..., w_{jn}\}$. Also assume that w_{jj}=1 when the jth feature word appears in text x_j. Otherwise w_{jj}=0.

ARC-BC is a local association classification algorithm which looks for frequent patterns in each class in stead of in whole training set. The frequent pattern is used for condition part and the class label is used for consequent part to form a classification association rule.

The algorithm mainly includes two phases:

(1) Generating classification association rules.

Regard training samples with the same class label y_i as a subset. Then find the frequent item set *T* whose support is larger than given minimum support threshold, and generate classification association form as $T \Rightarrow y_i\, (\sigma = n\%)$.

Support of rule $T \Rightarrow y_i$ is: $\varphi(T \Rightarrow y_i) = \dfrac{|\{j \mid x_j \in y_i \wedge T \in x_j\}|}{|\{j \mid x_j \in y_i\}|}$;

Confidence of rule $T \Rightarrow y_i$ is: $\sigma(T \Rightarrow y_i) = \dfrac{|\{j \mid x_j \in y_i \wedge T \in x_j\}|}{|\{j \mid T \in x_j\}|}$.

(2) Classification of testing samples.

The probability that testing sample d belongs to class y_i is sum of confidences of rules whose condition part appear in d and consequent part is y_i. This sum is defined as y_i class confidence for d:

$$\Omega(d, y_i) = \sum_{r \in R} \sigma(r)$$

Where *R* is the set of matching rules whose condition part belong to d.

During classification, group matching rules for d by class labels so that rules in the same group have the same class label. Calculate sums of confidence for all groups of rules, namely all class confidence for d. Then sort all class confidence Ω by descending and d is assign to the class with the highest Ω.

2.2 Existent Problems

Association classification method ARC-BC uses all association rules to form classification association rule set, so accuracy of classification is highly influenced by the distribution of rules. Desirable accuracy can be achieved if the distribution of rules in different classes is even. However, when the distribution is uneven, classifier will prefer those classes with more rules. The number of association rules highly depends on distribution of feature words in the training sample set, so classifier prefers classes with more rules, namely the classes in which feature words appear highly frequently. The rules in such class are too strong. Other classes in which feature words are less and appear less frequently have fewer rules and are weak.

In Fig.1 every point represents a classification rule, such as government⇒A, going abroad⇒B, studying abroad⇒B, going abroad∧studying abroad⇒B, exam⇒C and so on. Fig.1 shows that class A has fewer rules and confidences of rules are lower, so it is weaker. Class B has more rules and confidences of rules are higher, so it is stronger. Assume that testing sample d matches with all three rules of class A while only matches with one rule of class B. Because the confidence sum of rules matching with d in class A is less than the corresponding sum in class B, classifier probably will classify sample d which belongs to class A into class B falsely.

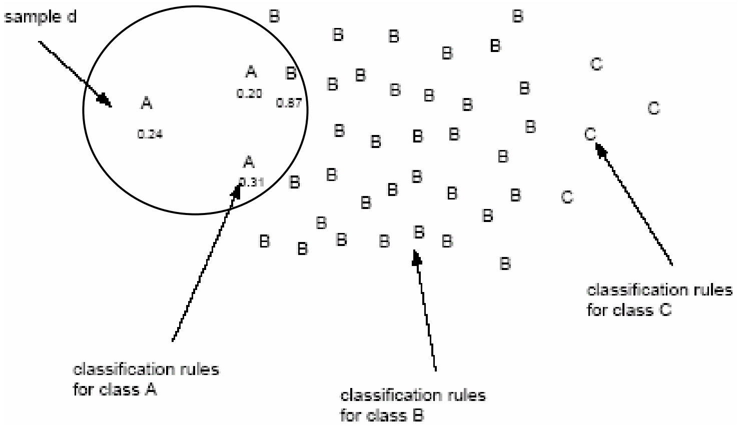

sample d
A
0.24
A
0.20
B
0.87
A
0.31
B
C
classification rules for class A
classification rules for class B
classification rules for class C

Fig. 1. Classification Rules Matching With Sample d

Due to the existence of strong rules, samples belonging to other classes are often classified to class to which strong rules belong. How to solve the uneven distribution of strong and weak rules? Straight idea is to give strong rules low weights so that the disturbance of these rules giving to other classes can be decreased. Meanwhile, the wear rules are given high weights so that their classification ability can be increased.

3 Weight Adjustment for Classification Rule

Current problem is how to decide weights of rules. The number of training samples of each class which are classified falsely is used to describe the attribute whether rules are strong or weak and to adjust weights of rules in order to improve the classification accuracy.

3.1 Rules Intensity

Rule set for class y_i forming as $T \Rightarrow y_i$ may classify samples falsely in following two aspects: (1) Classifying sample which does not belong to class y_i into class y_i; (2) Classifying sample which belongs to class y_i into other classes.

So the intensity of rules of class y_i can be described as following qualitative criterion:

(1) If the number of samples classified into class y_i falsely is large and the number of samples belonging to class y_i which are classified into other classes falsely is small, then the rules for y_i is the strongest.

(2) If the number of samples classified into class y_i falsely is large and the number of samples belonging to class y_i which are classified into other classes falsely is also large, then the rules for y_i is strong.

(3) If the number of samples classified into class y_i falsely is small and the number of samples belonging to class y_i which are classified into other classes falsely is also small, then the rules for y_i is moderate.

(4) If the number of samples classified into class y_i falsely is small and the number of samples belonging to class y_i which are classified into other classes falsely is large, then the rules for y_i is the weakest.

Weights should be adjusted according to criterions above. Strong rules should be given low weights in order to decrease their influence to other classes. Weak rules should be given high weights in order to increase their influence to other classes. The ultimate purpose is to make all rules to be neither strong nor weak and reduce the influence of uneven distribution.

Definition of strong rule is given according to criterions above. For convenience sake, classification using rules is described by assumption as $h_t : X \times Y \rightarrow [0,1]$. $h_t(x_i, y_i) = 1$ or $h_t(x_i) = y_i$ indicates that sample x_i is classified into class y_i.

Definition 1. *Rule Intensity*
The intensity for rule set $T \Rightarrow y_i$ of class y_i is defined as:

$$\rho_t(y_i) = \varepsilon_1^t(y_i) + \varepsilon_2^t(y_i) \tag{1}$$

Where $\varepsilon_1^t = \dfrac{|\{j \mid h(x_j) = y_i, x_j \notin y_i\}|}{|\{j \mid x_j \notin y_i\}|}$, $\varepsilon_2^t = \dfrac{|\{j \mid h(x_j) = y_i, x_j \in y_i\}|}{|\{j \mid x_j \in y_i\}|}$

ε_1 is the ratio of number of samples which are classified into class y_i falsely to number of all samples which belong to no class y_i. ε_2 is the ratio of number of samples which are classified into class y_i correctly to number of samples which belong to class y_i. Obviously ρ is higher when rules are stronger. When all samples which do not belong to class y_i are classified into class y_i falsely and all samples which belong to class y_i are classified into class y_i correctly ($\varepsilon_1 = 1$, $\varepsilon_2 = 1$, $\rho = 2$), the rules of class y_i is the strongest. Otherwise, when no sample which does not belong to class y_i is classified into class y_i falsely and all samples which belong to class y_i are classified into other classes falsely ($\varepsilon_1 = 0$, $\varepsilon_2 = 0$, $\rho = 0$), the rules of class y_i is the weakest.

Definition 2. *Rule weight vector*

It is defined as $\vec{W}_t = [w_1^t, w_2^t, ..., w_k^t]$ and $\sum_{i=1}^{k}(w_i^t \times n_i) = n$. $\vec{W}_t$ is the weight before the *t*th adjustment of rules. w_i^t is the weight of rules of class y_i while n_i is the number of rules whose consequent part is class y_i, n is the number of all rules.

Give initial value for $\vec{W}_t$: $\vec{W}_1 = [1,1,...,1]$, i.e. $w_i^1 = 1$ (*i=1,2,...k*) , so that rules has the same weight 1.

Definition 3. *Weight adjustment factor*
According to Definition 1, weight adjustment factor of every class is given :

$$\alpha_t(y_i) = 2 - \rho_t(y_i) \tag{2}$$

New weight adjustment factor can be given by multiplying $\alpha_t(y_i)$ by w_i^t , namely $w_i^{t+1} = w_i^t \times \alpha_t(y_i)$. According to Definition 3, when rule is weak ($0 \leqslant \rho_t < 1$), weight adjustment factor $\alpha_t > 1$, so new weight which is given by multiplying original weight by weight adjustment factor will increase. When rule is moderate ($\rho_t = 1$), weight adjustment factor $\alpha_t = 1$, so new weight which is given by multiplying original weight by weight adjustment factor will remain the same. When rule is strong ($1 < \rho_t \leqslant 2$), weight adjustment factor $0 < \alpha_t < 1$, so new weight which is given by multiplying original weight by weight adjustment factor will decrease.

3.2 Weight Adjustment

The training phase of association classification method based on weight adjustment is an iterative process. Firstly, association rule mining algorithm is used to generate classification association rules. These rules are used to do classification test on training samples for the first time. Intensities of rules of different classes, corresponding weight adjustment factors and new weights of rules are calculated according to classification result. New confidence of rule is the product of confidence and new weight. If rules of some classes are too strong, their confidences are multiplied by weight adjustment factor which is less than 1 in order to decrease the confidence. Otherwise, the confidence is increased. Adjusted classification rules are used to classify training samples again and rules are adjusted according to the classification result. Repeat such adjustment till one of following conditions is met:

(1) no too strong or too weak rules exist, namely, intensities of rules incline to be equal (2) satisfying classification accuracy is reached (3) given maximum iterative time is reached.

In detail, weight of rule in the t+1 time iterative step is calculated according to following expression.

$$\vec{W}_{t+1} = [w_1^t \times \alpha^t(y_1), w_2^t \times \alpha^t(y_2), ..., w_k^t \times \alpha^t(y_k)] \tag{3}$$

Namely $$w_i^{t+1} = w_i^t \times \alpha_t(y_i) \tag{4}$$

During weight adjustment, in order to make sure new weight $\vec{W}_{t+1}$ satisfies $\sum_{i=1}^{k}(w_i^{t+1} \times n_i) = n$, expression above is normalized:

$$w_i^{t+1} = \frac{w_i^t \times \boldsymbol{\alpha}_t(y_i) \times n}{Z^t} \tag{5}$$

Where $Z^t = \sum_{i=1}^{k} w_i^t \times \alpha_t(y_i) \times n_i$.

4 Analysis of Experimental Results

In order to test the classification method proposed in this paper, 2000 news web pages are downloaded as test data from http://www.xinhuanet.com/. Hyperlinks and advertisements in these web pages are deleted so that only news texts remain. This data set is not easy to be classified and feature distribution is extremely uneven. Therefore, association classification algorithm ARC-BC on this data set has low accuracies. In table 1, values in columns y_1~y_6 are rule intensities and weights generated by ARC-BC when t=1. Obviously rules intensity of class y_1 is the weakest and is only 0.31 while rules intensity of class y_4 is the strongest to be 1.18. After rule weight adjustment, rules intensities ρ_t in Table 1 become close each other. Comparing ρ_t in rows where t=1 and t=2, we can find that: differences of ρ_t decrease evidently only through weights adjusting once. Classification accuracy for training samples increases from 63.3% to 83.3%. The improvement is obvious.

Table 1. Rule intensity and weights of training sample after each iterativeing

	y_1		y_2		y_3		y_4		y_5		y_6		accuracy
	ρ_t	W_{t+1}	ρ_t	W_{t+1}	ρ_t	W_{t+1}	ρ_t	W_{t+1}	ρ_t	W_{t+1}	ρ_t	W_{t+1}	
t=1	**0.314**	1.7	**0.82**	1.18	**0.96**	1.04	**1.18**	0.82	**0.4**	1.6	**0.4**	1.6	0.633
t=2	**0.72**	1.59	**0.94**	0.88	**0.93**	0.79	**0.9**	0.72	**0.83**	1.35	**0.9**	1.28	0.833
t=3	**0.72**	1.8	**0.86**	0.89	**0.94**	0.74	**0.92**	0.69	**0.86**	1.37	**0.9**	1.25	0.833
t=4	**0.72**	2	**0.86**	0.88	**0.84**	0.75	**0.92**	0.65	**0.86**	1.4	**0.9**	1.2	0.817
t=5	**0.72**	2.2	**0.98**	0.87	**0.98**	0.75	**0.7**	0.6	**0.84**	1.3	**0.9**	1.1	0.817

After five iterative steps, intensities of rules can not be changed obviously by weight adjustment, and the classification accuracy for training sample set (closed testing) also becomes stable using adjusted rules (see Fig. 2). The algorithm ends at this time.

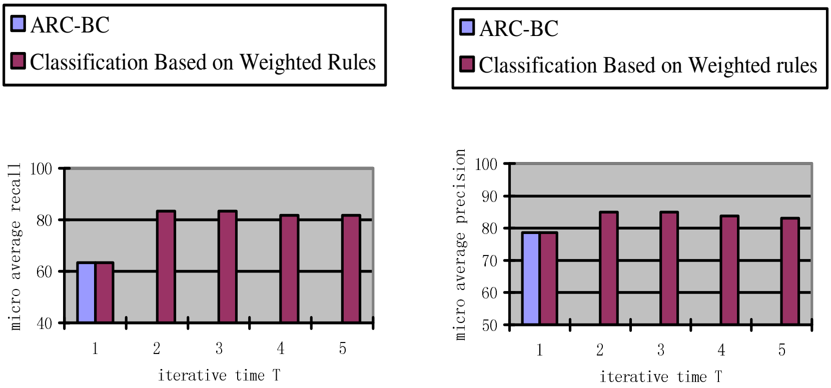


Fig. 2. Micro-Average-Recall and Micro-Average-Precision on the Closed Testing

Fig. 2 shows that: recall and precision of closed testing increase obviously after rule weight adjustment. Micro average recall increases from initial 63% to 83% and micro average precision increases from initial 78% to 85%.

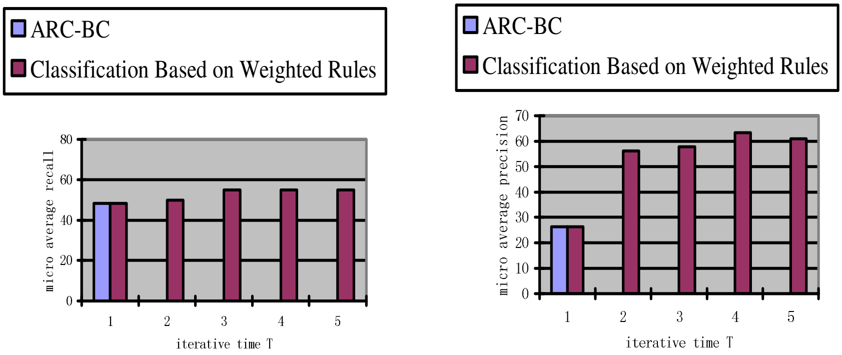


Fig. 3. Micro-Average-Recall and Micro-Average-Precision on the Open Testing

Fig. 3 shows the classification results on testing samples set (open testing) using each rule set generated in each iterative step. Obviously the classification accuracy after rule weight adjustment for open testing is better than ARC-BC. Micro average recall increases from 48% to 55% and micro average precision increases from initial 26% to 61%.

5 Conclusion and Further Research

Stability of association classification has long been concern of researchers. Association classification performs well on many famous training sets; especially result of text classification using ARC-BC on text data set Reuters-21578 is particularly inspiring. However, the sample sets used are evenly distributed and are all English corpuses. In order to test the classification capability and stability of this algorithm on Chinese corpuses, Chinese web pages are downloaded from XinHua Website and so on. This data set is preprocessed for test. Result of the experiment shows that ARC-BC can not perform well and classification accuracy is very low. The main reason for the lower accuracy of classification is the extreme uneven distribution of feature words frequencies in this sample set. The uneven feature word distribution leads to uneven rule distribution in different classes. Therefore, rule intensity is proposed in this paper. During the iterative training, rules weights are adjusted using rule intensities and corresponding adjustment factors in order to decrease the effect made by uneven distribution of rules in various classes. Experimental result shows that this method improves obviously classification accuracy of ARC-BC.

Performance of classifier is related to classification method as well as method of text feature extraction and text feature selection. How to select proper feature is always a disturbing problem. Though many researches aim at contrast of existing methods, these works are based on one or some particular test data sets. No method performs well on various data sets [8][9]. Besides, during the selection of text feature, many parameters are set by human experiences. Especially the number of necessary feature words is decided by experiences. A too small number will make feature set unable to cover training samples while a large one will decrease classification efficiency and increase system expense. Our further work is to find a self-adapting feature selection method by researching the text feature selection. This new method should use no or few parameter configuration to generate proper feature set.

Acknowledgments

This work has been supported by the National Nature Science Foundation of China under Grant No.60473070, the Key Project Foundation of Fujian Province under Grant No.K04005 and JB02069, and the Science and Technology Development Foundation of Fuzhou University under Grant No.2004-XQ-17.

References

1. B. Liu, W. Hsu, and Y. Ma. Integrating classification and association rule mining. In ACM Int. Conf. on Knowledge Discovery and Data Mining (SIGKDD'98), pages 80-86, NewYork City, NY, August 1998.
2. J.R. Quinlan. C4.5: programs for machine learning. San Mateo, CA:Morgan Kaufmann. 1993.

3. W. Li, J. Han, and J. pei. CMAR:accurate and efficient classification based on multiple classification rules. In San Jose, California, November 29-December 2001.
4. O.R. Zaïane and M.L. Antonie. Classifying text documents by associating terms with text categories. In Proceeding of the Thirteenth Australasian Database Conference (ADC'02), pages 215-222, Melbourne, Australia, January 2002.
5. R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In: Proceeding of the 1994 International Conference on Vary Large Data Bases, Santiago.chile, 1994. pages 487-499.
6. S.G. Zhou, J.H. Guan, Y.F. Hu, and A.Y. Zhou. A Chinese text classification algorithm without lexicon and segmentation. In Computer Research and Development, 2001, No.7, Vol.38 (In Chinese)
7. Yoav Freund and Robert E.Schapire, Experiments with a New Boosting Algorithm. In Machine Learning: Proceedings of the Thirteenth International Conference, pages 148-157, 1996.
8. Y. Yang and X. Lin, A Re-Examination of Text Categorization Methods. In Proceedings of SIGIR 99, 1999
9. Y. Yang and Jan P.Pedersen, A comparative study on feature selection in text categorization, in Proceedings of the Fourteenth International Conference on Machine Learning (ICML'97), Jr.Doughals H.Fisher, Ed.,Nashville, TN, July 8-12 1997
10. T.M. Mitchell. Machine Learning, McGraw Hill, New York, US, 1996.
11. J. Han. J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In
12. SIG MOD '00, Dallas, TX, May 2000.
13. D. Koller and M. Sahami. Hierarchically classifying documents using very few words. International Conference on Machine Learning, 1997, 170-178.
14. X.Y. Chen, Y. Chen, L. Wang, and Y.F. Hu. Text Categorization Based on Association Rules with Term Frequency. In Proceeding of the 3th International Conference on Machine Learning and Cybernetics. Shanghai, China, August26-29, 2004, 1610-1615.

Word Segmentation and POS Tagging for Chinese Keyphrase Extraction

Xiaochun Huang, Jian Chen, Puliu Yan, and Xin Luo

School of Electronic Information, Wuhan University,
Wuhan, 430079, Hubei, China
xiaochun.huang@gmail.com
jchen_2000@163.com
ypl@whu.edu.cn

Abstract. Keyphrases are essential for many text mining applications. In order to automatically extracting keyphrases from Chinese text, an extraction system is proposed in this paper. To access a particular problem of Chinese information processing, a lexicon-based word segmentation approach is presented. For this purpose, a verb lexicon, a functional word lexicon and a stop word lexicon are constructed. A predefined keyphrase lexicon is applied to improve the performance of extraction. The approach uses a small Part-Of-Speech(POS) tagset to index phrases simply according to these lexicons. It is especially effective for identifying phrases in form of combinations of nouns, adjectives and verbs. Keyphrases are sifted by their weighted TF-IDF (Term occurrence Frequency-Inverse Document Frequency) values. New keyphrases are added into the keyphrase lexicon.

1 Introduction

Keyphrases are usually called as keywords, but in fact it is not exactly the same. Most keyphrases consist of one more words that express more specific and more precise information than singular words can. Taking keyphrases to represent full text content is an effective approach to improve the performance of text related applications [1]. Keyphrases can describe the prime information of documents very concisely, thus can be applied as metadata in automatic summarization, indexes of electronic library, keywords in search engine and features for text categorization [2]. Moreover, they are especially practical to be displayed as short document surrogates on small screen devices, such as PDA [3].

Our research focuses on keyphrase extraction from Chinese computer science technical papers. There are no explicit separators except punctuation marks among Chinese words, thus word segmentation become necessary to make Chinese text readable for computer. Many word segmentation and Part-Of-Speech (POS) tagging methods have been proposed to identify Chinese words or phrases in near decades. Lexicon-based approaches like what [4] applys compare text fragments with lexicon entries, find all possible words, then select the schema that conforms to sementic rules best as segmentation result. [5] is a non-lexicon based appraoch, which groups

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 364–369, 2005.

character pairs with high value of mutual information into words. Both the two kinds of approaches tend to accurately recognize every word at the cost of system resources. However, it is not necessary for many specific applications. In the case of extracting keyphrases from scientific papers, keyphrases are usually composed by continuous words that may be nouns, adjectives and verbs [6] respectively, which means that other classes of words are useless in our application. Nouns and adjectives take up above 50 percent of Chinese words [7], and new nouns and adjectives appear so fast that no lexicon can ever be completed. Considering that, we present a simple word segmentation approach without lexicons for these two classes of words.

The paper proposes a unique tiny POS tagse in Section 2.1, and introduces four lexicons needed for segmentation in Section 2.2 and the simple word segmentation approach in Seciton 2.3. The Process of keyphrase extraction is in Section 3. Section 4 discusses the experimental results of our system on a corpus of computer science technical papers. We summarize our work and findings at the end of the paper.

2 Word Segmentation and POS Tagging

2.1 Tagset

Our segmentation method aims to divide sentences correctly into phrases using minimal system resources. Hockenmaier and Brew point out in [8] that it is worth trying to use a tagset encoding a small number of carefully chosen major-class distinctions. The base tagset used consists of only six POS tags: *n* for noun, *a* for adjective, *v* for verb, *f* for functional word, *s* for stop word and *x* for uncertain word or phrase. In our system, *x* is generally a noun, an adjective or their combinations.

2.2 Lexicons

In our system, four lexicons are used: a verb lexicon, a functional word lexicon, a stop word lexicon and a keyphrase lexicon.

Verbs are always the cores of sentences as well as of keyphrases. When segmentinh, any word that can be found in the verb lexicon will be treated as a verb by default, even though it is a noun or a word of any other class grammatically. For example, 数据库(database) is assigned as 数/v 据/v 库/x, although 数据(data) is a noun here，it is considered as two verbs because 数(count) and 据(refuse) appear in the verb lexicon individually. However, there would be no loss of word sense, because the keyphrase extraction algorithm in Section 3 concatenates adjacent words that are assigned as *v* and *x* and treat them as a whole phrase.

Since Chinese sentences are generally organized by functional words like 和(and), it is very important to make full use of functional words. We build a functional word lexicon based on [9]. Ancient functional words are not contained in our lexicon because they are extremely rare in computer science papers. Chinese functional words includes adverb, preposition, conjunction, auxiliary word and particle [10].

Stop words are words that are too common or meaningless for specific applications. In our stop word lexicon, we cover three kinds of stop words: numeral, quantifier, pronoun, localizer, onomatopoeia and fixed exclamation etc.; frequent but meaningless verbs; special characters, such as Roman numerals. Functional words are treated as stop words after segmenting but not indexed in the lexicon, because they are already in the functional word lexicon. Whether a word in the lexicon should be cleaned up in the process of keyphrase extraction depends on a set of rules (i.e. if one or more non-Chinese characters exist in a sequence of Arabic numerals, length of which is less than 20, we think the sequence may be a keyphrase and will not treat it as a stop word. It is a very common phenomenon in papers, such as C4.5).

Features of the lexicons include: length of words, internal state number, frequency, pinyin (the official phonetic spelling for chinese characters) and compound words. The last feature lists words of other classes containing the functional word. For instance, the noun 把手(handle) is a compound word of functional word 把(hold). Entries in these lexicons are ordered by the pinyin of their first characters; entries started with the same character are indexed by their length.

A keyphrase lexicon of computer science is constructed, which is an electronic extension of [11]. For entries in it, besides the features listed above, we add frequency and the number of documents including the keyphrase or its synonyms.

2.3 Lexicon-Based Word Segmentation and POS Tagging

In our segmentation algorithm, long sentences are cut into short fragments by punctuation marks first, fragments then are segmented into possible phrases and tagged with *v*, *f*, *s* and *x* according to the three lexicons mentioned in Section 2.2 respectively, and finally a synthetic result is given from the tagged phrases according to predefined rules. Since we use neither a noun lexicon nor an adjective lexicon, it is much likely to make mistakes when segmenting. To avoid it, we predefine a set of rules to save word senses as much as possible.

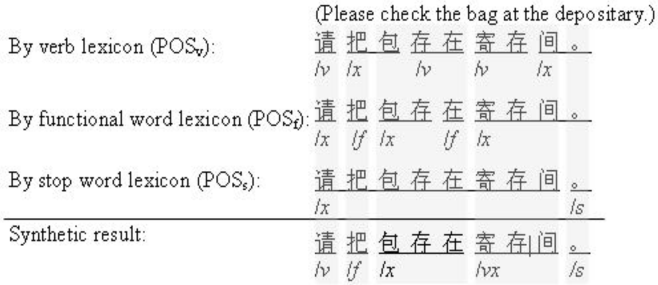


Fig. 1. Examples of segmentation with ambiguity

The following four rules are the basic part of the rules, where A, B and C are undetermined segments; $p, q \in \{v, f, s\}$ denotes the verb, functional word or stop word lexicon; $\mathrm{POS}_p(\mathrm{A}) \in \{0, 1\}$ is the tagged result of A by p, 0 for unknown phrase, 1 for true result; POS(A) is the finally assigned POS of A; NEIGHBOR(A,B) and

COMPOUND(B, A) denote respectively whether B is a neighbor of A and whether A is a compound word of B, 1 for true, 0 for false.

1) NEIGHBOR(A,B)=true, POS_p(A)=POS_p(B) → POS_p(AB)=POS_p(A)=POS_p(B). For example, 打印(print)/*v* 输出(output)/*v* can be combined as 打印|输出/*v*.
2) XOR(POS_p(A), POS_q(A))=1, POS_p(A)=1 → POS(A)=*p*. Take 请(please) in Fig. 1 as an example, POS_f(请)=POS_s(请)=0, POS_v(请)=1, then POS(请)=*v*, that is, the tagged result is 请/*v*.
3) XOR(POS_p(A), POS_q(A))=0 → POS(A)=*x*. If a fragment exists in different lexicons, we consider its POS is indeterminable and tag it as *x*.
4) NEIGHBOR(A,C)=1, COMPOUND(B,A)=1, B∈C, POS_p(A)=1, POS_q(C)=1 → POS(AC)=*x*.在(at) in Fig. 1 assigned as a part of verb 存在(exist) (that is, 存在 is a compound of 在) by the verb lexicon, while assigned as *f* by the founctional word lexicon, so the result is *x*.

3 Keyphrase Extractions

After word segmentation, we get a sequence of all possible phrases tagged by v, f, s, x and their combinations. After clean up meaningless words (most stop words and functional words), consolidate synonyms, and delete phrases with less than two characters and phrases that occur only once in the document, we get a set of candidate keyphrases which are in form of v, x, vx, xv, xvx or vxv etc. The steps of sifting keyphrase from the candidates are:

```
Input: candidate keyphrase set $CK$ of the document $D$, keyphrase lexicon $KL$
Ouput: keyphrase set $DK$ of $D$, added keyphrase lexicon $KL$
  1)  $DK$=null;
  2)  while($CK$ != null): $weight(i) = (\sum n_{ij} w_j) * log(N / n(i)) * len(i)$        (1)
  3)  if $weight(i) > \mu$ then $DK = DK \cup \{i\}$
          if $i$ is not in $KL$ then $KL = KL \cup \{i\}$
      else if $i$ is with POS $x$ or combination of $x$ and $v$
           then $i = subphrase(i, k)$) and go to 2) until $k<2$.
```

Here, n_{ij} denotes the occurrence of *i* at location *j* (i.e title, abstract, headings and body); *n*(*i*) is the number of documents including *i*; w_j is the given weight for phrases at *j*; μ is a threshold; *len*(*i*) is the corresponding value according to the number of characters *i* has. According to [12], longer phrases are more likely to be keyphrases. The last three are application-specified. *subphrase*(*i*, *k*) is a fragment including the first character to the *k*th character of *i*.

4 Experiments

To evaluate the performance of our keyphrase extraction approach, we used the Corpus for Text Categorization [13] as our training and testing dataset, which are provided by the NLP group of Shanghai International Database Research Center, Fudan University. It includes a category of computer science that has 2,715 papers totally. Keyphrases have been selected from 10 papers of the category with different parameters. The results are list in Table 1.

Table 1. Average values of keyphrases from 10 computer science papers

	μ=0.3		μ=0.4		μ=0.5		μ=0.6		μ=0.7	
	n_{avg}	a	n_{avg}	a	n_{avg}	a	n_{avg}	a	n_{avg}	a
N=50	10.4	0.385	9	0.389	8.2	0.439	6.4	0.390	6	0.4
N=100	10.5	0.314	8.4	0.321	6.2	0.411	4.7	0.404	3.4	0.426
N=500	9.9	0.391	8.7	0.401	7.1	0.424	5.5	0.430	4.2	0.451
N=1000	10.8	0.422	8.2	0.451	6.9	0.502	5.2	0.493	4.1	0.480

N - the number of papers in the corpus; n_{avg} - average number of automatically selected keyphrases for each paper; a - accuracy of automatically selected keyphrases (the ratio of keyphrases correctly chosen by our system to keyphrases output totally).

Keyphrases that are equal to ones provided by authors are taken as correct keyphrases. The results in Table 1 show that with the increasing of μ and N, the system selected fewer keyphrases with higher accuracies. Thus, we can improve the performance of our system by adding large amounts of papers. However, the accuracy is relatively low. We noted that most authors choose not only the most frequent phrases but also their superordinate concepts as keyphrases. For example, in [14] the author uses *computer vision* as a keyphrase, which is the superordinate concept of another keyphrase *object tracing* that doesn't appear in the paper at all. Take concept network into account will improve the accuracy of our experiment greatly.

Table 2. Numbers of keyphrases with different POS combinations (100 for training and 10 for testing)

	n_v	n_x	$n_{vx}+n_{xv}$	n_{other}
μ=0.3	19	178	102	754
μ=0.4	12	154	113	572
μ=0.5	4	122	51	442
μ=0.6	0	98	40	328
μ=0.7	0	64	31	242

The content of Table 2 is the statistical results of keyphrases selected automatically at different μ, from which we can see singular verbs, nouns or adjectives are seldom keyphrases, however most keyphrases are compounded by them. So it was coincident with the actual facts not to identify out all the words in sentences one by one.

5 Conclusions

Our research starts up for just a short time. The word segmentation approach in the paper is particularly useful in keyphrase extraction. It is cheap and easy, but in pursuit of neither correct part-of-speech generation nor precise understanding of Chinese papers. The output of our system can be the preliminary information of other applications, but some remedies for precise segmentation should be made to satisfy different requirements, and a necessary tradeoff must be considered between concision and accuracy. In order to improve the performance of our system, we are trying to use simple syntax rules and semantic rules for segmenting, to train the keyphrase lexicon with large corpus, and to introduce concept network of computer science into the extraction algorithm.

References

1. P. Turney: Learning to Extract Keyphrases from Text. National Research Council of Canada (1999)
2. E. D'Avanzo, A. Lavelli, B. Magnini and R.Zanoli: Using Keyphrases as Features for Text Categorization. ITC-irst, Technical report, November, 12 (2003) Ref. No.: T03-11-01
3. Steve Jones, Matt Jones, Shaleen and Deo_andA2: Using Keyphrases as Search Result Surrogates on Small Screen Devices. Personal and Ubiquitous Computing, Springer-Verlag, Vol. 8, Issue 1 (2004) 55 - 68
4. ICTCLAS (Institute of Computing Technology, Chinese Lexical Analysis System). http://www.nlp.org.cn/project/project.php?proj_id=6
5. Sun Maosong, Shen Dayang and Benjamin K. Tsou: Chinese Word Segmentation without Using Lexicon and Hand-Crafted Training Data. Proceedings of the 17th international conference on Computational linguistics, Vol. 2, (1998) 1265 - 1271
6. Dou Shen, Yan Cong, Jiantao Sun and Yuchang Lu: Studies on Chinese Web Page Classification. Proceedings of the Second International Conference on Machine Learning and Cybernetics, Xi'an, China, Vol. 1, (2003) 23 - 27
7. Shiwen Yu, Xuefeng Zhu and Yunyun Zhang: The Specification of the Synthetic Knowledge-base of Contemporary Chinese. Journal of Chinese Information Processing, Vol. 10, (1996) 1-22
8. Hockenmaier, J. and Brew, C: Error-driven learning of Chinese word segmentation. In 12th Pacific Conference on Language and Information, Singapore. Chinese and Oriental Languages Processing Society (1998) 218-229
9. Xiaoqi Li: A Guide to Functional Words in Modern Chinese. Peking: Peking University Press (2003)
10. Teching Material of Modern Chinese. Peking: Peking University. http://ccl.pku.edu.cn/course/5_jiaocai.Asp?Folder=%2Fcourse%2Flecturenotes&MyOrderBy=标题
11. Nengqin Wang and Qi Liu: Chinese Thesaurus of Computer Science and Technology. Peking: Tsinghua University Press (1990)
12. K. S. Han, Y. C. Wang, Z. Shen and F. F. Wu: Extract Subject from Chinese Text with Three Different Levels. Journal of Chinese Information Processing, Vol. 15, No.4, (2000) 20-27
13. http://www.nlp.org.cn/docs/doclist.php?cat_id=16&type=15
14. Xiaokun Li: Tracing the Position of Aim Point Based on 2D Modes. Computer Engineering, Vol.25, No.5 (1999)

Learning User Profiles from Text in e-Commerce

M. Degemmis, P. Lops, S. Ferilli, N. Di Mauro, T.M.A. Basile, and G. Semeraro

Dipartimento di Informatica, Università di Bari,
Via E. Orabona, 4 - 70125 Bari - Italia
{degemmis, lops, ferilli, ndm, basile, semeraro}@di.uniba.it

Abstract. Exploring digital collections to find information relevant to a user's interests is a challenging task. Algorithms designed to solve this *relevant information problem* base their relevance computations on *user profiles* in which representations of the users' interests are maintained. This paper presents a new method, based on the classical Rocchio algorithm for text categorization, able to discover user preferences from the analysis of textual descriptions of items in online catalogues of e-commerce Web sites. Experiments have been carried out on a dataset of real users, and results have been compared with those obtained using an Inductive Logic Programming (ILP) approach and a probabilistic one.

1 Introduction

E-commerce sites often recommend products they believe a customer is interested in buying. Often users are swamped with (product) information and have difficulty in separating relevant from irrelevant information. Many Web sites have started to embody recommender systems as a way of personalizing their content for users. Recommendation algorithms use input about a customer's interests to generate a personalized list of recommended items. A possible way to achieve personalization is to use static profiles that must be manually updated by users when their interests change. These limitations clearly call for alternative methods that infer preference information implicitly and support automated content recommendation. Machine Learning techniques are being used to recognize regularities in the behavior of customers and to infer a model of their interests, referred to as user profile.

The paper presents a new method, based on the classical Rocchio algorithm for text categorization [11], able to discover user preferences from the analysis of textual descriptions of items in the catalogue of an e-commerce Web site. The novelty of the method can be summarized as follows:

a) positive and negative examples are weighted differently for each user, according to the rates given during the training phase. The classical Rocchio method uses specific control parameters that allow setting the relative importance of positive and negative examples;

b) the method is able to manage documents structured in different slots, each corresponding to a specific feature of an item, for example title, authors,

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 370–381, 2005.

abstract. This strategy permits to give a different weight to words on the basis of the slot in which they appear.

In order to evaluate the effectiveness of the proposed approach, a comparison with different learning strategies has been carried out, namely an ILP approach and a naïve bayes method. Our experiments evaluated the effects of the above mentioned methods in learning intelligible profiles of users' interests. The experiments were conducted in the context of a content-based profiling system for virtual bookshop on the World Wide Web. In this scenario, a client side utility has been developed in order to download documents (book descriptions) for a user from the Web and to capture users feedback regarding his liking/disliking on the downloaded documents. Then, this knowledge can be exploited by the three different learning techniques so that when a trained system encounters a new document it can intelligently infer whether this new document will be liked by the user or not.

The paper is organized as follows: Section 2 describes the main principles for learning user profiles from textual description. Section 3 describes in more details the new Rocchio-based algorithm for inferring user profiles and gives an overview of the systems used for comparison. Section 4 presents the detailed description of the experiments. Finally, some conclusions are drawn in Section 5.

2 Learning User Profiles from Textual Descriptions

Recent research on intelligent information access and recommender systems has focused on the content-based information recommendation paradigm that exploits textual descriptions of the items to be recommended and relevance ratings given by users to infer a profile of user interests [7]. Text categorization is commonly described as follows: given a set of classes $C = \{c_1, \ldots, c_n\}$ and a set of training documents labelled with the class the document belongs to, the problem consists in building a classifier able to assign to a new document the proper class. We consider the problem of learning user profiles as a binary classification task: each document has to be classified as interesting or not with respect to the user preferences. The set of classes is restricted to c_+, the positive class (user-likes), and c_-, the negative one (user-dislikes). The application of text categorization methods to the problem of learning user profiles is not new: several experiments have shown that the naïve Bayesian classifier offers several advantages over other learning algorithms [8, 10]. Thus, we compared the proposed Rocchio-based algorithm with the naïve Bayesian classifier implemented in our Item Recommender system. Moreover, our research aims at comparing these techniques with a symbolic approach able to induce profiles that are more readable from a human understandability viewpoint.

2.1 Documents Representation

The representation that dominates the text classification literature is known as [illegible] (BOW). In this approach, each feature corresponds to a single word found in the training set. Usually a list of [illegible] that are assumed to have no information content is removed from the original text. In order to make the features statistically independent, typically a [illegible] algorithm is used to remove suffixes from words. In our application scenario, item to be recommended are books. Each book is represented by a set of [illegible], where each slot is a textual field corresponding to a specific feature of the book: title, author and textual annotation, that is the abstract of the book. The text in each slot is represented using the [illegible] model taking into account the occurrences of words in the original text. Thus, each instance is represented by three BOWs, one for each slot. This strategy considers separately the occurrences of a word in the slots in which it appears. The idea behind this approach is that by considering the number of occurrences separately in each slot could supply a more effective way to catch the informative power of a word in a document. Stemming and stop words removal have been applied to the documents used in the experiments.

2.2 Related Works

Content-based systems have been used successfully in various domains including Web browsing, news filtering, recommendation services.

[illegible] is a content-based Web agent that suggests Web pages of interest to the user [4]. The system, a Web-browser extension that tracks the users browsing behavior, relies on implicit feedback and uses a set of heuristics to infer the users preferences. For example, Letizia interprets bookmarking a page as strong evidence for the users interests in the page. [illegible] & [illegible] [10] is a software agent that learns a user's interests saved as a user profile, and uses this profile to identify interesting Web pages. The learning process is conducted by first converting HTML source into positive and negative examples, then using algorithms like Bayesian classifiers, a nearest neighbor algorithm and a decision tree learner. [illegible] [2] reads interesting news articles via a speech interface. The news source is Yahoo! News, with an initial training set of interesting news articles provided by the user. Length of listening time provides implicit user feedback on articles read out. A short-term user model is based on TFIDF (cosine similarity), and long-term model based on a naïve Bayes classifier. Mooney and Roy [8] adopt a text categorization method in their [illegible] system, that makes content-based book recommendations exploiting the product descriptions obtained from the Web pages of the Amazon[1] on-line digital store, using a naïve Bayes text classifier. [illegible] [1] is a system that supports users in document searching. User profiles are stored in form of weighted semantic network, that represents terms and their context by linking nodes (words) with arcs representing co-occurrences in some documents. [illegible] support explicit feedback and takes into account not only interest, but also explicit [illegible] interest, and therefore presumably represents a user's idiosyncrasies more accurately. In this aspect, this approach is similar to our method based on the Rocchio relevance feedback that learns both a positive and a negative profile. SiteIF [15] is a personal agent for a multilingual news

[1] http://www.Amazon.com

Web site that learns user's interests from the requested pages that are analyzed to generate or to update a model of the user. Exploiting this model, the system tries to anticipate which documents in the Web site could be interesting for the user. As in ifWeb, profiles and documents are stored as semantic networks. A more recent version of the system is presented in [5], where the authors propose the use of a sense-based document representation to build a model of the user's interests. The system builds the user model as a semantic network whose nodes represent senses (not just words) of the documents requested by the user. Then, the filtering phase takes advantage of the word senses to retrieve new documents with high semantic relevance with respect to the user model.

3 A Relevance Feedback Method for Learning User Profiles

The Rocchio algorithm is one of the most popular learning methods from Information Retrieval and document classification. In this algorithm, documents are represented with the vector space representation and the major heuristic component is the TFIDF (Term Frequency/Inverse Document Frequency) word weighting scheme [12], that reflects empirical observations regarding text:

$$\text{TFIDF}(t_k, d_j) = \underbrace{\text{TF}(t_k, d_j)}_{\text{TF}} \cdot \underbrace{log \frac{N}{n_i}}_{\text{IDF}} \tag{1}$$

where N is the total number of documents in the training set and n_i is the number of documents in which the term t_k appears. $TF(t_k, d_j)$ is a function that computes the frequency of the token t_k in the document d_j. Learning is achieved by combining document vectors of positive and negative examples into a prototype vector $\overrightarrow{c}$ for each class in the set of classes C. Formally, the method computes a classifier $\overrightarrow{c_i} = \langle \omega_{1i}, \ldots, \omega_{|T|i} \rangle$ for category c_i (T is the [illegible], that is the set of distinct terms in the training set) by means of the formula:

$$\omega_{ki} = \beta \cdot \sum_{\{d_j \in POS_i\}} \frac{\omega_{kj}}{|POS_i|} - \gamma \cdot \sum_{\{d_j \in NEG_i\}} \frac{\omega_{kj}}{|NEG_i|} \tag{2}$$

where ω_{kj} is the $TFIDF$ weight of the term t_k in document d_j, POS_i and NEG_i are respectively the set of positive and negative examples in the training set for the specific class, β and γ are control parameters that allow setting the relative importance of all positive and negative examples. The vector model gives the possibility to evaluate the degree of similarity between two vectors using the concept of correlation. This correlation can be quantified, for instance, by the [illegible] between these two vectors. In order to assign a class $\widetilde{c}$ to a document d_j, the similarity between each prototype vector $\overrightarrow{c_i}$ and the document vector $\overrightarrow{d_j}$ is computed and $\widetilde{c}$ will be the c_i with the highest value of similarity. We propose a modified version of this method, that manages documents represented using different slots. As described in Section 2.1, a document is represented by three slots: title, authors and annotation. If [illegible] is the index of the slot ($m = 1, 2, 3$), a book is represented by the concatenation of three bag of words:

$$d_j = \langle w^m_{1j}, \ldots, w^m_{|T_m|j} \rangle$$

where $|T_m|$ is the cardinality of the vocabulary for the slot s_m and w^m_{kj} is the weight of the term t_k in the document d_j, in the slot s_m. Each weight w^m_{kj} is computed as follows:

$$\text{TFIDF}(t_k, d_j, s_m) = \text{TF}(t_k, d_j, s_m) \cdot log \frac{N}{n_{km}} \tag{3}$$

$\text{TF}(t_k, d_j, s_m)$ is the frequency of term t_k in the document d_j in the slot s_m; the inverse document frequency of the term t_k in the slot s_m is computed as the logarithm of the ratio between the total number of documents N and the number of documents containing the term t_k in the slot s_m.

In on-line catalogues, items are often grouped in a fixed number of categories. Our goal is to learn a profile of items preferred by a user in a specific category. Given a user u and a set of rated books in a specific category of interest (for example [illegible] & [illegible]), the goal is to learn a profile able to recognize books liked by the user in that category. The learning process consists in inducing a prototype vector for [illegible]: these three vectors will represent the user profile. The rationale of having distinct components of the profile is that words appearing in a "heavy" slot such as the title could be more indicative of preferences than words appearing in other slots such as the annotation, having a low informative power. For these reasons, each prototype vector of the profile could contribute in a different way to the calculation of the similarity between the vectors representing a book and the vectors representing the user profile. Another key issue of our modified version of the Rocchio algorithm is that it separately exploits the training examples: it learns two different profiles $\overrightarrow{p_i} = \langle \omega^m_{1i}, \ldots, \omega^m_{|T_m|i} \rangle$, for a user u and a category c_i by taking into account the ratings given by the user on documents in that category. The rating $r_{u,j}$ on the document d_j is a discrete judgment ranging from 1 to 10. It is used to compute the coordinates of the vectors in both the positive and the negative user profile:

$$\omega^m_{ki} = \sum_{\{d_j \in POS_i\}} \frac{\omega^m_{kj} \cdot r'_{u,j}}{|POS_i|} \quad (4) \qquad \omega^m_{ki} = \sum_{\{d_j \in NEG_i\}} \frac{\omega^m_{kj} \cdot r'_{u,j}}{|NEG_i|} \quad (5)$$

where $r'_{u,j}$ is the normalized value of $r_{u,j}$ ranging between 0 and 1 (respectively corresponding to $r_{u,j} = 1$ and 10), $POS_i = \{d_j \in T_r | r_{u,j} > 5\}$, $NEG_i = \{d_j \in T_r | r_{u,j} \leq 5\}$, and ω^m_{kj} is the weight of the term t_k in the document t_j in the slot s_m computed as in equation (3) where the *idf* factor is computed over POS_i or NEG_i depending on the fact that the term t_k is in the slot s_m of a book rated as positive or negative (if the term is present in both positive and negative books two different values for it will be computed). Computing two different idf values for a term led us to consider the rarity of a term in positive and negative books, in an attempt to catch the informative power of a term in recognizing interesting books. Equations (4) and (5) differ from the classical formula in the fact that the parameters β and γ are substituted by the ratings $r'_{u,j}$ that allow to give a different weight to each document in the training set. As regards the computation of the similarity between a profile $\overrightarrow{p_i}$ and a book $\overrightarrow{d_j}$, the idea is to compute three partial similarity values between each pair of corresponding vectors in $\overrightarrow{p_i}$ and $\overrightarrow{d_j}$. A weighted average of the three values is computed, by assigning to the similarity for the slots [illegible] and [illegible] an heavier weight than the one assigned to the [illegible] :

$$sim(\overrightarrow{d_j}, \overrightarrow{p_j}) = \sum_{s=1}^{3} sim(\overrightarrow{d_j^s}, \overrightarrow{p_j^s}) \cdot \alpha_s \tag{6}$$

where α_s reflects the importance of a slot in classifying a book. In our experiments we used $\alpha_1 = 0.5$ (title), $\alpha_2 = 0.4$ (authors) and $\alpha_3 = 0.1$ (annotation). Since the user profile is composed by both the positive and the negative profiles, we compute two similarity values, one for each profile. The document d_j is considered as interesting only if the similarity value of the positive profile is higher than the similarity of the negative one.

3.1 INTHELEX

INTHELEX (INcremental THEory Learner from EXamples) is a learning system for the induction of hierarchical theories from positive and negative examples. It is fully and inherently incremental: in addition to the possibility of taking as input a previously generated version of the theory, learning can also start from an empty theory and from the first available example. INTHELEX can learn simultaneously various concepts, possibly related to each other, expressed as (sets of) function free clauses to be interpreted according to the Object Identity assumption [14]. Examples describe the observations by means of only basic non-negated predicates of the representation language, and specifies all the classes for which the observed object is a positive example and all those for which it is a negative one. A positive example for a concept is not considered as a negative example for all the other concepts, unless it is explicitly stated. INTHELEX incorporates two inductive operators, one for generalizing definitions that reject positive examples, and the other for specializing definitions that explain negative examples. Both these operators, when applied, change the set of examples the theory accounts for. In particular, when a positive example is not covered, completeness is restored in one of the following ways:

- replacing a clause in the theory with one of its generalizations against the problematic example;
- adding a new clause to the theory, obtained by properly turning constants into variables in the problematic example;
- adding the problematic example as a positive exception.

When a negative example is covered, consistency is restored by performing one of the following actions:

- adding positive literals that are able to characterize all the past positive examples of the concept (and exclude the problematic one) to one of the clauses that concur to the example coverage;
- adding a negative literal that is able to discriminate the problematic example from all the past positive ones to the clause in the theory by which the problematic example is covered;
- adding the problematic example as a negative exception.

We were led by a twofold motivation to exploit INTHELEX on the problem of learning user profiles. First, its representation language (First-Order Logic) is more intuitive and human readable than values exploited and provided by numeric/probabilistic approaches. Second, incrementality is fundamental in the given task, since new information on a user is available each time he issues a query, and it would be desirable to be able to refine the previously generated profile instead of learning a new one from scratch. Moreover, a user's interests might change in time, a problem that only incremental systems are able to tackle.

Each book description is represented in terms of three components by using predicates `slot_title(b,t)`, `slot_author(b,au)`, and `slot_annotation(b,an)`, indicating that the objects `t`, `au` and `an` are, respectively, the title, author and annotation of the book `b`. Any word in the book description is represented by a predicate corresponding to its stem, and linked to both the book itself and the single slots in which it appears. For instance, predicate `prolog(slott, slottitleprolog)` indicates that the object `slottitleprolog` has stem "prolog" and is contained in slot `slott`; in such a case, also a literal `prolog(book)` is present to say that stem "prolog" is present in the book description. Formerly, INTHELEX was not able to handle numeric values; thus, a discretization was needed. In the new version, it can represent numeric information and manipulate numeric intervals, so the number of occurrences of each word in each slot was represented by means of a predicate `occ(Y,X)`, indicating that term Y occurs X times. Instead of learning a definition for each of the 10 possible votes, just two possible classes of interest are learnt: "likes", describing that the user likes a book (ratings from 6 to 10), and its opposite "not(likes)" (ratings from 1 to 5). Such a discretization step is automatically carried out by an abstraction operator embedded in INTHELEX, whose cost is negligible since each numeric value is immediately mapped onto the corresponding discretized symbolic value. Figure 1 shows an example for the class `likes`. The clause means that the user [illegible] the book with [illegible] that contains in the slot [illegible] a word whose stem is [illegible] (one or two times) and a word whose stem is [illegible] (one or two times), in the slot [illegible] the word [illegible] (one or two times).

3.2 Item Recommender

ITR (ITem Recommender) implements a Bayesian learning algorithm [6] able to classify text belonging to a specific category as interesting or not interesting for a particular user. For example, the system could learn the target concept "[illegible]".

```
likes(50147799) :-
 slot_title(50147799,slott), practic(slott,slottitlepractic),
 occ_1(slottitlepractic), occ_12(slottitlepractic),
 prolog(slott, slottitleprolog), occ_1(slottitleprolog),
 occ_12(slottitleprolog), slot_authors(50147799,slotau),
 l_sterling(slotau,slotauthorsl_sterling), occ_1(slotauthorsl_sterling),
 occ_12(slotauthorsl_sterling), slot_annotation(50147799, slotan),
 l_sterling(50147799), practic(50147799), prolog(50147799).
```

Fig. 1. First-Order Representation of a Book

According to the Bayesian approach to classify text documents, given a set of classes $C = \{c_1, \ldots, c_{|C|}\}$, the conditional probability of a class c_j given a document d is calculated as follows:

$$P(c_j|d) = \frac{P(c_j)}{P(d)} P(d|c_j)$$

In our problem, we have only 2 classes: c_+ represents the positive class (user-likes, corresponding to ratings from 6 to 10), and c_- the negative one (user-dislikes, ratings from 1 to 5). Since instances are represented as a vector of documents, (one for each BOW), and assumed that the probability of each word is independent of the word's context and position, the conditional probability of a category c_j given an instance d_i is computed using the formula:

$$P(c_j|d_i) = \frac{P(c_j)}{P(d_i)} \prod_{m1}^{|S|} \prod_{k=1}^{|b_{im}|} P(t_k|c_j, s_m)^{n_{kim}} \tag{7}$$

where $S = \{s_1, s_2, \ldots, s_{|S|}\}$ is the set of slots, b_{im} is the BOW in the slot s_m of the instance d_i, n_{kim} is the number of occurrences of the token t_k in b_{im}.

In (7), since for any given document the prior $P(d_i)$ is a constant, this factor can be ignored if the only interest concerns a ranking rather than a probability estimate. To calculate (7), we only need to estimate the terms $P(c_j)$ and $P(t_k|c_j, s_m)$, from the training set. Each instance is weighted according to the user rating , normalized in order to obtain values ranging between 0 and 1:

$$w_+^i = \frac{r-1}{9}; \qquad w_-^i = 1 - w_+^i \tag{8}$$

The weights in (8) are used to estimate the two probability terms from the training set TR:

$$\hat{P}(c_j) = \frac{\sum_{i=1}^{|TR|} w_j^i}{|TR|} \tag{9}$$

$$\hat{P}(t_k|c_j, s_m) = \frac{\sum_{i=1}^{|TR|} w_j^i n_{kim}}{\sum_{i=1}^{|TR|} w_j^i |b_{im}|} \tag{10}$$

In (10), n_{kim} is the number of occurrences of the term t_k in the slot s_m of the i^{th} instance, and the denominator denotes the total weighted length of the slot s_m in the class c_j. The length of b_{im} is computed by adding the occurrences of the words in the slot s_m of the i^{th} instance. Therefore, $\hat{P}(t_k|c_j, s_m)$ is calculated as a ratio between the weighted occurrences of the term t_k in slot s_m of class c_j and the total weighted length of the slot. The final outcome of the learning process is a probabilistic model used to classify a new instance in the class c_+ or c_-. The model can be used to build a personal profile including those words that turn out to be most indicative of the user's preferences, according to the value of the conditional probabilities in (10).

4 Experimental Sessions

The experiments have been carried out on a collection of textual book descriptions rated by real users according to their preferences. Eight book categories were selected at the Web site of a virtual bookshop. For each book category, a set of book descriptions was obtained by analyzing Web pages using an automated extractor. Each user involved in the experiments was requested to choose one or more categories of interest and to rate 40 or 80 books in each selected category, providing 1-10 discrete ratings. For each pair user-category, a dataset of 40 or 80 rated books was obtained (see Table 1). For each user we considered:

– [illegible] - number of rated books with the indication of negative (ratings in the range 1-5) and positive (ratings in the range 6-10) ones;
– [illegible] - number (and percentage) of books with a textual annotation (slot annotation not empty);
– [illegible] - average length (in words) of the annotations;
– [illegible] μ [illegible] σ - average rating and standard deviation.

The number of books rated as positive and negative for each user is balanced, except for the user 23 that has rated 39 books as positive and only 1 as negative.

Table 1. Dataset information

User ID	Category	Rated books	Books with Annot.	Avg. Ann. Length	Avg. Rating μ / σ
37	SF, Horror & Fantasy	40 (22+, 18-)	40 (100%)	30.475	4.87 / 2.731
26	SF, Horror & Fantasy	80 (46+, 34-)	70 (87.5%)	19.512	5.49 / 3.453
30	Computer & Internet	80 (40+, 40-)	80 (100%)	56.425	5.31 / 2.462
35	Business	80 (30+, 50-)	78 (97.5%)	64.150	4.21 / 3.488
24c	Computer & Internet	80 (38+, 42-)	76 (95%)	49.100	5.71 / 3.174
36	Fiction & literature	40 (25+, 15-)	40 (100%)	40.225	5.87 / 1.805
24f	Fiction & literature	40 (27+, 13-)	38 (95%)	45.500	6.40 / 2.662
33	Sport & leisure	80 (35+, 45-)	49 (61.25%)	23.337	4.34 / 3.342
34	Fiction & literature	80 (42+, 38-)	70 (87.5%)	44.925	5.61 / 2.492
23	Fiction & literature	40 (39+, 1-)	36 (90%)	45.875	7.25 / 1.089

Almost the totality of books rated by the users contain annotations: the user 33 is the only one with a low percentage. As far as the average annotation length, only users 26 and 33 have values lower than the others. On each dataset, a 10-fold cross-validation was run and several metrics were used in the testing phase. In the evaluation, a book in a specific category is considered as [illegible] by a user if the rating is greater than 5. This corresponds in the Rocchio-based profiling algorithm to having similarity greater than 0; ITR classifies a book d_i as interesting if $P(c_+|d_i) \geq 0.5$, calculated as in equation (7). Symmetrically, INTHELEX considers as relevant books covered by the inferred theory. Classification effectiveness is measured in terms of the classical Information Retrieval notions of [illegible], [illegible] and [illegible], adapted to the case of text categorization [12].

Table 2 shows the results of the experiments using the new Rocchio algorithm and the classical Rocchio one obtained by setting the values of the control param-

Table 2. Performance of the Rocchio algorithms on 10 different datasets

Id	**Precision**		**Recall**		**F1**		**Accuracy**	
	$\beta = 16$ $\gamma = 4$	New Rocchio	$\beta = 16$ $\gamma = 4$	New Rocchio	$\beta = 16$ $\gamma = 4$	New Rocchio	$\beta = 16$ $\gamma = 4$	New Rocchio
37	0.641	0.925	1	0.717	0.781	0.808	0.675	0.800
26	0.797	0.845	1	0.830	0.887	0.837	0.837	0.812
30	0.500	0.534	1	0.875	0.666	0.663	0.500	0.550
35	0.391	0.690	1	0.700	0.562	0.695	0.412	0.762
24c	0.471	0.675	0.966	0.583	0.633	0.626	0.475	0.687
36	0.616	0.767	0.916	0.700	0.737	0.732	0.600	0.675
24f	0.675	0.825	1	0.833	0.805	0.829	0.675	0.750
33	0.651	0.743	0.966	0.917	0.778	0.821	0.737	0.812
34	0.525	0.644	0.975	0.645	0.682	0.644	0.525	0.625
23	0.966	0.975	0.966	0.975	0.966	0.975	0.950	0.950
Mean	0.623	0.762	0.979	0.777	0.750	0.763	0.639	0.742

Table 3. Performance of the systems on 10 different datasets

Id	**Precision**			**Recall**			**F1**			**Accuracy**		
	ITR	INTH.	New Rocchio	ITR	INTH.	New Rocchio	ITR	INTH.	New Rocchio	ITR	INTH.	New Rocchio
37	0.767	0.967	0.925	0.883	0.500	0.717	0.821	0.659	0.808	0.731	0.695	0.800
26	0.818	0.955	0.845	0.735	0.645	0.830	0.774	0.770	0.837	0.737	0.768	0.812
30	0.608	0.583	0.534	0.600	0.125	0.875	0.604	0.206	0.663	0.587	0.488	0.550
35	0.651	0.767	0.690	0.800	0.234	0.700	0.718	0.359	0.695	0.725	0.662	0.762
24c	0.586	0.597	0.675	0.867	0.383	0.583	0.699	0.467	0.626	0.699	0.599	0.687
36	0.783	0.900	0.767	0.783	0.300	0.700	0.783	0.450	0.732	0.700	0.513	0.675
24f	0.785	0.900	0.825	0.650	0.350	0.833	0.711	0.504	0.829	0.651	0.535	0.750
33	0.683	0.750	0.743	0.808	0.308	0.917	0.740	0.437	0.821	0.730	0.659	0.812
34	0.608	0.883	0.644	0.490	0.255	0.645	0.543	0.396	0.644	0.559	0.564	0.625
23	0.500	0.975	0.975	0.130	0.900	0.975	0.206	0.936	0.975	0.153	0.875	0.950
Mean	0.679	0.828	0.762	0.675	0.400	0.777	0.662	0.520	0.763	0.627	0.636	0.742

eters β and γ according to the literature ($\beta = 16$, $\gamma = 4$) [13]. We have carried out a pairwise comparison of the methods, using the nonparametric Wilcoxon signed rank test [9]. Requiring a significance level $p < 0.05$, the test revealed that there is a statistically significant difference in performance both for precision and accuracy in favor of the modified Rocchio and for recall in favor of the classical Rocchio method, but not as regards F1. These results led us to conclude that the new method is more effective than the traditional one in the e-commerce domain, where [illegible] is a key word in giving recommendations: the systems should minimize false positive errors because it is better to provide users with a few number of high quality recommendations than to overload users with many recommendations that they should manually filter. Table 3 shows the results of the second experiment aimed at comparing the new Rocchio method with the ones implemented by INTHELEX and ITR in terms of average precision, recall, F1 and accuracy of the models learned in the 10 folds for each dataset. The last row of the table reports the mean values, averaged on all datasets. The results of INTHELEX and ITR are described in more details in [3]. The most important result is that the proposed method outperforms the other ones as regards accuracy. It is surprising to observe that the algorithm reaches high values of precision and recall for the users 26 and 37, even if the average annotation lengths of the documents rated by the users are among the shortest in the dataset. This means that the profiles contain few words for computing similarity on new documents, but these words are indicative of the users' preferences. In general, the new Rocchio algorithm outperforms ITR in precision, but not INTHELEX (requiring a significance level $p < 0.05$ the systems are equivalent). Another remark worth noting is that theories learned by the symbolic system are very interesting from a human understandability viewpoint, in order to be able to explain and justify the recommendations provided by the system. From what said above, it seems that the approaches compared in this paper have complementary, [illegible]. This naturally leads to think that some cooperation could take place in order to reach higher effectiveness of the recommendations. For instance, since the new Rocchio method has a better accuracy, it could be used for selecting which items are to be presented to the user. Then, some kind of filtering could be applied on them, in order to present to the user first those items that are considered positive by the symbolic theories, that are characterized by a slightly better precision.

5 Conclusions

The paper proposed a new Rocchio-based method able to discover user preferences from textual descriptions of items in online catalogues of e-commerce Web sites. In order to evaluate the effectiveness of the approach, we performed an experimental session involving real users. Results have been compared with respect to the performance of an ILP approach and a probabilistic one. The comparison highlighted the usefulness and drawbacks of each method, suggesting possible ways of integrating the approaches to offer better support to users.

Acknowledgments

This research was partially funded by the European Commission under the 6th Framework Programme IST Integrated Project VIKEF - Virtual Information and Knowledge Environment Framework (Contract no. 507173, Priority 2.3.1.7 Semantic-based Knowledge Systems; more information at `http://www.vikef.net`).

References

1. F. Asnicar and C. Tasso. ifweb: a prototype of user model-based intelligent agent for documentation filtering and navigation in the word wide web. In *Proceedings of 1st Int. Workshop on adaptive systems and user modeling on the World Wide Web*, pages 3–12, 1997.
2. Daniel Billsus and Michael J. Pazzani. A hybrid user model for news story classification. In *Proceedings of the Seventh International Conference on User Modeling. Banff, Canada*, pages 99–108, 1999.
3. F. Esposito, G. Semeraro, S. Ferilli, M. Degemmis, N. Di Mauro, T.M.A. Basile, and P. Lops. Evaluation and validation of two approaches to user profiling. In *Proc. of the ECML/PKDD-2003 First European Web Mining Forum*, pages 51–63, 2003.
4. H. Lieberman. Letizia: an agent that assists web browsing. In Claire Nédellec and Céline Rouveirol, editors, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 924–929, 1995.
5. B. Magnini and C. Strapparava. Improving user modelling with content-based techniques. In *Proc. of 8th International Conference on User Modeling*, pages 74–83. Springer Verlag, 2001.
6. T. Mitchell. *Machine Learning*. McGraw-Hill, New York, 1997.
7. D. Mladenic. Text-learning and related intelligent agents: a survey. *IEEE Intelligent Systems*, 14(4):44–54, 1999.
8. R. J. Mooney and L. Roy. Content-based book recommending using learning for text categorization. In *Proceedings of the 5^{th} ACM Conference on Digital Libraries*, pages 195–204, San Antonio, US, 2000. ACM Press, New York, US.
9. M. Orkin and R. Drogin. *Vital Statistics*. McGraw-Hill, New York, 1990.
10. M. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.
11. J. Rocchio. Relevance feedback information retrieval. In Gerald Salton, editor, *The SMART retrieval system - experiments in automated document processing*, pages 313–323. Prentice-Hall, Englewood Cliffs, NJ, 1971.
12. G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, 1983.
13. F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47, 2002.
14. G. Semeraro, F. Esposito, D. Malerba, N. Fanizzi, and S. Ferilli. A logic framework for the incremental inductive synthesis of datalog theories. In N. E. Fuchs, editor, *Logic Program Synthesis and Transformation*, number 1463 in Lecture Notes in Computer Science, pages 300–321. Springer-Verlag, 1998.
15. A. Stefani and C. Strapparava. Personalizing access to web sites: The siteif project. In *Proc. of 2nd Workshop on Adaptive Hypertext and Hypermedia*, 1998.

Data Mining Based on Objects in Video Flow with Dynamic Background

Cheng Zeng, JiaHeng Cao, Ying Fang, and Pei Du

Computer School, Wuhan University, Wuhan 430072, P.R. China
{zengc, jhcao}@whu.edu.cn

Abstract. This paper presents a model OMDB for mining the region information of non-rigid foreground object in video flow with dynamic background. The model constructs RDM algorithm and optimize the strategy of region matching using Q-learning to obtain better motion information of regions. Moreover, OMDB utilizes NEA algorithm to detect and merge gradually object regions of foreground based on the characteristics that there is motion difference between foreground and background and the regions of an object maintain integrality during moving. Experimental results on extracting region information of foreground object and tracking the object are presented to demonstrate the efficacy of the proposed model.

1 Introduction

Mining objects information in video streams is gaining more attention of many research communities due to the needs of some applications such as surveillance for security, video content understanding, video database indexing, the kernel of which is to detect and track regions of object. Two key concepts of object detection are camera state and target type. The first including static state and dynamic state ascertains whether the background in scene changes. And the other differentiates tracked target as rigid or not.

Many traditional researches suppose the camera is static. Their research covers: multi-objects tracking, target partially occluded tracking, maintaining the edge integrality of objects during tracking, motion estimating and so on. Most applications require detecting target trajectories from video sequence and maintain effectually tracking while camera is dynamic.

Detecting and tracking rigid objects is widely applied in traffic control. For example, [1] realizes multi-vehicles tracking and estimating, [2] recognizes the contour of rigidly deformable object. However, the factors including shape change, interior activity or drape result in the phenomenon of self overlapping, occlusion or shadow during non-rigid objects are moving that makes it difficult to deal with. Romer [3] made progress in the field of tracking multiple of non-rigid objects, but the targets must be rudely expressed with a rectangular pane. [4] detects the edge information of tracked objects, however lots of contour errors emerge when color or texture is similar between foreground and background, because of lacking correlative restriction.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 382–390, 2005.

The principal difficulty of detecting and tracking objects in dynamic background is to mine object regions merely based on the difference of features between foreground and background during the period of camera moving without any relative object information provided. Then we utilize the methods [6] about object tracking in static background. This paper presents a novel model OMDB which mines non-rigid foreground objects in video flow with dynamic background. It combines region delayed matching algorithm (RDM) with Q-learning method to obtain motion vectors of regions in video frame. Then the regions with obvious foreground character are extracted in term of motion difference between foreground and background. At last, the whole foreground regions will be estimated by nibble-extend algorithm (NEA) utilizing the interrelation of all parts that compose the moving object, and multi-objects will be separated.

2 OMDB Model Framework

We describe a novel framework for detecting and tracking non-rigid objects in video flow with dynamic background, as shown in Fig.1. The model uses RDM to realize region matching between adjacent video frames and optimizes the matching effect with Q-learning. For some reasons, including light disturbing, overlapping, draping and so on, which result in morbid matching of regions, we adopt K-S statistics to resolve the problem and improve matching precision. The result of region matching will figure out motion vector of the region. In term of the status, there are many differences between motion manners of foreground regions which are active and that of background regions which are passive, we can segment those regions possessing obvious foreground characteristic at first and spread the foreground range with dynamic texture segmentation. According to some rules, NEA gradually extends foreground regions, and clusters the regions based on objects with path table. We store the features such as color, texture etc., which are extracted by the methods researched by ourselves before and motion vector of regions into a feature database. In subsequent frames, the object regions separated and correlative information in the feature database will be integrated to track the target. The system will extract anew the object features every other time, and update the feature database to maintain the self-adaptability.

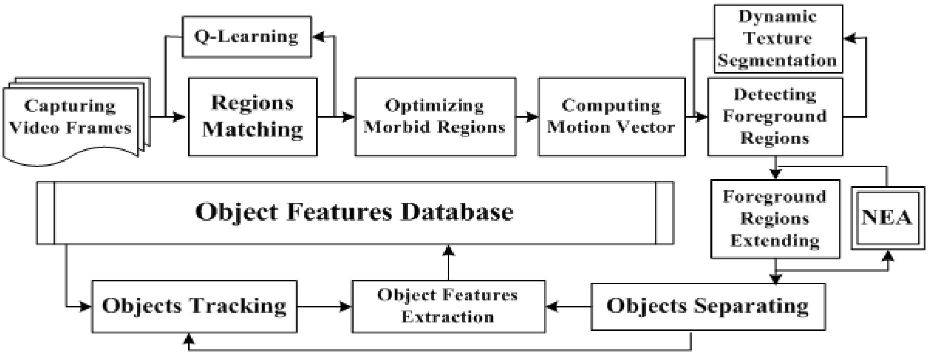

Q-Learning
Dynamic Texture Segmentation
Capturing Video Frames
Regions Matching
Optimizing Morbid Regions
Computing Motion Vector
Detecting Foreground Regions
Object Features Database
Foreground Regions Extending
NEA
Objects Tracking
Object Features Extraction
Objects Separating

Fig. 1. OMDB model framework

3 RDM Algorithm Based on the Relativity of Regions

Under the condition that camera is moving, all targets and the background are moving so we can't segment foreground objects with the methods based on static background and the differences between adjacent frames. As a result, we take the method of region matching to compute displacement vectors of all regions and detect regions belonging to foreground based on the motion difference between foreground and background.

3.1 Region Segmentation and Similitude Function

Motion estimation based on feature matching relies on the result of region segmentation. We segment static video frames with hierarchical watershed algorithm [5] which can avoid overly segmenting frame in an extent by selecting appropriate segmentation granularity.

We assume f_0, f_1 are two consecutive frames and f_0 is the reference frame. After segmenting f_0, f_1 with watershed algorithm, we respectively obtain region set R and S:

$$R=\{r_i : i=1,2,...,m\}, S=\{s_j : j=1,2,...,n\} \tag{1}$$

$$r_k \cap r_l = \Phi, s_k \cap s_l = \Phi, \forall k,l \quad k \neq l \quad k,l=1,2,...,m$$

For describing the relativity among regions of static frames, we assume two optional regions which have common boundary possess an adjacent relation. G_r and G_s respectively denote the adjacent relation set of regions in R and S: $G_r=(R,E)$, $G_s=(S,E')$. E and E' is the adjacent relation. We merge the region whose area is smaller than a threshold into an adjacent region which has longest common boundary with it. The purpose of regions matching is to detect the suited region in S which corresponds to the one in R.

Definition 1: Region r $\in$R, S denotes the region set of subsequence frames of R. For each region s $\in$S, if $r \cap s \neq \Phi$ in the same space domain, then s is called the candidate of region matching.

Definition 2: Similitude function shows the extent of comparability between the selected region and its candidate region, evaluated by color mean, covariance, area and shape parameter of the region. The Similitude function of r_i and s_j is represented by $RS(r_i, s_j)$:

$$RS(r_i,s_j)=\omega_{color}\cdot RS_{color}+\omega_{\mathrm{var}}\cdot RS_{\mathrm{var}}+\omega_{area}\cdot RS_{area}+\omega_{overlap}\cdot RS_{overlap}+\omega_{form}\cdot RS_{form} \tag{2}$$

Where the weight ω corresponding to each feature is able to be adjusted based on different application. RS_{color} denotes the color comparability of regions which is computed by color average μ:

$$RS_{color}=1-\left|\mu_i-\mu_j\right| / \left|\mu_i+\mu_j\right| \tag{3}$$

In (3), μ_i and μ_j denote the color average of region r_i and s_j, respectively. RS_{var}、RS_{area} and RS_{form} respectively denote the comparability of color covariance σ, area A and shape parameter F=$\left\|\Gamma\right\|^2/4\pi A$, whose computing method is the same as RS_{color}, Where ||Γ|| denotes the number of pixels of region contour, A is the area of region.

$RS_{overlap}$ denotes the comparability of region area which is described as follows:

$$RS_{overlap} = \left|A_{ij}\right| / \left|\min(A_i, A_j)\right| \tag{4}$$

Where A_i and A_j respectively denote the area of region r_i and s_j, A_{ij} denotes the overlapping area between two regions.

3.2 Optimizing Region Matching Based on Q-Learning

The result error of region matching which uses only similitude function is so big that it is necessary to regard the characteristic, adjacent relations among local regions maintain stabilization in video sequence, as the restrictive condition. We utilize the thought of cumulatively delayed feedback to express the extent that all regions in the same object approve the candidate of region matching.

Definition 3: The most optimistic matching strategy is represented by the set of the best selecting of all regions in R:

$$\pi^* = \arg\max_{\pi} V^{\pi}(R) \tag{5}$$

Where $V^{\pi}(R)$ denotes the value function applying matching strategy π which is defined as follows:

Definition 4: Suppose matching strategy π is a function from R to S, $V^{\sigma}(r)$ is the matching function while region r selects matching σ ($\sigma \in \pi$), so we call

$$V^{\pi}(R) = \sum_{r \in R} V^{\sigma}(r) \tag{6}$$

as the value function of matching strategy π in R.

In other words, global matching strategy will be the most optimistic one when matching functions of all regions are best.

We look the region and its adjacent regions as a group of states, and different candidate of region matching as different action. The computation of each region matching function is regarded as a study model that an agent obtains the best feedback in a group of “state-action” sequence with Q-learning theory. Then we select and reserve the regions which belong to the same object, and feed back gradually the matching results. At last the matching function will be optimized with cumulatively delayed feedback.The most optimistic matching of local region is calculated as follows:

$$V^*(r) = \arg\max_{a} Q(r, \alpha) \tag{7}$$

Where Q(r,α) is the matching function of region r with matching action α, and is looked as evaluating function in the model which denotes the best feedback that r selects action α.

$$Q(r,\alpha) \equiv RS(r,s) + \gamma Adjoin*(r,\alpha) \tag{8}$$

Where γ is a conversion function. The value of Q function reflects the overall feedback of r and its adjacency regions after r selects the matching action α. Function Adjoin*(r, α) denotes the approved extent that r selects matching action α for adjacency regions. Q-learning corresponds to the process that an agent searches the most optimistic matching strategy, the kernel of which is to calculate delayed feedback of the region based on the immediate feedback of its adjacency regions with an appropriate method. We can calculate delayed feedback with the accumulatively matching result of subsequent states:

$$Adjoin*(r,\alpha) = \frac{1}{k}\sum_{Q_j>T} \max_j Q_1(r_1,\alpha_{1j}) + \max_j Q_2(r_2,\alpha_{2j}) + \cdots \tag{9}$$

Where r_1, r_2, … denote the adjacency regions of r, k is the sum of adjacency regions whose Q values is larger than threshold T. α_{ij} denotes the candidate matching of region r_i which is similar to action α. The choice of α_{ij} and the condition Q_j>T ensure that selected adjacency regions and r belong to the same object.

4 Updating Morbidly Matching Regions with K-S Statistics

In the process of camera moving, the interior regions of object emerge morbid n:m matching, which affects the computation of motion vector. If the overall differences between region r_i and its adjacency regions are larger than a certain threshold, we consider a morbidly matching emerges in the region. K-S statistics is used to update region matching:

① Region r_i and its optionally adjacency region will be merged into a initial region set R_k^*. We assume that the gray intensity distribution of R_k^* is modeled by a Gaussian distribution $N(x,\mu_k,\sigma_k)$. The matching region in S of r_i is s_i. The matching regions in S which correspond to all regions in R_k^* constitute set S_l^* ;

②Measure the overall difference of cumulative distribution functions between R_k^* and S_l^* using K-S statistics (D):

$$D(R_k^*,S_l^*) = \max_{0\le x\le 255}\left|\int_0^x \frac{1}{\sqrt{2\pi\sigma_k}} e^{\frac{(x-\mu_k)^2}{2\sigma_k^2}} dx\right| - \left|\int_0^x \frac{1}{\sqrt{2\pi\sigma_l}} e^{\frac{(x-\mu_l)^2}{2\sigma_l^2}} dx\right| \tag{10}$$

The significance level of D is then computed by the formula as follows:

$$Q_{ks}(\xi) = 2\sum_{m=1}^{\infty}(-1)^{m-1} e^{-2m^2\xi^2} \tag{11}$$

$$\xi = (N_e + 0.12 + 0.11/\sqrt{N_e}) \cdot D \quad N_e = \frac{size(R_k^*) \times size(S_l^*)}{size(R_k^*) + size(S_l^*)}$$

Where N_e is the effective area of the region set.

③ Add an adjacency region r_j into R_k^* and add a matching region corresponding to r_j into S_l^*. Repeatedly perform step 2 to figure out $Q_{ks}(\xi)'$;

④ Calculate $Q^*=|Q_{ks}(\xi) - Q_{ks}(\xi)'|$;

⑤ When Q^* tends to be stable, i.e. the variance of Q^* is less than a threshold along with the increase of adjacency region, we should stop expanding set R_k^* and respectively substitute R_k^* and S_l^* for r_i and s_i.

5 Separation of Moving Object from Globally Dynamic Video Flow

For the integrality of object regions, it is primary to separate the regions of moving object from dynamic background. The classical methods assume the camera is static, so that it is easy to distinguish moving foreground from static background. But while camera also moves, overall regions in frame are in the state of movement and the difference between foreground and background is very blurry.

The different distances between each region of background and the camera generate dissimilar motion vector. Each object region has different motion vector during not-rigid foreground objects moving. Therefore, it is difficult to directly distinguish the regions between establishing shot in background and slowly moving non-rigid object in foreground. In this section, we firstly detect the variance of motion modality between foreground and background and separate obvious region sets of foreground associating with dynamic texture segmentation from video flow. Then NEA algorithm is presented to gradually extend the region sets to obtain the integrated regions based on object.

5.1 Initial Estimation of Foreground Regions

Let us assume the camera maintains flatly moving whose unit motion vector is $\vec{v}$ during △t in order to reduce the complication of the problem. The motion vector of each region in video frame is $\vec{v}_i$. We consider the regions which satisfy any one of the below rules can be estimated to belong to foreground.

① The angle between two lines respectively containing $\vec{v}$ and $\vec{v}_i$ is larger than θ, i.e. the difference of motion trajectory between the object and the camera is very large.

② The unit vector of $\vec{v}_i$ is equal to $-\vec{v}$, i.e. the velocity of object is obviously faster than camera.

It should be noticed that we can’t detect the regions of object if the object is static relative to the background and we haven’t any information about the object.

5.2 Foreground Regions Expanding Based on Dynamic Texture Segmentation

Because the regions segmented in section 3 are small, the number of foreground regions, which are detected in last section and possibly appear dispersancy, is very large. It is not suitable to directly recognize the object regions. We use the method of dynamic texture segmentation to deal with foreground regions which can cluster the regions possessing strong relation each other.

We make use of the existing technology [8] about dynamic texture segmentation to cluster the regions which are recognized foreground regions by the following rules. For more describing, A_x and B_y denote the region segmented by dynamic texture method and watershed algorithm, respectively.

① If $\exists x$, A_x contains multiple of B_y, then B_y possesses larger boundary weight. In other words, if A_x intersect B_y and the area of A_x is far bigger than that of B_y, then we expand the boundary of A_x to that of B_y.

② If $\exists x$, B_y contains or intersect A_x and the area of B_y is bigger than that of A_x, then A_x will be merged into the region of texture segmentation between which and A_x there is the longest boundary.

③In general, the set of adjacency regions which have similar characters possesses the similarly global motion feature. If the proportion of foreground regions to A_x exceeds a certain percent (60%), then we transform A_x into foreground region Φ_i whose motion vector is the mean of that of overall region in A_x.

5.3 NEA Algorithm

While the velocity of object is slower than that of the camera, the greater part of object regions are confused with establishing shot in background. NEA algorithm monitors the regions which are relative to the ones recognized before based on the characteristic that the interior regions maintain integrality during object moving, and gradually extend foreground regions. At last, it can separate multi-objects.

Suppose { φ_i } denotes the set of foreground regions and $\{O_j\}$ is the set of regions monitored. We can ascertain two public tangents for two optionally disjoint foreground regions. The tangents and the boundaries of foreground regions surround an area R_x. These regions contained in or intersecting R_x will be added into $\{O_j\}$.

① If $\exists l$, which makes the dynamic texture feature of region O_l change during $\triangle$t, then it will be recognized as background;

② Otherwise if $\exists l$, motion vector of region O_l maintain invariability, then it will become the candidate region. $\{H_m\}$ denotes the set of candidate region.

③ Compare regions in $\{H_m\}$ with adjacent regions not in $\{O_j\}$ by turns. If motion vector of one adjacent region is similar to that of region compared in $\{H_m\}$, then the adjacent region will be added into $\{H_m\}$.

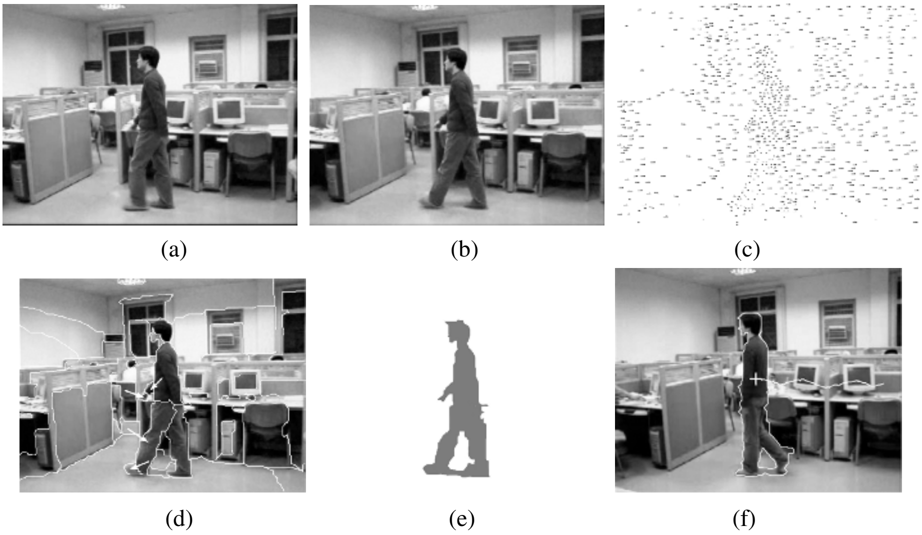
④ Recursively perform step 3 until $\{H_m\}$ stops extending.

⑤ If the dynamic texture feature of any one region in $\{H_m\}$ changes, the region will be recognized as background; Otherwise, $\{H_m\}$ will be added into { φ_i }.

While there are multi-objects in video, we utilize path table to separate them. A link-line will be constructed between optionally two adjacent regions. If φ_i is able to attain φ_j $(i \neq j)$ through link-lines, we consider φ_i and φ_j belong to the same object. Otherwise, they belong to different objects. The rest can be done in the same way. At last, we will obtain the region sets based on object.

6 Experimental Results

For evaluating the effectiveness of OMDB model, we have rested the process of distinguishing the regions of moving foreground objects from video flow with dynamic background. The experiment were performed on a Pentium Ⅳ 2.0GHz PC with the model that was implemented using Matlab 6.5. The speed of sampling is 3 frames/s. The moving direction of camera and target are all from right to left, but the velocity of camera is faster than that of target.



(a) (b) (c)

(d) (e) (f)

Fig. 2. Separating foreground regions from dynamic background: (a,b)two adjacent video frames; (c) region motion vector with RDM algorithm; (d) dynamic texture segmentation; (e)region set of object with NEA algorithm; (f)object tracking

Fig.2(a, b) show two adjacent frames captured from video flow. The motion vectors of regions are calculated by RDM algorithm and K-S statistics (Fig.2(c)). We can see the moving direction of overall regions of background is almost the same but opposite to the camera, although their values are variants due to different distance from the camera. Because global moving velocity of foreground object is slower than that of camera, most of region moving vectors of the object are similar to that of background. But the velocity of regions pointed by arrowhead whose moving direction are different from that of other regions is faster (shown in Fig.2(d)) so that they can be separated at first. Although a few of regions have been initially recognized as foreground in term of the rule introduced in section 5.2, the proportion of theirs in the range segmented based on dynamic texture is too low. As a result, we will look these regions as noise and ignore them. Fig.2(e) shows the integrated object regions dealt with NEA algorithm. The regions around head and feet emerge a few of errors because of the similarity of texture and color between foreground and background. At last, we implement to track the target in subsequent frames and store the motion trajectory as shown in Fig.2(f).

7 Conclusions

We present a model OMDB for mining the region information of non-rigid foreground object in video flow with dynamic background. It can be used to construct region delayed matching RDM algorithm, optimize the strategy of region matching using Q-learning and solve the problem of morbid regions matching with K-S statistic to obtain better motion information of regions. Besides, OMDB utilizes NEA algorithm to gradually detect and merge object regions of foreground based on the characteristics that there is motion difference between foreground and background and object regions maintain integrality during moving. Combining the research production of ourselves before with OMDB, we realize to extract object features and track the object in subsequent frames.

In order to achieve better results, we take the methods of machine learning which is time-consuming especially in complex scene so that the speed of capturing video frame is restrained. In future work we plan to reduce the computing complication of OMDB and extend our work to mine object information in video flow while the camera is in complex motions such as rotation, scale and so on.

References

1. Magee, D.R. Tracking multiple vehicles using foreground, background and motion models. Image and Vision Computing 22 (2004) 143–155
2. Sclaroff, S. and Isidoro, J. Active blobs: region-based, deformable appearance models. Computer Vision and Image Understanding 89 (2003) 197–225
3. Rosales, R. and Sclaroff, S. A framework for heading-guided recognition of human activity. Computer Vision and Image Understanding 91 (2003) 335–367
4. Patras, I., Hendriks, E.A. and Lagendijk, R.L. Semi-automatic object-based video segmentation with labeling of color segments. Signal Processing: Image Communication 18 (2003) 51–65
5. Grau, V. and Mariano A.R. Hierarchical image segmentation using a correspondence with a tree model. [J] Pattern Recognition 37(2004) 47-59
6. ZENG, C., Cao, J.H. and Peng, Z.Y. A novel 3D video trajectory tracking method. The Fourth International Conference on Computer and Information Technology (2004) 221-226
7. Kok, J.R. and Vlassis, N. Sparse tabular multiagent Q-learning. Proceedings of the Annual Machine Learning Conference of Belgium and The Netherlands (2004) 65-71
8. Doretto, G. and Chiuso, A. Dynamic Textures. International Journal of Computer Vision 51(2), 91–109, 2003

An Approach to Compressed Image Retrieval Based on JPEG2000 Framework

Jianguo Tang[1,2], Wenyin Zhang[1,2], and Chao Li[1]

[1] Chengdu University of Information Technology, 610041, P.R. China
[2] Chengdu Institute of Computer Applications,
Chinese Academy of Sciences, Chengdu 610041, P.R. China
zwy218@hotmail.com

Abstract. As the latest effort by JPEG in international standardization of still image compression, JPEG2000 contains a range of important functionalities superior to its earlier DCT based versions. In the expectation that the compression standard will become an important digital format for many images and photographs, we present our recent work in this paper on image indexing and retrieval directly in wavelets domain, which is suitable for JPEG2000 compressed image retrieval without involving its full decompression. Our methods mainly extract histogram features from those significant wavelet coefficients according to the EBCOT of JPEG2000 for compressed image retrieval. While our method gains the advantage of eliminating decompression, the experiments also support that the retrieving accuracy is better than the existing counterparts.

Keywords: Image Retrieval, Image Index, JPEG2000.

1 Introduction

Since the launch of the DCT-based JPEG image compression standard in early nineties, it has proved to be a great success in providing tools for digital imaging, digital photography, multi-media, and computer vision. As the digital technology advances, the modern digital imagery is becoming more and more demanding, not only from the quality point of view, but also from the image size aspect, which makes the DCT based JPEG out of date. As a result, a new standard, JPEG2000 [1] developed from wavelets transform, emerges in 2000, which provides a range of features that are of importance to many high-end and emerging applications by taking advantage of new technologies developed over the past decade. Together with other digital compression standards, JPEG2000 is no doubt expected to be one of the popular compressed image formats in the coming years. So, content based retrieval for JPEG2000 images has been paid more and more attention recently.

The key technique of the JPEG2000 is that the entropy coding employs a bit-modelling algorithm, where the wavelet coefficients are represented in the form of combination of bits that are distributed in different bit planes. The use

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 391–399, 2005.

of bit modelling provides a hierarchical representation of wavelet coefficients by ordering the bit-planes of wavelet coefficients from the MSB to the LSB. Hence, the formed bits-streams with inherent hierarchy can be stored or transferred at a given bit rate without destroying the completeness of the content of the image.

Compared with earlier versions of JPEG standards, JPEG2000 can be highlighted to have: (a) superior performance in low bit-rate compression; (b) continuous tone and bit-level compression; (c) integration of lossless and lossy compression; (d) progressive transmission by both pixel precision and resolution; (e) region-of-interest coding; (f) open architecture; (g) robustness to bit errors and (h) potential and possibilities for protective image security via digital watermarking, encryption and signature etc.

While those compression techniques enable those image data to be manageable, content access to large number of such compressed images become a new challenge to the community Extracting Image Features in JPEG-2000 Compressed Images of information systems, multimedia and computer science. Therefore, it is important to develop indexing techniques based on the JPEG2000 standard algorithm. Before the appearance of JPEG2000, there are many researches on content-based image indexing and retrieval in DWT domain such as [2], but it is more meaningful for these index or retrieval methods to intergrade with the JPEG2000 framework. Many people have done much work in this fields [3, 4, 5, 6, 7]. Liu and Mandal provided a progressive bit plane indexing scheme in the JPEG2000 framework [3]. A 2D significant bit map and a 2D histogram of significant bit of wavelet coefficients are used as the image indices. Image retrieval is performed by matching the index of the query and candidate images from the the database. In [4], features are extracted from the header of bitstream packets. In [5] a fast, block-based JPEG2000 image indexing system in compressed domain which achieves high memory efficiency. Bhalod et al [6] proposed a region based indexing technique based on JPEG2000 formats, in which specific regions of interest (ROI) are tracked and analyzed through different layers of the wavelet transform in the coding process and then shape features are extracted from the ROI sketch in the uncompressed domain and texture and color features are extracted in the compressed domain at different wavelet resolutions corresponding to these regions. Indexing and retrieval are based on the combination of these features. Jiang . . extract shape and texture features from those significant wavelet coefficients and transform their energy into histogram-based indexing keys for compressed image retrieval [7], which obtained much better results than the method [2]. Ni . . [8] defined a tree distance in the JPEG2000 framework and achieved a better retrieval results. Though Chen . . [9] tended to detect tamper in the JPEG2000 images, their methods is suitable for image retrieval.

To extract content information or indexing keys which are capable of characterizing the content of those compressed images via JPEG2000, the first step is to perform entropy decoding to get us into those significant coefficients in bit-planes inside each sub-band. In this paper, we directly obtain the bit-planes within the JPEG2000 compressed bitstream by entropy decoding, then we extract two image indices for image retrieval. One index is Stage-Plane Bit Histogram (SPBH), and the other is Local-Block Bit Histogram (LBBH). Both of them are extracted in a progressive manner and the extraction can be stopped at any point without touching the remaining bit stream of the compressed image. Since the entropy decoding designed in JPEG2000 works in bit-planes, it is perfect for extracting content information

The remaining part of the paper is organized as follows. Section 2 describes image feathers extraction from wavelet domain and the Distance Metrics between two images. Section 3 discusses our experimental results, and finally Section 4 concludes the paper.

2 Proposed Retrieval Scheme

2.1 Some Basic Knowledge for JPEG2000

The organization of the compressed bitstream of JPEG2000 is mainly featured in two hierarchies: resolution scalability and SNR scalability. The resolution scalability is achieved easily by DWT domain, and the SNR scalability is derived from the decomposed binary representation of the wavelet coefficients, i.e. bit-planes. The two image features, SPBH and LBBH histograms proposed in this paper, are based on the two hierarchies in a progressive manner. By matching these two Histograms between the query image and the candidate images, the similar images can be retrieved.

According to the JPEG2000 FCD15444-1, if an image is wavelet transformed with N_L decomposition levels, the image will have $N_L + 1$ distinct resolutions with a total of $3{*}N_L{+}1$ sub-bands, as shown in fig.1. In each resolution, we define the three high frequency sub-bands (HL, LH and HH) as a Stage, with total of $N_L + 1$ stages [3], among which the lowest resolution is defined as Stage 0, as shown in fig.1. According to the parent-children relation of the DWT coefficients, we combine them as a Local Block, as shown in fig.2. If the wavelet filter is of limited support, a Local Block only represents a limited image area.

According to the bit-plane encoding algorithm in EBCOT, For each Stage and Local Block, we use binary format to represent each DWT coefficient with the order of the bits descending from the MSB to LSB. A bit-plane is the decom-

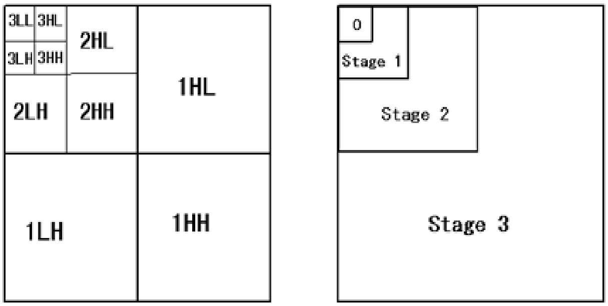


Fig. 1. Illustration of Resolutions and Stages for three levels decomposition of wavelet

position of the binary representation for a given set of coefficients. In JPEG2000 International Standard, a bit-plane refers to all the bits of the same magnitude in all coefficients or samples. This could refer to a bit-plane in a component, tile-component, code-block, region of interest, or other.

2.2 The Construction of Image Indices

In this part, we present a progressive wavelet bit-plane indexing scheme, one of which is SPBH, and the other is LBBH.

SPBH Histogram. For each of the $N_L + 1$ Stages, Let M_b be the Number of bit-planes determined by the maximum value of wavelet coefficients. Given a specified Stage $j, 0 \leq j < N_L + 1$, we use M_b bit-planes from MSB to LSB to construct the SPBH. The value of each bin of the SPBH is equal to the number of the "1" bits derived from the specified bit-plane and the specified stage. The SPBH is different from the index (named Index-2, defined by [3]) which only uses the most significant bits while abandons other bits.

Let $H_{sp}(X)$ be the SPBH index of the image X, and $B[i,j]$ the value of the SPBH at i^{th} plane of j^{th} stage, $H_{sp}(X)$ is then obtained as follows:

$$H_{sp}(X) = \begin{bmatrix} B[0,0] & b[0,1] \cdots\cdots & & B[0,s-1] \\ B[1,0] & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \vdots & & \ddots & \vdots \\ B[L-1,0] & \cdots & \cdots & B[L-1,s-1] \end{bmatrix} \tag{1}$$

As we know, most information (most important coefficients) of an image concentrates in the lower resolutions (stages) sub-bands, at the same time, MSB contains more information than the LSB. If we use stages with higher frequency and Bit-planes with LSB, the size of the index will be larger which in turn will be increase the computational complexity for image retrieval. In addition, the larger index may degrade the retrieval performance. Hence, a smaller value of L and S are used in practice.

LBBH Histogram. The index SPBH provides a global information about the spread of energy across an image. Here, by considering the Father-Children relation of DWT coefficients, we give another index LBBH, which presents the local information about the image energy. The EBCOT algorithm doesn't make use of the relativity between different coefficients, which is regarded as a main disadvantage. Though using the relativity can promote the coding ratio, it puts the limitation on the optimal truncation technique of the EBCOT. The LBBH index not only take the Father-Children relativity of DWT coefficients into consideration, but also combine with the EBCOT algorithm.

At first, according to the Father-Children relationship of DWT coefficients, we reorganize a low frequency coefficient and its descendants into a block with

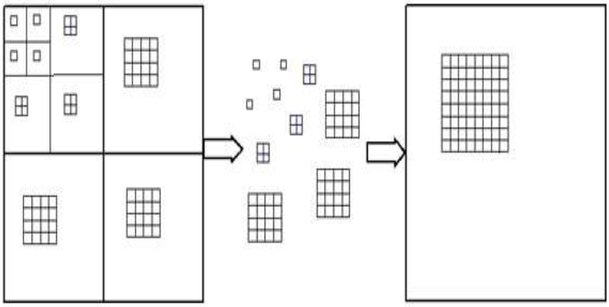


Fig. 2. The reorganization of the DWT coefficients from three levels decomposition according to their Father-Children relationship

the size of $2^d \times 2^d, d$ is the level number of decomposition. The Fig.2 shows the process of rearrangement.

For a $2^d \times 2^d$ block, there are M_b bit-planes at most from the MSB to LSB with the size of $2^d \times 2^d$. Let the total number of blocks be N_b. We use $L^i_j[k,l]$ to present a bit-plane of a block, where i is the number of bit-layer $(0 \leq i < M_b)$ and j is the number of block $(0 \leq j < N_b)$. Let W and H be the image's Width and Height. Now we build a series of LBBH histograms from MSB layer to LSB layer with 4^d bins as follows:

$$H_i[m] = \sum_{w=0}^{W/2^d} \sum_{h=0}^{H/d^d} \left\{ \begin{array}{l} 1; \sum_{k=0}^{2^d} \sum_{l=0}^{2^d} L^i_j[k,l] = m, 0 \leq m \leq 4^d; \\ 0; Otherwise. \ \ 0 \leq i < M_b, 0 \leq j < N_b. \end{array} \right\} \tag{2}$$

Both indices are based on histogram scheme which is invariant to translation and rotation. For scale invariance, we normalize the both indices as follows:

$$H^*_{sp}[X] = H_{sp}[X]/(W \times H) \tag{3}$$

$$H^*_i[m] = H_i[m]/(W/2^d \times H/2^d) \tag{4}$$

2.3 Distance Metrics

For image retrieval, the index of a query image is compared with the corresponding index of a candidate image. For a query image Q and a candidate image C, the distance metrics of SPBH and LBBH are denoted as $D_1(Q,C)$ and $D_2(Q,C)$ respectively.

Let the distance between the image Q and C be $D(Q,C)$, which is defined as follows:

$$\begin{aligned} D(Q,C) &= k \times D_1(Q,C) + (1-k) \times D_2(Q,C) \qquad (5) \\ &= k \times \sum_{l=0}^{L} \sum_{s=0}^{S} |B^Q(l,s) - B^C(l,s)| + (1-k) \times \sum_{i=0}^{L} \sum_{m=0}^{4^d} |H^Q_i[m] - H^C_i[m]| \end{aligned}$$

In the equation (5), because different images maybe have different values of M_b and L, for convenances of comparison, we choose a value of L which is larger enough than all the other images' values of L. As such, some images may have some bit-planes with no image information. The parameter k is used to control the weights of both indices.

3 Experimental Results and Analysis

In this section, the performance of the proposed indexing scheme is presented and analyzed. An experimental image database which comprises 1460 images downloaded from *http : //www.benchathlon.net* and *http : //sipi.usc.edu* is used to test the proposed approach. For performance evaluation, all images in the database are decomposed to three levels with the lifting factorization of the CDF 9/7 wavelet, and converted into the JPEG2000 format by the EBCOT algorithm. The image data adopt YCbCr color space, and only the Y component is used to evaluate the retrieval performance for it contains the most important image information. Our experiments were executed on an IBM computer with P4 1.5G CPU and 256M memory.

In the experiments, we use the Average Retrieval Ratio (ARR) to investigate the retrieval effectiveness of the proposed method. The ARR is defined as follow:

$$ARR = \frac{1}{N}\sum_{i=0}^{N}\frac{m_i}{N} \tag{6}$$

where N is the total number of one group of similar images, m_i is the number of found relevant images before rank N queried by the i^{th} image belonging to the group.

At first, we want to examine the effect on the both indices by Rotation, Translation and Scale (RTS) transform. We create a set of image by rotation, translation and scale from three original images and 100 references as test. For each of the original images, we get five images by translation ($15°, 30°, 45°, 60°$ and $90°$), five by translation (in 1-5 pixels) and three by scale, totally 14 images. We use each of the 14 images as query image to retrieval. The results is shown in Table.1, from which we can see our proposed index scheme has good ability of anti-RTS, so the effect on the both indices by those three transform in minor disturbance is insignificant. We also can see from the Table.1 that the LBBH

Table 1. The ARR for evaluating the effects on the SPBH and LBBH by the RTS

ARR (%)	SPBH	LBBH	SPBH+LBBH
Image 1	77.2	82.8	93.3
Image 2	61.3	74.1	84.3
Image 3	70.5	79.6	89.7
Average	69.7	78.8	89.1

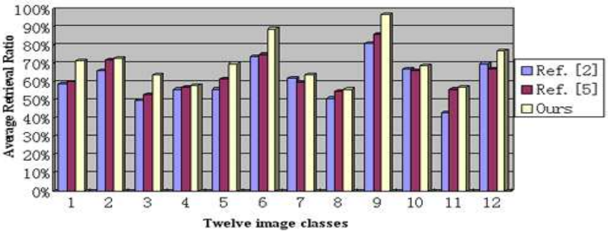


Fig. 3. The ARR of 12 groups of similar images with the methods provided by Ref.[2], Ref.[5] and our proposed

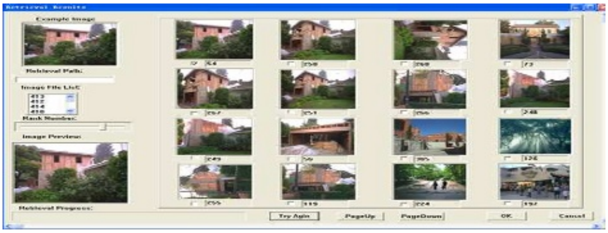


Fig. 4. The first example of image retrieval by our proposed method

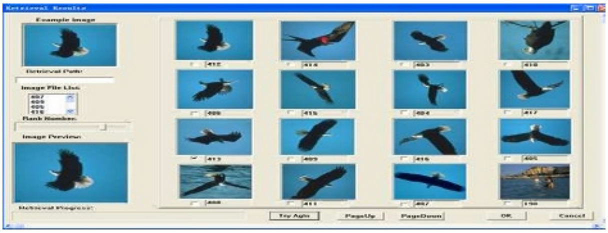


Fig. 5. The second example of image retrieval by our proposed method

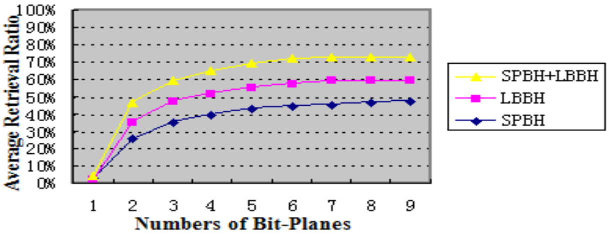


Fig. 6. The experimental results with different numbers of bit-planes

has gained 9.1% higher on average than the SPBH. So in the equation (5), we give more weight to LBBH after the experiment.

Next, in order to evaluate the retrieval performance of our proposed method, 12 classes of similar images including peppers, buildings, fires, airplanes, flowers, animals, birds, toys and scenes and so on are selected from the image database. We use each of one class of images as query image to compute the ARR among the database (1476 images), and we also compare our results with the method of Ref.[3] and the method of Ref.[2]. In the experiment, $L = 9$ and $k = 0.3$. Fig.3 shows the experimental results, from which we can see our proposed method obtained better results than the method provided by [3]. Fig.4 and Fig.5 provides two retrieval examples.

We also want to know the effect on retrieval results by the numbers of bit-planes used to image retrieval. So we choose one of nine classes of similar images and make use of different numbers of bit-planes to build image index and then compute the Average Retrieval Ratio. The Fig.6 gives the experimental results ($L = 9, k = 0.3$) when different number of bit-planes are used. Seeing from the Fig.6, we know that with the increasing of the numbers of bit-planes from MSB to LSB, the ARR is on the rise, but gradually inclines to be stable, which tell us that we can not use all bit-planes but parts of important ones to retrieval, as such retrieval time is saved. We also can see the LBBH index has better ability of characterizing the image content than the SPBH index.

4 Conclusions

In this paper, we provided an approach to compressed image retrieval based on the JPEG2000 framework. Two index schemes were used to retrieval JPEG2000 images. Both of them are extracted in a progressive manner and the extraction procedure can be stopped at any point without touching the remaining bit stream of the compressed image. While our method gains the advantage of eliminating decompression, the experiments also support that the retrieving accuracy is better than the existing counterparts. The proposed method may be used to fast JPEG200 image retrieval from WWW or dynamic image database.

References

1. Skodras A. et. al: The JPEG-2000 Still Image Compression Standard. IEEE Signal processing Magazine. **9**(2001)36C58
2. Liang K.C., Kuo C.C.: Waveguide: a joint wavelet-based image representation and description system. IEEE Trans. Image Processing. **8** (1999)1619C1629
3. Liu C., Mandal M.: Image indexing in the JPEG-2000 framework. Proceedings of SPIE:Internet Multimedia Management Systems, Vol 4210(2000)272C280
4. Chuping Liu and Mandal M.K.: Fast image indexing based on jpeg2000 packet header. Proceedings of 3rd Intl Workshop on Multimedia Information Retrieval(2001)
5. Ziyou Xiong and Thomas S. Huang: Block-based, Memory-efficient JPEG2000 Images Indexing in Compressed-domain. Fifth IEEE Southwest Symposium on Image Analysis and Interpretation(2002)

6. Bhalod J., Fahmy G.F. and Panchanathan S.: Region based indexing in the JPEG-2000 framework. Proceedings of SPIE: Internet Multimedia Management Systems II. Vol **4519**(2001)91C96
7. Jianmin Jiang, Baofeng Guo, Pengjie Li: Extracting Shape Features in JPEG-2000 Compressed images. ADVIS 2002, LNCS, Vol 2457(2002)123C132
8. Lin Ni: A novel image retrieval scheme in jpeg2000 compressed domain vased on tree distance. IEEE, The Fifth International Conference on Information and Communications Security, Singapore (2003)1591-1594
9. Tung-shou Chen, Jeanne Chen and Jian-Guo Chen: Tamper Detection and Retrieval Technique Based on JPEG2000 with LL Subband. IEEE, International Conference onNetwork, Sensing and Control, Taiwan (2004)1235-1241

Target Segmentation and Feature Extraction for Undersea Image Based on Function Transformation

Fuyuan Peng, Yan Tian, Xi Yu, Guohua Xu, and Qian Xia

Dept. of Electronic &Information Eng,
Huazhong University of Science &Technology, 430074 Wuhan, China
pfuyuan@163.com

Abstract. Because of the specialty of undersea channel and the complexity of undersea environment, many uncertain factors affect the quality of undersea image. Consequently, it is a difficult problem to segment and identify targets for undersea image. In this paper, a novel target segmentation and feature extraction approach for undersea image based on function transformation is presented. The approach overcomes the influence of complex environment and uneven illumination effectively. Experimental results demonstrate that the approach is valid for target segmentation and feature extraction for undersea hydrothermal vent image.

1 Introduction

The 21st century features a rapid growth of sea exploitation. The theories and techniques of undersea information processing attract more and more attentions. Because of the effect of light scattering and absorption by water during image acquiring, the undersea image has some defaults such as low contrast, blur edges and feeble textures. All these add difficulties to the automatic recognition and self-determination of undersea robot. Furthermore, undersea hydrothermal vent image mainly consists of water, smoke and rock. The targets themselves are non-structured and the distribution of features is uncertain, which result in the ineffectiveness of image analysis based on structural and common grey features. The study on the approach to undersea image processing is a challenge for conventional algorithms.

In the middle of 1980s, Pal and King applied fuzzy theory to image processing[1]. This approach translates fuzzy issue into exact issue or issue that can be more easily coped with by computers. This approach is widely used in diverse fields [2][3]. However, presently, study of applying fuzzy algorithm to undersea image processing is still in its commencement. Although the present fuzzy algorithm cannot provide ideal processing result for undersea images, it can make the expression of target information more reasonable, and thus the information can be better used. It indicates a novel way to study suitable mathematic model for undersea image.

Considering that the grey character of undersea image is affected by uneven illumination, this paper proposes a segmentation approach based on combination of grey character and chroma character. Considering the poor ability of grey character to dis-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 400 – 406, 2005.

tinguish different targets in undersea images, this paper employs some effective feature extraction algorithms based on models of function transformation, such as models of fuzzy and texture, so that the targets can be recognized easily.

2 Target Segmentation Based on Combination of Characters

Based on the analysis of undersea hydrothermal vent image, we find that for most images with uneven illumination, chroma character has better separability than grey character, while for some other images with even illumination, the situation is just the opposite. So, segmentation using only one kind of character is in-appropriate. In order to enhance the adaptability of algorithm and make it feasible to diverse underwater images, this paper brings forward a segmentation algorithm based on the combination of grey character and chroma character.

2.1 Nonlinear Inversion Algorithm Based on Fuzzy Transformation

Since the distribution of chroma character and grey character are contrary for most undersea images, an inversion transformation is implemented for chroma character before combination. The model of fuzzy transformation function is defined as follows:

$$P(i,j)=\sin(\pi*(1-(\max-B(i,j))/d)/2) \quad (1)$$

B(i,j) represents chroma character of pixel (i,j) in original image . max is the maximum of B(i,j). P(i,j) represents the grade of membership pixel (i,j). d is the parameter of the fuzzy model, and P(i,j) is related to d. If d is lower than max, part of low eigenvalue will be set to zero after fuzzy transformation, so d is often equal to max in the application. This algorithm has poor adaptability due to the fixed parameter. Furthermore, the transformation is increasing by degrees and can not inverse the eigenvalue. So we design the following algorithm:

1. Let certain eigenvalue B(i,j) transform into a sequence $P=\{p_1,p_2,\ldots p_n\}$ according to (1) when d gets different value;
2. Calculate the average value of absolute difference of P as the fuzzy eigenvalue in the transformation domain;
3. Repeat 1) and 2) for each eigenvalue in the original image, then normalize the fuzzy eigenvalue in the transformation domain by their maximum, until we get final fuzzy eigenvalue for each pixel.

The transformation can be described as the curve in Fig. 1:

From Fig.1, it can be seen that the nonlinear transformation inverses the eigenvalue, while at the same time, it compresses the eigenvalue's distribution of both ends. So, the compactness between pixels in both ends is improved, which is benefit for the following processing.

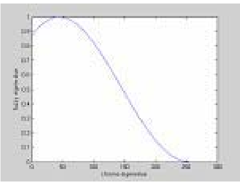


Fig. 1. Nonlinear transformation curve

2.2 Combination Algorithm Based on Chroma and Grey Information

Let I represent original grey image and B be the chroma image after the nonlinear transformation, and M be the combination image. Suppose r represents the weight of I in M $(0 \le r \le 1)$, then 1-r is the weight of B in M. M can be expressed as follows:

$$M=(1-r)*B+r*I \qquad (2)$$

From equation (2), we can see that the separability of M relies on the value of r. In order to optimize the separability, we introduce a method to choose optimal value of r based on OTSU. Considering that uneven illumination has little impact on chroma, we should take B into account firstly in the combination. If r is not equal to zero, the separability will be decreased, in this case, r is set to zero. The algorithm consists of five steps:

1. Extract the chroma information from original image, then employ the nonlinear inversion transformation described in 2.1, the result is read as B;
2. Calculate the maximal standard deviation of B, denote by D0;
3. Let r be larger than zero, calculate the maximal standard deviation of M, denote as D1.If D1< D0, r=0, algorithm is over; otherwise, r gets different value in the increasing order by certain space, compute the maximal standard deviation of M each time, denote as Di.
4. The optimal value of r is the value when Di gets local maximum. If Di is increasing by degrees, r will be set to 1.
5. Compute the combination image according to (2).

This algorithm pays attention to both grey character and chroma character. r is adaptively determined according to certain image, and the separability is optimized. If the image has even illumination, which means that the grey character has better separability than chroma character, then r gets big value. If the condition is opposite, r gets small value accordingly.

2.3 Fuzzy Enhancement Algorithm

Fuzzy enhancement algorithm consists of two steps: the first one is fuzzy transformation using certain membership function; then the second step completes enhancement transformation. In this procedure, membership function plays an important role. One of classic membership function is defined as follows:

$$P(i,j)=[1+(max-X(i,j))/Fd]^{-Fe} \qquad (3)$$

P(i,j) represents the grade of membership of pixel (i,j) after transformation, while X(i,j) represent the eigenvalue of pixel (i,j) in original image. max is the maximum of X(i,j), while Fd and Fe are fuzzy factors, which determine the shape of the transformation curve. The model of fuzzy enhancement is given as follows:

$$P''=T_r(P)=T_1(T_{r-1}(P)),\ r=1,2,\cdots \qquad (4)$$

$P'$is the result after multiple enhancement. r is the time of enhancement. T represents transformation function:

$$T_1(P)=2P^2 \quad , \text{ if } P<0.5$$
$$T_1(P)=1-2(1-P)^2 \quad , \text{if } P>=0.5 \tag{5}$$

The time of enhancement can be set according to the result of enhancement, then after an inverse transformation, an enhanced image can be obtained.

The traditional method to determine Fd and Fe in (4) is not adaptive. An algorithm is designed to decide the optimal parameter adaptively by the criterion of maximal fuzzy entropy in this paper. In the definition of Shanon entropy, the sequence of data is replaced with the sequence of frequency distribution of image in the fuzzy domain, so fuzzy entropy can be described by (6).

$$H=\Sigma(-(1-P_i)\log(1-P_i)-P_i\log(P_i)), \quad i=1,2,\cdots,256 \tag{6}$$

P_i is the normalized sequence of frequency distribution. H is fuzzy entropy. Fuzzy entropy can decide the quantity of information. If the fuzzy entropy after fuzzy transformation with certain parameter is the nearest to that of the original image, the transformation is optimal, and the corresponding parameter is the optimal parameter.

After the transformation of fuzzy enhancement, the area of target is enhanced and the area of background is declined. Thus, the difference between target and background is enhanced which is beneficial for the following segmentation.

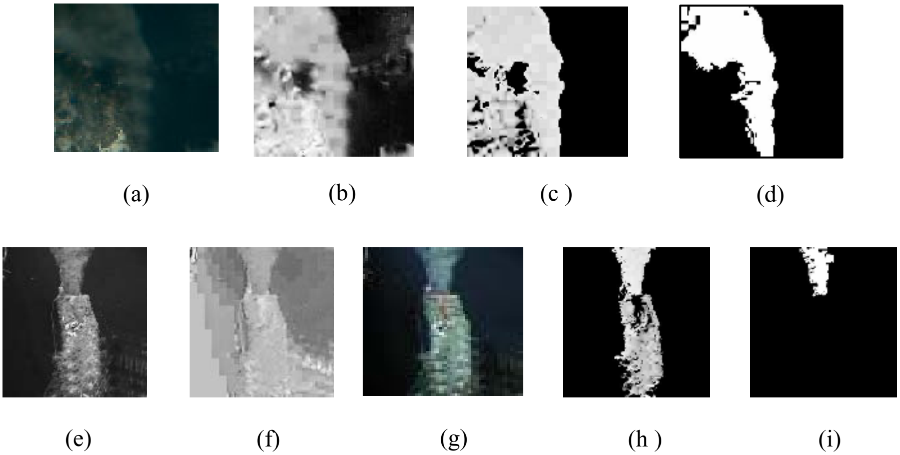
2.4 Target Segmentation and Experiment

Considering the complexity of the undersea hydrothermal vent image, this paper adopts the strategy of laying and partitioning. The algorithm is design as follows:

1. Extract the chroma information from original undersea hydrothermal vent image, then employ the nonlinear inversion transformation described in 2.1;
2. Information combination by way of 2.2;
3. Segment the image using optimal threshold and thus get rid of the region of water;
4. Extract the chroma information of remained area after step 3, then execute fuzzy enhancement to enlarge the difference between chimney and rock;
5. Segment the image using optimal threshold, the area of chimney will be defined.

In order to validate the algorithm above, we chose two representational undersea hydrothermal vent images, shown in Fig.3(a) and Fig.3(e).

As shown in Fig.3(a), the images have poor illumination and unclear transition between species in grey image. While on the other hand, the separability of chroma between species is good. If grey information is added to chroma information, the separability will be tampered. So, when computing the value of r by (2), r will be set to zero. The combination result is shown as Fig.3 (b). Then, after step (3), the result of step (4) is shown as Fig.3(c). The compactness of chimney is improved. The final area of chimney is presented in Fig.3(d). It is apparent that chimney area is integrated. In Fig.3(e), on the contrary, illumination is even. The chroma image is shown as Fig.3(f). The separability of grey character is obviously better than that of chroma character, when we compute the value of r by (2), r will be set to 1. The combination result is shown as Fig.3(g). Then, after step 3), the result of step (4) is shown as Fig.3(h), the compactness of chimney is improved,. The final area of chimney is presented in Fig.3(i), it is apparent that segmented chimney area is ideal.



(a) (b) (c) (d)

(e) (f) (g) (h) (i)

Fig. 2. (a)Original image with poor illumination,(b) combination image of (a), (c)fuzzy enhancement image of (a), (d) binary image of chimney in (a), (e) original image with even illumination, (f) chorma image of (e), (g) combination image of (e), (h)fuzzy enhancement image of (e), (i) binary image of chimney in (e)

3 Feature Extraction Based on Function Transformation

In section 2, the target is segmented. In order to determine the target is just what we want, a pattern recognition procedure is needed. The key of pattern recognition is to extract appropriate features. Since features in transformation domain is often notable and suitable for further analysis of image that is bad imaged and lack of redundant information, we designed some feature extraction methods based on function transformation for undersea image.

3.1 Feature Extraction Base on Fuzzy Transformation

The model of fuzzy transformation function is the same as (1). After transformation, the fuzzy rate of image is related to d. If d changes, the fuzzy rate will be different. Furthermore, the sensitivity of fuzzy rate is connected to the histogram distribution of image. So a new approach to extract fuzzy feature is presented: Let d in (1) increase evenly, when d gets a value, a fuzzy rate will be computed accordingly, then a sequence of fuzzy rate P={p_1,p_2,…p_n}is obtained. The average of absolute difference of P is right the fuzzy feature.

In simulation experiment, the mean values of the fuzzy features for each species are calculated and shown in Table 1.

Table 1. fuzzy feature

	Water	Chimney	Rock
Fuzzy feature	0.02	0.005	0.003

3.2 Feature Extraction Based on Semi-variogram

Semi-Variogram reflects the structural property of image, and it is an effective tool to describe the texture of image. The definition is represented in (7):

$$\gamma(h)=(\Sigma[D(x_i+h)-D(x_i)]^2)/(2N(h)),i=1,2,\ldots N(h) \tag{7}$$

N(h) is the number of pixel pair and the space between them is h, D(.) represents grey scale of pixel (x_i+h) and x_i. γ (h) is the Semi-Variogram at the scale of h.

Based on experiment and analysis, we find that the texture of chimney and water is exquisite and the correlation between pixels is high, while the texture of rock is rough and the correlation between pixels is small. So a simple method is presented, that is to let h be 1 and calculate semi-variogram vertically.

Through simulation, the mean values of the semi-variogram features for each kind are figured out and shown in Table 2.

Table 2. semi-variogram with h=1

	Water	Chimney	Rock
Semi-variogram(h=1)	5.1355	7.3132	118.339

4 Conclusions

Due to many factors such as the effect of light scattering and absorption by water during image acquiring, in general case, segmentation and recognition for undersea images is a difficult problem. In this paper, an image segmentation method based on information combination and feature extraction methods based on function transformation are proposed. Experimental results show the effectiveness of the proposed methods.

Acknowledgements

This paper is supported by the National High Technology Research and Development Program of China and the National Science Foundation Project (60475024).

References

1. Pal S. K., King R. A. Image enhancement using fuzzy sets. Electronics Letters, 1980,16(10),P:376-378
2. M.A.D. Wirth，J. Lyon and D. Nikitenko. A Fuzzy Approach to Segmenting the Breast Region in Mammograms. Fuzzy Information. Processing NAFIPS'04. V:1, 27-30 June 2004,P:474-479
3. Ioannis K. Vlachos and George D. Sergiadis. Fuzzy reasoning scheme for edge detection using local edge information based on Renyi's entropy. Signal Processing and Its Application.V:1, 1-4July 2003,P:549-552

ART in Image Reconstruction with Narrow Fan-Beam Based on Data Mining

Zhong Qu[1,2], Junhao Wen[2,3], Dan Yang[2,3], Ling Xu[3], and Yu Wu[1]

[1] College of Computer Science and Technology,
Chongqing University of Post and Telecommunication,
400065 Chongqing, China
quzhong@cqupt.edu.cn, cq789@hotmail.com
[2] College of Computer Science, Chongqing University,
400030 Chongqing, China
quzhong@hotmail.com,
jhwen@cqu.edu.cn, dyang@cqu.edu.cn
[3] School of Software Engineering,
Chongqing University, 400030 Chongqing, China
xuling@cqu.edu.cn

Abstract. Image reconstruction is one of the key technologies of industrial computed tomography. Algebraic method has un-replaceable advantage when the data is incomplete or the noise effect is high because of data mining. However the use of algebraic method has been highly limited because of the low speed reconstruction. In this paper, a new iterative method (algorithm reconstruction technique) is introduced to accelerate the iteration process and increase the reconstruction speed. Besides, algebraic reconstruction method will be used more widely with the development of computer technology and increase of computer speed. Experiment results clearly demonstrate that algorithm reconstruction technique can efficiently improve quality of images reconstruction when processing the incomplete projection data or noisy projection data based on data mining.

1 Introduction

Industrial Computed Tomography (ICT) [1], has been widely used in nondestructive testing (NDT) and nondestructive evaluation (NDE) areas. ICT can give the cross section image of the inspected object: the detailed spatial position; shape and size of target particular can be viewed directly from the image. The interested part will not be blocked by the particulars around it and the image is easy to be distinguished and understood. Image reconstruction is one of the key technologies of ICT. The quality of reconstructed image by ICT is decided by movement control, data acquisition and the algorithm and so on [1].

The quality of reconstructed image by algebraic method is at less equal to the convention back projection method [1],[2] when the data is complete. But algebraic method has un-replaceable advantage when the data is incomplete or the noise effect is high [3]. However the use of algebraic method has been highly limited because of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 407–414, 2005.

the low speed reconstruction. In this paper, a new iterative method is introduced to accelerate the iteration process [4],[5],[6] and increase the reconstruction speed. Besides, algebraic reconstruction method [4],[5],[6] will be used more widely with the development of computer technology and increase of computer speed.

2 The Description of Image Reconstruction Algorithm

In order to acquire the data for cross section image, the inspected object must be scanned. The narrow fan-beam scan model uses single radiation active source, small angle, fan shape and multiple detectors. A frame of image contains the descriptive information for the objectively existed substance it describes [7],[8],[9]. Usually image is assumed has following properties:

(1) Image area is a square; its center coincides to the origin of coordinate.

(2) Image can be expressed by 2D function $f(x, y)$, its value is zero when it is outside of image area. (x, y) is the coordinate of space point. The value of $f(x, y)$ at any point is proportional to the light intensity (grayness level) of that point.

(3) Function $f(x, y)$ of image is non-negative and limited, expressed as $0 \leq f(x, y) \leq L$.

In right angle coordinate system, if image f is function of variable x, y, then the value of f at point (x, y) is called the density of image at point (x, y).

A frame of digital image is a discrete image of $f(x, y)$ both in 2D coordinate and intensity. It can be treated as a matrix; the row and column indicate the position in 2D space for every point in image. And the value of matrix element indicates the grayness level of every point. In general, the image area can be divided into $n \times n$ squares and every square is a pixel.

3 The Theory of ART Algorithm

An $n \times n$ square grid is added onto the image $f(r, \Phi)$ which need be reconstructed. Then divide it into $J = n^2$ small squares. The discrete values are expressed by $f(r, \Phi)$, which is uniform within the pixel.

A ray can be treated as a line with width τ. It covers partial area of each pixel. The fraction of area multiples the pixel value x is its contribution to the ray projection. Ray integral is called ray sum here. The value (grayness or density) of pixel j is x_j the overlapped area between ray i and pixel j is the shadow area. Its ratio to pixel area is $r_{ij} = S_{shadowed\ area} / \delta^2$ .So the contribution of pixel j to ray i is $p_{ij} = r_{ij} x_j$.

Ray i also intersects with other pixels, the sum of ray projection is $p_i = \sum_1^N p_{ij} = \sum_1^N r_{ij} x_j$.

But in reality, in order to reduce the computation, one level approximation is usually used to simplify r_{ij}: assuming pixel value is at the center. r_{ij} is 1 or 0 decided by if ray i is passes the center of pixel j, expressed as $r_{ij} = \begin{cases} 1 & ray\ i\ passes\ the\ center\ of\ pixel\ j \\ 0 & others \end{cases}$

The third definition of r_{ij} is similar as the second one: the width of all rays is 0, but the distance between rays is τ.

The ray width τ and the definition of r_{ij} will affect the value of sum in $p_i = \sum_1^N p_{ij} = \sum_1^N r_{ij} x_j$, therefore affect the accuracy of the reconstructed image. Weighted factor r_{ij} is decide ed by geometric angle and position using whichever calculation method. It can be expressed in matrix $p = Rx$.

p is *J* dimensional vector, called measured vector, *x* is *I* dimensional vector, called image vector. Our duty is to get *x* according to measured p and matrix *R*.

4 The Implementation of ART Algorithm

Equation $p = Rx$ has a following correction $p = Rx + e$. *e* is a random vector. If *e* is omitted, it will not be accord with reality and cause no solution for the equation.

What the authors worked on is belong to the latter and this kind of algorithm is called ART. It converts solving equation $p = Rx + e$ to solve non-equal equation: $Rx \leq p$ which can be expanded by relaxation method used for solving linear non-equal equation: $r_i^T \leq p_i$, *i*=1,2,3,4...*I*. The solution is:

$$\begin{aligned} &X(0) && ,anything \\ &x^{(k+1)} = \begin{cases} x^{(k)} & ,r_{ij}^T x^{(k)} \leq p_{ik} \\ x^{(k)} + \lambda^{(k)} \times \frac{p_{ik} - r_{ik}^T x^{(k)}}{\|r_{ik}\|^2} r_{ik} & ,elsewhere \end{cases} \end{aligned} \tag{1}$$

$$i_k = k(\mathrm{mod})I + 1 = \left[k - Int\left(\frac{k}{I}\right) + 1 \right] \tag{2}$$

In the equations above, Int represents integer, $\lambda^{(k)}$ is relaxation factor. It can be proved: when $0 < \varepsilon_1 \leq \lambda^{(k)} \leq \varepsilon_2 < 2$, the list $x^{(0)}$, $x^{(1)}$, $x^{(2)}$...resulted from relaxation method will be convergent to a vector in *R* (assuming *R* is not empty).

Now let us look at the collection meaning using relaxation method to solve non-equal equation: $H_i = \{x \mid r_i^T \approx p_i\}$

H_i is the vector collection that satisfies the number i equation in (1). Obviously H_i is the sub collection of *R*. Every H_i represents a hyper plane. If the dimensional number *J* of *x* is 3, the hyper plane is a plane in a three dimensional space. If *J* = *2*, the hyper plane is a line in a 2D space (a plane).

The second part(correction part) of right side of equal in equation(1) can be modified as: $\frac{p_{ik} - r_{ik}^T x^{(k)}}{\|r_{ik}\|} = \frac{p_{ik} - r_{ik}^T x^{(k)}}{\|r_{ik}\|} \times \frac{r_{ik}}{\|r_{ik}\|}$

The format of addition ART algorithm is $x^{(k+I)} = x^{(k)} + \Delta x^{(k)}$.

The format of multiplication ART algorithm is

$$\begin{cases} x^{(0)} \quad \textit{initial vector}, J\ \mathrm{dim}\textit{ensional}, \textit{normally every element is } 1 \\ x_j^{(k+1)} = \left(\dfrac{p_{ik}}{r_{ik}} \right)^{\lambda^{(k)} r_{ik,j}} \qquad j = 1,2,3,\cdots,J \end{cases}$$

$i_k = k(\mathrm{mod})I + 1$, $\lambda^{(\mathrm{k})}$: relaxation factor $0 < \varepsilon \leq \lambda^{(k)} \leq 1$; $x_j > 0$: an element in image vector *x*; $r_{ik,j}$: 0 or 1;

It can be proved: the sequence $x^{(0)}$, $x^{(1)}$, $x^{(2)}$… from equation(1) is convergent to the maximum entropy of $p = Rx$.

The implementation of ART image reconstruction can be summarized as:

(1) First set the initial value $x^{(0)}$ of the reconstructed image, usually every value of pixel is set to 0 in addition ART algorithm if knowing nothing about it in advance; every value of pixel is set to 1 in multiplication ART algorithm and iterative convergent coefficient $\lambda^{(k)}$ is selected.

(2) Correct the grayness value for every pixel using $Rx \leq p$.

(3) Repeat step (2), use every projection orderly to correct image until the number I equation. Then one cycle of iterative calculation is finished.

(4)Decide if it reaches the pre-select iterative times, if yes then the iterative process is stopped. Otherwise re-calculate i_k , go back to step (2) and enter next iterative cycle.

5 Programming for ART Algorithm

This system provides very good user graphic interface with data acquisition, noise simulation, algorithm selection, image reconstruction and whole process of result display and analysis. The framework of whole image reconstruction process is showed in the picture - see Fig. 1.

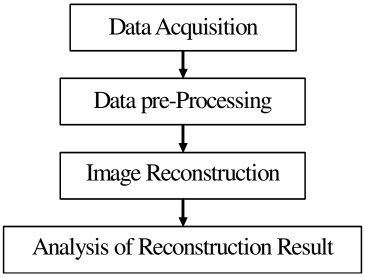


Fig. 1. Fundamental Sketch map of Image Reconstruction System

During experiments of image reconstruction, the inspected models used for image reconstruction were provided in the bitmap (BMP) format. The reconstructed results were also displayed and saved in bitmap format. Therefore bitmap format will make program more compatible. At the same time because bitmap format is uncompressed image, it is easier to get simulated scanning data. Besides, Windows provides SDK package supporting bitmap operations, which makes software development easier.

BMP file consists in four parts. The first part is Bimapfileheader, a structure that contains the information about the type of BMP file, size and print format. The size of this structure is constant, 14 bytes, the definition is as following:

```
typedef struct tagBITMAPFILEHEADER
{ WORD    bfType;  //the type of BMP file, must be0x424D, which is string BM
  DWORD bfSize;   //the size of BMP file, including this 14 bytes
  WORD    bfReserved1; //reserved, must be 0
  WORD    bfReserved2; //reserved, must be 0
  DWORD bfOffBits;  /*the number of offset bytes from head of file to the actual
BMP data, same as the sum of  size of bitmapfileheader, bitmapinfoheader and palette*/
}BITMAPFILEHEADER;
```

The second part is Bitmapinfoheader, which contains the information about the size and color of BMP file. The size of this structure is also constant, 40 bytes, and the definition is as following:

```
typedef    struct   tagBITMAPINFOHEADER
{ DWORD    biSize;       //the size of this structure, it is 40
  LONG      biWidth;      //the width of bitmap, using pixel as unit
  LONG      biHeight;      //the height of bitmap, using pixel as unit
  WORD     biplanes;      //the level of target equipment, must be 1
  WORD     biBitCount;  /* the bit number of pixel, must be 1(monochrome), 4(16
color), 8(256 color) or 24(true color)*/
  DWORD   biCompression; /*the compression type, must be 0(non-compression),
1(BI_RLE8 compression) or 2(BI_RLE8)*/
  DWORD   biSizeImage;    //the size of bitmap, using byte as unit
  LONG   biXPelsPerMeter;  /* the horizontal resolution of target equipment, unit
is pixel number per meter*/
  LONG   biYPelsPerMeter; /*/ the vertical resolution of target equipment, unit is
pixel number per meter*/
  DWORD  biClrUsed;    /*the number of color actually used by bitmap, if it is 0,
then it is bitBitCount power of 2*/
  DWORD   biClrImportant;  /*the number of important color for displayed bitmap, if it is 0, then every color is important*/
}BITMAPINFOHEADER;
```

The third part is palette, of course it is for those bitmaps that need palette. Some bitmaps, such as true color, don't need palette and BITMAPINFOHEADER is followed directly by bitmap data. Palette is actually an array, which has totally biClrused elements(if it is 0, then the used color number has $2^{\text{biBitcopunt}}$ elements). The type of every element in array is a RGBQUAD structure, the definition of RGBQUAD with 4 byte size is as following:

```
typedef    tagRGBQUAD
{ BYTE   rgbBlue;          //the lightness of blue
  BYTE   rgbGreen;         //the lightness of green
  BYTE   rgbRed;           //the lightness of red
  BYTE   rgbReserved;    //reserved, must be 0
}RGBQUAD
```

The forth part is the actual image data. For a frame of gray image, rgbBlue, rgbGreen and rgbRed are equal. When generating bitmap file, Windows scan bitmap row by row from left-down corner (from left to right then from down to up), the last element is at the right-upper corner. The bytes of these pixels compose the bitmap.

The authors will use 24 byte bitmap and 256 color bitmap in the reconstruction experiments. 24 byte bitmap doesn't have palette, every pixel maps to 3 bytes, representing the intensity of red, green and blue respectively. In gray image, the R, G and B values are same for every pixel. 256 color bitmap has palette, its actual bitmap data is the grayness value of pixel. Calculating projection data is to calculate the sum of the sum of grayness of pixels passed through by every ray, which is the linear integration of pixels. This is different from data acquisition in ICT, where the data is the left energy value after every ray passes through target. When reconstructing image, the linear integration can be got from the difference of incident energy and that data.

6 The Result of ART Algorithm

The advantages of these methods are more objective and easy to measure and compare. The normal quantitative evaluation criteria have following: Correlation coefficient (e), Normalized mean square error (d), Normalized mean absolute error (r).In the experiment, the number of detector is 129, the open angle of detector array is 12°, the average movement distance is 63, paralleling movement number is 6, rotation angle is 15, the distance between origin paralleling movement and scanning center is 600. The convergent factors for ART are 0.1, 0.5 and 1. The square grid is 256×256 pixels.

(1)The reconstructed image from original collected data

Reconstruction quality contrast with complete projection data after one times iteration by ART algorithm is shown in the following table.

Table 1. Reconstruction qualities with complete projection data (Iterative times is 1)

Algorithm Quality criteria	ART		
	$\lambda^1 = 1$	$\lambda^2 = 0.5$	$\lambda^3 = 0.1$
e	0.877296	0.883778	0.795250
d	0.482312	0.475307	0.648877
r	0.387806	0.382970	0.595144

Table 2. Reconstruction qualities with complete projection data (Iterative times is 2)

Algorithm Quality criteria	ART		
	$\lambda^1 = 1$	$\Lambda^2 = 0.5$	$\lambda^3 = 0.1$
E	0.896507	0.901682	0.841499
D	0.445105	0.436635	0.563370
R	0.353301	0.346497	0.486550

From the two tables above, it can be seen that: when iterative times is small, the image quality is worse as λ is close0r to both side of (0 < λ < 2); but image is much smoother when λ is small (λ ≤ 0.5) than it when λ is bigger (λ > 0.8).

When λ is big (λ > 0.5), the major difference between reconstructed image and original model lies on big errors of a few elements. Therefore black or white spots will occur in the image, which is called salt and pepper phenomena. When λ is small, the major difference is the small errors of many elements. Thus the image is much smoother than it when λ is bigger. The reconstructed image is more satisfactory when λ is small (0.1 < λ < 0.8) and iterative times are between 5 to 8 – see Table 3.

Table 3. Quality contrast with different constringent through complete projection data of ART

Quality criteria / Iterative Times	e		d		R	
	$\lambda^1 = 0.1$	$\lambda^2 = 0.5$	$\lambda^1 = 0.1$	$\lambda^2 = 0.5$	$\lambda^1 = 0.1$	$\lambda^2 = 0.5$
1	0.795250	0.883778	0.648877	0.475350	0.595144	0.382970
2	0.841499	0.901682	0.563370	0.436635	0.486550	0.346497
3	0.860098	0.907530	0.520704	0.423393	0.434507	0.334191
4	0.876301	0.910637	0.494428	0.416382	0.404486	0.327898
5	0.884815	0.912603	0.476505	0.411956	0.385265	0.323987
6	0.890807	0.910991	0.463548	0.408827	0.371890	0.321223
7	0.895258	0.915029	0.453721	0.406486	0.362144	0.319122
8	0.898682	0.915841	0.446037	0.404653	0.354787	0.317539

From table 3, it is shown that the improvement of image quality is not very obvious when the iterative time is more than 5. So it will satisfy the reconstruction requirement when iterative time ranges from 4 to 6 for normal image reconstructions.

(1)The image reconstruction with noise

1) Adding 10% multiplication type noise (the iterative times is 5 in ART)

Table 4. Reconstruction quality contrast with contaminated projection data

Algorithm / Quality criteria	ART		
	$\lambda^1 = 1$	$\lambda^2 = 0.5$	$\lambda^3 = 0.1$
e	0.819228	0.868399	0.954012
d	0.635753	0.533094	0.311433
r	0.537781	0.451402	0.265995

2) Adding 10% addition type noise (the iterative times is 5 in ART)

Table 5. Reconstruction quality contrast with contaminated projection data

Algorithm Quality criteria	ART		
	$\lambda^1 = 1$	$\lambda^2 = 0.5$	$\lambda^3 = 0.1$
e	0.797514	0.836868	0.946164
d	0.673941	0.594517	0.328575
r	0.603448	0.526129	0.288995

7 Conclusion

From the analysis in this paper, it can be concluded that the image reconstruction can be improved by adjusting the convergent factor (λ) in ART. When the projection data have no noise, the image quality will be better if λ is bigger (λ=0.5); when the projection data have noise, the image quality will be better if λ is smaller (λ=0.1).

ART algorithm has good performance even with noise. High quality image can be reconstructed after 4~6 iterations. When the projection data have little noise, the reconstruction speed can be improved as λ is bigger (λ=0.5). When the projection data have noise, the image quality can be better as λ is smaller (λ=0.1).

References

1. Zhong Qu. Research on Multi-Channels' Data Acquisition and Storage System for Industrial Computed Tomography Based on CPLD Technology [Ms.D. Thesis]. Chongqing University, (2003) (in Chinese with English abstract)
2. Li JG, Si PF. Image Processing [M]. Shanghai Jiaotong University, (1990) (in Chinese)
3. MilanSonka,Vaclav Hlavac,Roger Boyle. Image Processing,Analysis,and Machine Vision (Second Edition) [M]. Thomson Brooks/cole, People's Post and Telecom Press. (2002)
4. G.T.Herman. Algebraic Reconstruction Techniques Can be Made Computationally Efficient [J]. IEEE Trans Med Image. Vol. 12 (1993) 600~611.
5. G.T.Herman. ART: Mathematics and Applications [J]. A Report on the Applicability to Real Data of Algebraic Reconstruction Techniques. J.Theo.Biol,42 (1973) 1~32.
6. H. Malcon Hudson, Richard S. Larkin. Accelerated Image Reconstruction Using Ordered Subsets of Projection Data [J]. IEEE Trans Med Image. Vol.3 (1994) 581~609.
7. Calvin A. Johnson. A Parallel-Processing Solution For Iterative Image Reconstruction Algorithms [J]. Phy. Med. Biol, Vol.39 (1996) 563~574.
8. S. W. Rowland. Computer implementation of image reconstruction formulas. Image Reconstruction from Projections: Implementation and Applications, G. T. Herman Ed. Berlin, Germany: Springer-Verlag, (1979) 9–70.
9. M. Unser, P. Th´evenaz, and L. Yaroslavsky. Convolution-based interpolation for fast, high-quality rotation of images [J]. IEEE Trans. Image Processing, vol. 4 (1995) 1371–1381.

Digits Speech Recognition Based on Geometrical Learning

Wenming Cao [1,2], Xiaoxia Pan[1], Shoujue Wang [2] , and Jing Hu[1]

[1] Institute of Intelligent Information System, Information College,
Zhejiang University of Technology, Hangzhou 310032, China
[2] Institute of Semiconductors, Chinese Academy of Science,
Beijing 100083, China
csann@zjut.edu.cn

Abstract. We investigate the use of independent component analysis (ICA) for speech feature extraction in digits speech recognition systems.We observe that this may be true for a recognition tasks based on geometrical learning with little training data. In contrast to image processing, phase information is not essential for digits speech recognition. We therefore propose a new scheme that shows how the phase sensitivity can be removed by using an analytical description of the ICA-adapted basis functions via the Hilbert transform. Furthermore, since the basis functions are not shift invariant, we extend the method to include a frequency-based ICA stage that removes redundant time shift information. The digits speech recognition results show promising accuracy, Experiments show method based on ICA and geometrical learning outperforms HMM in different number of train samples.

1 Introduction

For audio signals, Bell and Sejnowski [1] proposed ICA to learn features for certain audio signals. Our goal was to investigate this approach without any constraint setting and provide new analysis and options to cope with the main problems of the standard ICA features, namely in providing features that are phase insensitive and time-shift invariant. In this paper, we apply ICA to speech signals in order to analyze its intrinsic characteristics and to obtain a new set of features for automatic digits speech recognition tasks. Although we would like to ideally obtain features in a complete unsupervised manner since the ICA is a data-driven method. At last, digits speech recognition based on Geometrical learning [2][3][4][5][6] is used. Compared with the conventional HMM-based method, ICA and Geometrical learning (ICA-GL)-based method mentioned in this paper has a good advantage when the number of training samples is very few. The trend of recognition results shows that the difference of recognition rates between these two methods decreases as the number of training increases but the recognition rate of ICA-GL -based method is always higher than that of HMM-based. And both of these recognition rates will reach 100% if there are enough training samples. The recognition accuracy of HMM-based method is lower than that of GL-based method. It is because that ICA-GL -based method can describe

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 415–422, 2005.

the morphological distribution of the speeches in GL however HMM-based method can only calculate the probability distribution of them.

2 Speech Learning Algorithm Based on ICA- Geometrical Learning

2.1 The Collection and Establishment of the Speech Database

There are two speech databases. One is the spontaneous speech in daily life that has no special preparation in terms of speech pattern. It is always slack and goes with random events (filled pauses etc.). The other is the reading speech database. Its speech pattern and speech context should be prepared beforehand and accorded with grammar as well.

The continuous speech database we adopted in this paper is between those two above-mentioned speech databases. Phone numbers is the context of our database. The read pattern is similar to the spontaneous speech that has some background noise, e.g., stir of cars on road.

Segment the continuous speech into syllables by hand and then select the better result as "the learning database". Here we must point out that these syllable samples are different from the isolated samples, which have characteristics of the continuous speech.

Finally, these samples are classified into 11 classes according to their pronunciation in Chinese and phoneticized as table 1.

Table 1. The digits classification of pronunciation in Simplex Chinese

Digit	0	1		2	3	4	5	6	7	8	9
Pronunciation in Simplex Chinese	ling	yi	yao	er	san	si	wu	liu	qi	ba	jiu

These databases are collected in 8000Hz (Sampling Frequency) and 16bits (Bit Depth).

2.2 Proposed Method

Phase sensitivity and time variance seemed to be the most profound factors prohibiting the use of the ICA-adapted basis functions for speech recognition tasks. We took log of the obtained coefficients. The coefficients show large correlation because ICA was not learned to optimize the independence of magnitude coefficients. Therefore, we apply an additional ICA transformation for the log spectral coefficients to obtain as independent coefficients as possible. The mel filter and log operation used in the conventional feature extraction were applied to the ICA coefficients in order to reflect the human speech perception characteristics. Fig 1. compares feature extraction methods using MFCC and ICA. The ICA in the time domain in the proposed method replaces the FFT of the MFCC-based method.

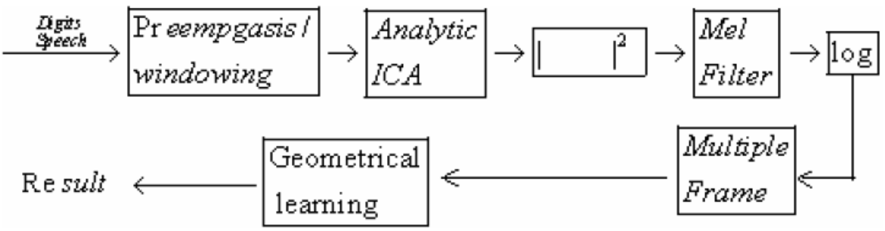


Fig. 1. In ICA-based feature method, speech signals are filtered by analytic ICA filters, the coefficients are taken log of squared magnitude split into mel frequency bands, multiple frames are concatenated

2.2.1 Geometrical Learning

Now, the task of geometrical learning is to cover a given sample set by a chain of hyper sausage units with a minimum sum of volumes via determining the end points of each line segment and the radius of each hyper-sphere. As introduced in section 2.2 the main idea is similar to finding the center and radius of the outer hyper-sphere via successive projection from a higher dimensional space to a lower dimensional space in help of descriptive high dimensional geometry.

2.2.2 Hyper Sausage Neuron

Use of only one hyper-sphere applies to only samples distributed isotropically in all the directions. Also, one sphere that covers a finite size of samples may contain hollows with a huge sum of volumes and thus is actually not a compact representation to a given sample set. To improve, one possible extension is to use multiple smaller hyper-spheres to cover a set of N samples. In fact, only one hyper-sphere for all the samples is one extreme case, while one hyper-sphere per sample is another extreme. Between the two extremes, we can also use k hyper-spheres jointly to cover all the samples. However, among samples from a same pattern class there is usually certain topological connectivity. Dividing a sample set into N hyper-spheres separately will lose the connectivity totally, while dividing a sample set into k<N hyper-spheres will at least lost the connectivity partially. Moreover, there may also be possibilities that the volumes of one or more hyper-spheres are still too large.

Among possible types of connectivity, we believe that one dimensional connectivity takes a major role, especially for a finite size of samples in a high dimensional space. Instead of using hyper-spheres, we consider a manifold resulted from a product of a hyper-sphere and a one dimensional continuous curve, in a sense

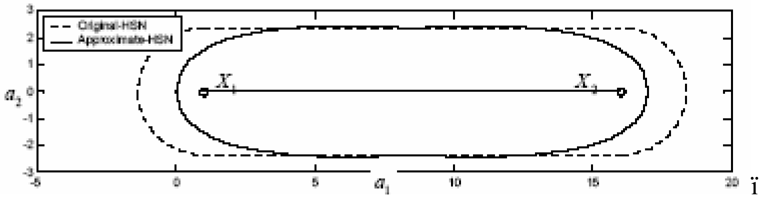


Fig. 2. Original HSN and Approximate HSN in same two-dimensional section

that the manifold is generated by rolling this hyper-sphere with its center moving along the one dimensional curve. To facilitate implementation, the one dimensional curve is further approximated by a chain of straight-line segments. Rolling one hypersphere with its center moving along one line segment, we get one sausage like shape as a basic geometrical unit, as illustrated in Fig.2. For simplicity, we call such a unit Hyper Sausage Neuron (HSN) [3]. Each pair of neighbor line segments is located such that their corresponding HSN units are connected.

2.2.3 Geometrical Learning of HSN Chain

To simplify implementation, the HSN shape [9] is approximated by the shape (in solid line) that can be computed be the following characteristic function:

$$f_{HSN}(X) = \operatorname{sgn}\left[2^{\frac{d^2(X.\overline{X_1X_2})}{r^2}} - 0.5 \right] \tag{1}$$

which contains a radius parameter r and the distance between X and the line segment $\overline{X_1X_2}$ as follows:

$$d^2(X,\overline{X_1X_2}) = \begin{cases} \|X - X_1\|^2, q(X,X_1,X_2) < 0 \\ \|X - X_1\|^2, q(X,X_1,X_2) > \|X_1 - X_2\| \\ \|X - X_1\|^2 - q^2(X,X_1,X_2), otherwise \end{cases} \tag{2}$$

where $q(X,X_1,X_2) = (X - X_1)\cdot\frac{(X_1 - X_2)}{\|X_1 - X_2\|}$,Given a ordered set of samples $P = \{x_i\}_{i=j}^{n}$. The set is sampled in a certain order, which obeys the rule that the mid sample is more like the anterior sample than the latter one. This assures that the set of the samples is a continuously mutative chain. We select a parameter *D*, the distance between the two contiguous selected samples in S . This parameter determines the total of the HSN neurons. From P we choose a set $S\{s_i \mid d(s_{i+1}, s_i) \approx D, 1 \le i < m\}$ of n_j sample support points as the sausage parameters $\{X_{j1}, X_{j2}\}_{j=i}^{n}$ defined by (2) such that all the HSN units become overlapped, in help of the following algorithm:

Let S denote the filtered set that contains the samples which determine the network and X denote the original set that contains all the samples sampled in the order.

Begin

1) Put the first sample into the result set S and let it be the fiducial sample s_b, and the distance between the others and it will be compared. Set S ={ s_b }. $s_{\max} = s_b$ and $d_{\max} = 0$

2) If no sample in the original set X ,stop filtering. Otherwise , check the next sample in X , then compute its distance to s_b ,i.e., $d = \|s - s_b\|$.
3) If $d > d_{\max}$,goto step 6. Otherwise continue to step 4.
4) If $d < \varepsilon$,set $s_{\max} = s$, $d_{\max} = d$, goto step 2. Otherwise continue to step 5.
5) Put s into the result set: $S = S \cup \{s\}$,and let $s_b = s$, $s_{\max} = s$, and $d_{\max} = d$. Then go to step 2.
6) If $d_{\max} - d > \varepsilon_2$, go to step 2. Otherwise put $s_{\max}$ into the result set: $S = S \cup \{s_{\max}\}$,and let $s_b = s_{\max}$, $d_{\max} = \|s - s_{\max}\|$ go to step2.

3 Analysis of ICA Basis Functions

Fig.3 shows the basis functions sorted by the L2 norm and the corresponding frequency responses when the frame size is 128 ms and the number of sources is 67. The basis functions show a ICA waveform. To obtain these basis functions, we updated the ICA filter matrices every 1000 frames, with the convergence factor ε linearly decreasing from 0.0001 to 0.000033.

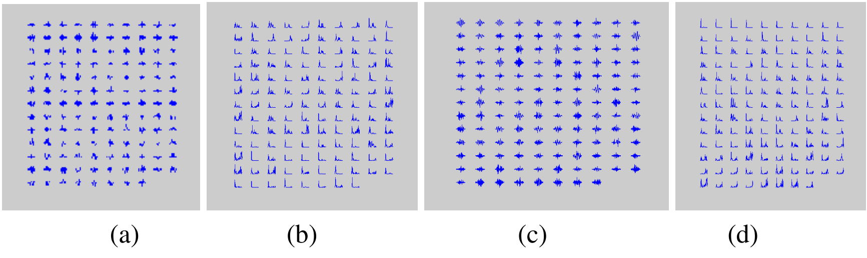


Fig. 3. Basis functions (a) and their frequency response (without sort order)(b) of ICA in the time domain(without sort order). (c) and their frequency response (b) of ICA in the time domain

4 Experiment and Analysis

This library includes 24 persons' 2640 MFCC samples of single digital syllable. They are divided into 5 groups (the table 2) according to different sample figures.

There are 29 persons' totally 7308 single digital syllable samples used to test the correctness of modeling. Construct each group's single digital syllable samples' HMM model. Use the HMM model that is from left to right without jumping. How many of hybrid figures in the gauss probability density function has huge influence on the recognition. Adjust the state figures and hybrid figures in the gauss probability density function to let the test set get the recognition rate as high as possible. Through the repeating tests, each groups' states and hybrid figures in the gauss probability density function are displayed in table 3 and each groups' results of recognition are displayed in Figs. 4-7.

Use the same samples to construct 11 classes' digital ICA-GL. Compared with the conventional HMM-based method, The trend of recognition results shows that the difference of recognition rates between these two methods decreases as the number of training increases.

Table 2. the modeling samples of different figures

Class Sample distribution	1	2	3	4	5	6
Each person's sample figure of each class	**1**	**2**	**4**	**5**	**7**	10
The figure of each class's sample point	24	48	96	120	168	240

Table 3. the HMM state figures and hybrid figures in the gauss probability density function

The sample points' quantity of each class		10	20	40
HMM	State figures	4	4	5
	Hybrid figures in the gauss probability density function	2	4	3

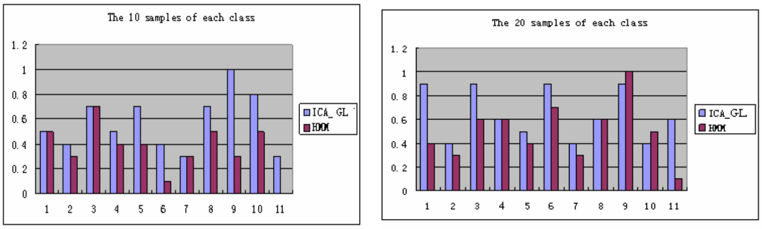


Fig. 4. (Left) Result compare with ICA_GL and HMM in the 10 samples of each class (Right) Result compare with ICA_GL and HMM in the 20 samples of each class

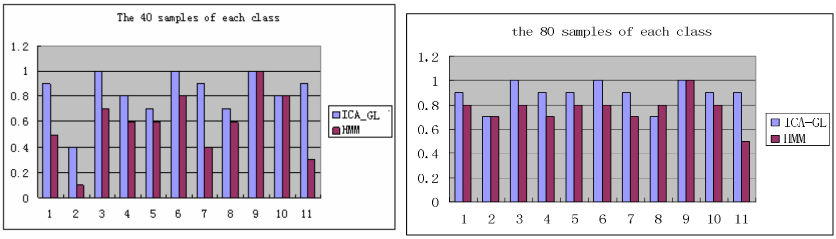


Fig. 5. (Left)Result compare with ICA_GL and HMM in the 40 samples of each class. (Right) Result compare with ICA_GL and HMM in the 80 samples of each class

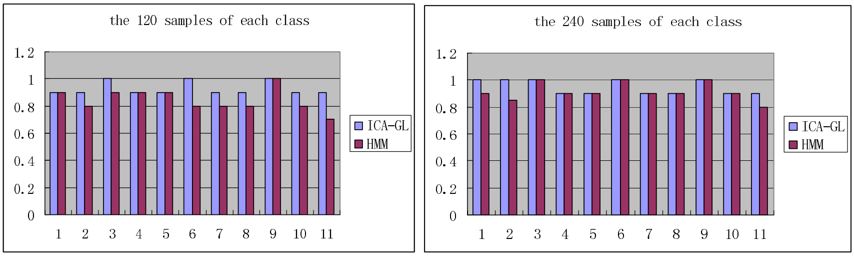


Fig. 6. (Left) Result compare with ICA_GL and HMM in the 120 samples of each class (Right)Result compare with ICA_GL and HMM in the 240 samples of each class

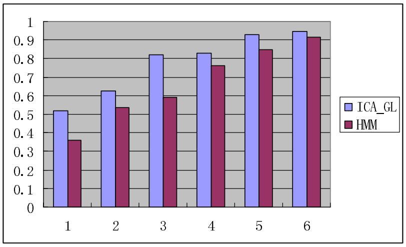


Fig. 7. Recognition Rate compare with ICA_GL and HMM

5 Conclusion

Dividing continuous speech is difficult, and it also directly influences the digits speech recognition rate. In this paper we changed the traditional speech recognition's pattern that is to be syncopated first before being recognized. The method employs an algorithm with ICA_GL. It achieved the continuous speech recognition without syncopating. The method is from the complicated two weights geometrical structure. We proposed a new algorithm in speech with ICA_GL learning algorithm. We hope it can be used in continuous speech of large vocabulary.

References

1. A.J. Bell and T.J. Sejnowski, Learning the higher-order structure of a natural sound. Network Comput. Neural Syst. **7** (1996), pp. 261–266.
2. Wang ShouJue, A new development on ANN in China - Biomimetic pattern recognition and multi weight vector neurons, LECTURE NOTES IN ARTIFICIAL INTELLIGENCE 2639: 35-43 2003
3. Wang Shoujue,etc. Multi Camera Human Face Personal Identification System Based on Biomimetic pattern recognition ,Acta Electronica Sinica 2003,31(1): 1-3
4. Wang Shoujue,etc. Discussion on the basic mathematical models of Neurons in General purpose Neurocomputer, Acta Electronica Sinica 2001, 29(5): 577-580
5. Xiangdong Wang, Shoujue Wang: The Application of Feedforward Neural Networks in VLSI Fabrication Process Optimization. International Journal of Computational Intelligence and Applications 1(1): 83-90 (2001)
6. Wenming Cao, Feng Hao, Shoujue Wang: The application of DBF neural networks for object recognition. Inf. Sci. 160(1-4): 153-160 (2004)
7. A. Hyvärinen, J. Karhunen and E. Oja. Independent Component Analysis, Wiley, New York (2001).
8. I. Csiszar and G. Tusnady, Information geometry and alternating minimization procedures, Statistics and Decisions,Supplementary Issue, No.1, 205-237, 1984.
9. S. Amari and H. Nagaoka, Methods of Information Geometry, AMS and Oxford University Press. 2000.

A Novel Information Hiding Technique for Remote Sensing Image

Xianmin Wang[1, 2], Zequn Guan[1], and Chenhan Wu[1]

[1] School of Remote Sensing Information Engineering,
Wuhan University, No.129 Luoyu Road,430079 Wuhan, China
[2] College of Hydropower and Information Engineering,
Huazhong University of Science and Technology,
No.1037 Luoyu Road,430079 Wuhan, China
wangxianmin781029@hotmail.com,
zequng@public.wh.hb.cn,
wuchenhan@etang.com

Abstract. In this paper, we introduce an information hiding technique into remote sensing area. We develop its connotation that the secret information is still hidden in the original remote sensing image. We propose a practical information hiding technique and a novel wavelet information hiding algorithm which is able to adapt to features of a remote sensing image. The technique is based on the embedding strategy of Discrete Wavelet Transform and HVS (Human Visual System) character. The algorithm is a blind one and has no influence on applied value of a remote sensing image.

1 Introduction

In a remote sensing image there sometimes exists secret information which cannot be seen by unauthorized users and should be hidden. In the past few years, the application of information hiding techniques in remote sensing area suffered from many restrictions, just because experts thought that remote sensing data should not be modified due to its scientific character. Until now, they become to realize that remote sensing data was influenced by connatural measure noise, so a mild modification can be accepted as soon as it contents proper quality requirements [1]. At the present international experts are all mostly engaged in the research on watermarking techniques and copyright protection of remote sensing images [1-5] and pay little attention to the secret information hiding technique for remote sensing images [6-7]. Barni et al. [1] applied the general DFT and DWT information hiding algorithms into remote sensing images and illuminated the general information hiding algorithms had much influence on applied value of a remote sensing image and didn’t adapt to it. So we should search for a proper algorithm suitable to remote sensing images. Yogesh et al. [2] proposed a spatial watermarking algorithm for remote sensing images that embedded watermarks into the region users were not interested in. However the algorithm should produce different remote sensing images with the watermark according to different users, so its operation was much complicated. Sean [3]

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 423 – 430, 2005.

proposed a Max-RMS sub-band algorithm based on DWT, but the algorithm was not a blind one, so was not practical. The article [6] advanced a spatial information hiding algorithm for remote sensing images which embedded the secret into the pixel-value compensation region. However the algorithm was not suitable to the secret object of abundant texture. The article [7] introduced Singular-Value-Decomposition (SVD) to well protect the textural and spectral information of the secret object. But the algorithm needed some original information and was a semi-blind one. In this paper, according to the requirements of information hiding technique for remote sensing images, we propose a wavelet algorithm based on DWT embedding strategy and HVS character which can rationally embed secret in terms of features of a remote sensing image and introduce a spread-spectrum technique and Hammin coding to enhance its robustness against image process.

2 Information Hiding Technique for Remote Sensing Image

In the paper we develop the connotation of information hiding, namely hiding the secret still in the original remote sensing image, which can reduce the transferring, storage and processing data amount and further enhance the confidentiality and safety of the information hiding system comparing with hiding secret into another irrelatively digital media. The information hiding technique is shown in Fig. 1.

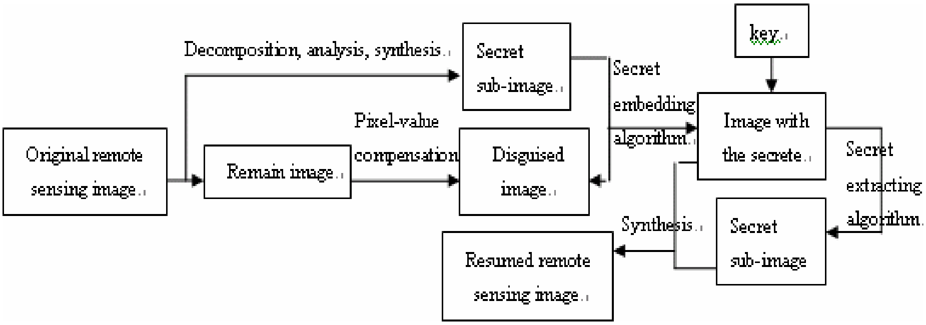

Decomposition, analysis, synthesis
Secret sub-image
Secret embedding algorithm
key
Image with the secrete
Original remote sensing image
Remain image
Pixel-value compensation
Disguised image
Secret extracting algorithm
Synthesis
Secret sub-image
Resumed remote sensing image

Fig. 1. Information hiding technique for remote sensing image

2.1 Decomposition, Analysis and Synthesis of Secret

We determinate or classify a remote sensing image into various coverages or classifications, according to the spatial and spectral characters of the secret recognized it in the corresponding coverage or classification and segmented and extracted it by its minimum external rectangle.

2.2 Pixel-Value Compensation and Production of Disguised Image

The final purpose of pixel-value compensation is to resume the original physiognomy without the secret ground object or replace the secret ground object with another open one to form a kind of visual deceiving for non-authorized users.

2.3 Resumption of Remote Sensing Image

In the remote sensing image with the secret, we exploit the keys to replace the pixel-value compensation sub-image with the extracted secret and obtain the resumed one.

3 Information Hiding Algorithm for Remote Sensing Image

The flow chart of the information hiding wavelet algorithm for remote sensing image is shown in Fig.2, in which ‘n’ is chosen as 3,4,5.

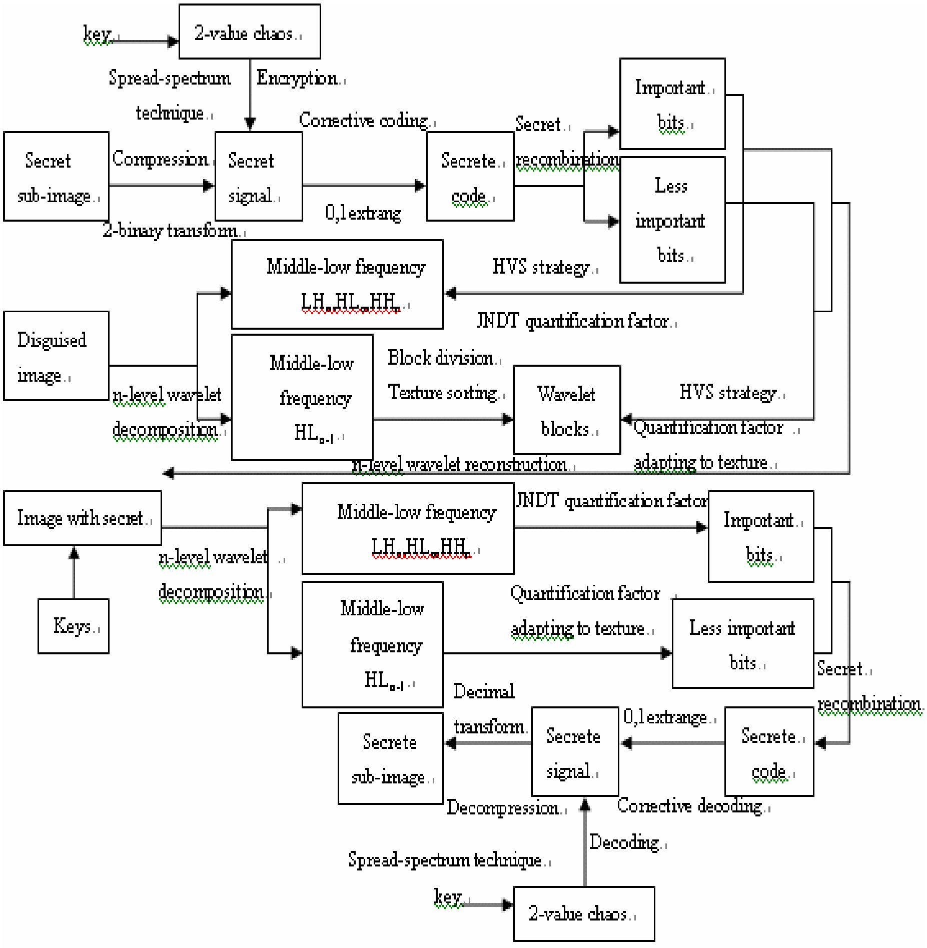


Fig. 2. Flow chart of information hiding wavelet algorithm for remote sensing image

3.1 Embedding Algorithm of Secret Information

Embedding Strategy Based on DWT

We choose the middle-low-frequency wavelet coefficients of a remote sensing image as the embedded region. We make n-level wavelet decomposition, firstly choose the sub-band HL_n to embed the secret; and if the secret remained, according to the importance order [8] of the frequency sub-bands, choose the sub-bands of LH_n、HH_n and $HL_{n-1}$to embed the remained secret signal.

Adaptive Embedding Algorithm Based on HVS

We exploit HVS to make the embedding energy and adapt to the feature of each wavelet sub-band, respectively. For the sub-bands of HL_n, LH_n and HH_n (n=5, 4 or 3), we exploit the just noticeable distortion threshold JNDT as the quantification factors. And as for the sub-band HL_{n-1}, we divide it into 8×8 wavelet blocks and compute their visual textural characters according to formulas (1) and (2), in terms of which we determine their quantification factor and sort them. Firstly, we embed the secret signal into the wavelet block of the biggest textural value; and if the secret is remained, in turn embed the remained signal into the corresponding wavelet blocks according to the descending order of their textural values.

$$B(k,l) = \frac{1}{64}\sum_{i,j=0}^{7} D^{k,l}(i,j) \ , \tag{1}$$

$$T(k,l) = \sum_{i,j=0}^{7} \left| D^{k,l}(i,j) - B(k,l) \right| \ , \tag{2}$$

in which $D^{k,l}(i,j)$ is the wavelet coefficient of the position (i,j) in the wavelet block (k,l) , $B(k,l)$ is the luminance of the wavelet block (k,l) , and $T(k,l)$ is the textural value of the wavelet block (k,l) .Then the quantification factor of the wavelet block (k,l) is

$$q(k,l) = T(k,l)/K \ , \tag{3}$$

in which K is a constant, and its value is determined by the concrete remote sensing image.

Embedding of Secret Information

We exploit DCT to compress the secret sub-image [6] and spread-spectrum technique and the key to encrypt the secret binary, utilize (7,4) Hammin code to encode the secret signal, and exchange ‘0’ and ‘1’ in the secret signal if ‘1’ is more than ‘0’ to reduce the influence on the carrier remote sensing image. Furthermore we embed the important bits of the secret signal to the important wavelet coefficients and less important to less important coefficients which would to the biggest degree enhance the robustness of the secret information.

3.2 Extracting of Secret Information

While extracting the secret information and resuming the remote sensing image, the algorithm doesn't need the original image and is a blind one. The key and the size and position of the secret sub-image in the original one can be used as the keys.

4 Simulant Experiments

We exploit a SPOT panchromatic image (2325×2225) of Yunnan as the experimental image and the airdrome in it as the secret information (390×254×8=792480bits) which are shown in Fig.3 (a)-(b). The image after extracting secrete is shown in Fig.3 (c). The pixel-compensated image and the image with the secret are shown in Fig.3 (d)-(e) respectively, in which PSNR=55.01dB. The 10-times difference image between the pixel-compensated image and the one with the secret is shown in Fig.3 (f). From this figure, it can be seen that the algorithm has very strong transparency and high fidelity. The extracted airdrome and the resumed image are shown in Fig. 3 (g)-(h). The correlative value and correct ratio of the extracted secret are NC=1, BCR=99.93%, respectively.

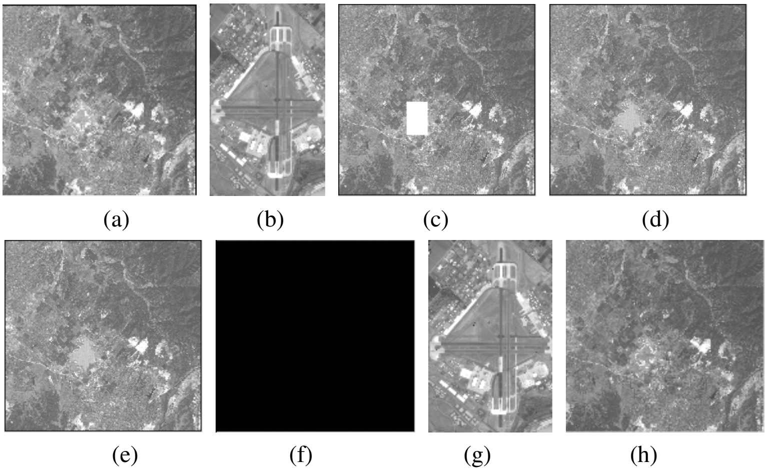


Fig. 3. Remote sensing image and secret sub-image

Table 1. Statistic results of edge detection

	NNMP	RNMP
Pixel-compensation image and the one with secret	12236	0.24%
Original image and the resumed one	12283	0.24%

We make experiments of edge detection (by canny operator) and image classification (9 classifications) of the remote sensing images, and the statistic results are shown in Tables 1 and 2 respectively, in which NNMP and RNMP denote number of not matching pixels and ratio of not matching pixels, respectively. Therefore the wavelet algorithm has no influence on applications of a remote sensing image.

Table 2. Statistic results of classification

	NNMP	RNMP
Pixel-compensation image and the one with secret	1544	0.03%
Original image and the resumed one	1569	0.03%

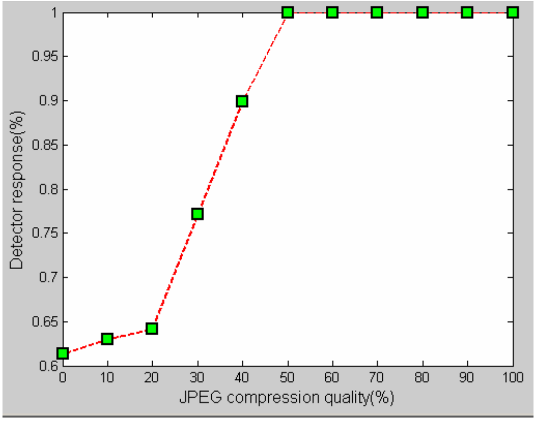
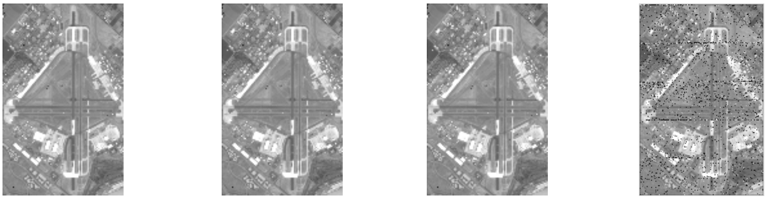


Fig. 4. Plot of the detecting response corresponding to JPEG lossy compression ratios



(a) JPEG quality 100% (b) JPEG quality 80% (c) JPEG quality 60% (d) JPEG quality 40%

Fig. 5. Extracted secret sub-images under JPEG lossy compression

Considering the transferring means of remote sensing images, robustness against image compression and noise-adding should be primarily considered. Figure 4 shows the detecting response plot under JPEG attacks. Figure 5 shows the extracted secret sub-images at JPEG qualities 100%, 80%, 60% and 40%, respectively. While JPEG quality is higher than 40%, the secret sub-image can be accurately extracted, so the wavelet algorithm has very strong robustness against JPEG compression.

Figure 6 shows the extracted secret sub-images under attacks of Gaussian noise and white-black noise of different strength, and Tables 3 and 4 show the detecting responses under the corresponding strength of Gaussian and white-black noises. So the wavelet algorithm proposed in the paper is quite robust against noise.

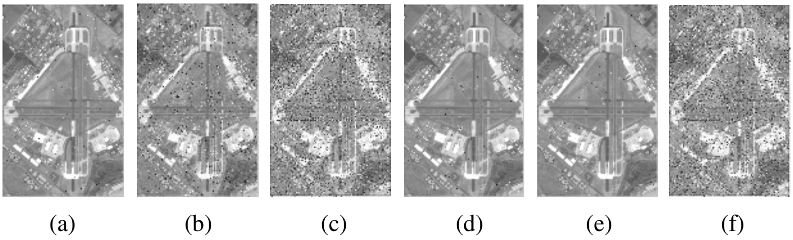


Fig. 6. Extracted secret sub-images under noise

Table 3. NC and BCR values of the extracted secret under Gauss noise

Noise type	Noise ratio	PSNR/dB	NC	BCR
Gaussian noise	1%	46.16	0.9895	99.78%
	5%	45.77	0.9697	98.57%
	12%	44.94	0.8749	85.73%

Table 4. NC and BCR values of the extracted secret under white-black noise

Noise type	PSNR/dB	NC	BCR
White-black noise	45.29	0.9928	99.84%
	41.68	0.9896	99.81%
	31.83	0.8702	86.04%

5 Conclusions

In this paper we develop the connotation of information hiding and propose a novel information hiding technique and wavelet algorithm for remote sensing images. The experimental results show that the algorithm proposed in the paper has the advantages of good transparency, large information capacity, correct extraction of secret, well protecting the textural and spectral features of the secret ground object and vividly resuming the remote sensing image, but also has a strong robustness against JPEG lossy compression and noise adding. Furthermore this algorithm has no influence on applied value of a remote sensing image. The technique and algorithm proposed in the paper can also be extended to all kinds of remote sensing images, such as fused ones, multi-spectral ones and hyper-spectral ones, and are a practical information hiding technique and algorithm for remote sensing images.

References

1. Barni M., Bartolin F., Cappellini V.: Watermarking-based protection of remote sensing images: requirements and possible solutions. IEEE Transactions on Image Processing, Vol.12. No.6. (2003) 1–12
2. Yogesh C., Gupta P., Majumder K.L.: Digital Watermarking of Satellite Images. In: Proc. Of IEEE International Conference on Image Processing, Vol.1. (2003) 288–297
3. Sean B.Z., Hrishikesh T., Fowler J. E.: Wavelet-Based Watermarking of Remotely Sensed Imagery Tailored to Classification Performance.In: Proc.of the IEEE Workshop on Advances in Techniques for Analysis of Remotely Sensed Data, Washington.DC (2003) 564–579
4. Heileman G. L., Yang Y.: The Effects of Invisible Watermarking on Satellite Image Classification. In: Proc. Of the 2003 ACM Workshop on Digital Rights Management, Washington DC, USA (2003) 120–132
5. Kaewkamnerd N., Rao K.R.: Wavelet Based Watermarking Detection Using Multiresolution Image Registration. In: Proc. Of the 2003 ACM Workshop on Digital Rights Management, Washington DC,USA (2003) 155–167
6. Guan Z.Q., Wang X.M., Wu C.H.: A Practical Blind Algorithm for 2-Dimension Secret Information Hiding In Remote Sensing Image. Geomatics and Information Science of Wuhan University, Vol.29. No.4. (2004) 296–301
7. Wang X.M., Guan Z.Q., Wu C.H.: An Adaptive Blind Algorithm for 2-Dimension Secret Information Hiding Based on Feature of Remote Sensing Image. Computer Engineering and Application, Vol.40. No.19. (2004) 17–20
8. Huang D.R., Liu J.F., Huang J.W.: An Embedding Strategy and Algorithm for Image Watermarking in DWT Domain. Journal of Software, Vol.13. No.7. (2002) 1290–1296

Content-Based News Video Mining*

Junqing Yu[1], Yunfeng He[1], and Shijun Li[2]

[1] Computer College of Science & Technology,
Huazhong University of Science & Technology , Wuhan, 430074, China
yjqinghust@126.com, heyunfeng@tom.com
[2] School of Computer, Wuhan University, Wuhan, 430072, China
shjli@public.wh.hb.cn

Abstract. It is a challenging issue to analyze video content for video mining due to the difficulty in video representation. A hierarchical model of video representation is proposed with a schema for content-based analysis of news video in this paper. The research problem targeted in this paper is to mine a massive video database to retrieve specific clip based on content defined by users. This is frequently encountered in entertainment and video editing. A novel solution to this problem is developed in this paper, in which the consecutive news video is segmented into shots, scenes and news items using multimodal features based on the hierarchical model. To summarize the content of video, a video abstract is developed. The experimental evaluation demonstrates the effectiveness of the approaches discussed in this paper.

1 Introduction

With the development of multimedia and web technology, the multimedia data, including image, audio and video, have been produced massively. Digital video rapidly becomes an important source for information, education and entertainment. It is needed urgently the advanced technologies for organizing, analyzing, representing, indexing, filtering, retrieving and mining the vast amount of videos to retrieve specific information based on video content effectively, and to facilitate new and better ways of entertainment and multimedia applications. Content-base video analyzing and retrieval are important technologies, which have been an international research focus in recent ten years. As challenging problem, content-based video mining is also emphasized by lots of researchers. Although numerous papers have been published on data mining [1, 2], few of them deal with video mining directly [3, 4]. Low features, such as color, texture, audio and motion, can be used to segment video sequence into shots and to extract caption or other region of interest for video data management and retrieval. However, the mining of video data based on its content is still in its infancy. Due to the inherent complexity of video data, existing data mining algorithms and techniques can not be used directly in video data. The new mining techniques or modified ones should be designed to facilitate the video data mining process.

* This paper is financially supported by Natural Science Foundation of Hubei Province (2004AA101C94).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 431–438, 2005.

Generally speaking, there are two kinds of videos in our daily life [5]: videos with some content structures and videos without any content structure. The former are videos such as movies and news where scenarios are used to convey video content. The latter is like surveillance videos, they have no scene change, therefore no content structure can be found among them. Just because of these, we can not process video data using a unified approach like dealing with text data. Specific processing schema should be designed for the different kinds of video data and many efforts had been made.

In today's society, the amount of news information generated is growing exponentially. Moreover, the data is made available in more than one dimension across different media such as video, audio, and text. This mass of news information poses serious technological challenges in terms of how news data can be integrated, processed, organized, and indexed in a semantically meaningful manner to facilitate effective retrieval. Because of its usefulness and importance, there have been many research efforts in news video analysis. R. Mohan [6] proposes to segment TV news by synchronizing images with the associated close-captions or teletext (the European version of close-captions). L. Chen [7] presents multi-criteria video segmentation based on image and sound analysis. Zhang et al [8] base their work on the anchorperson position in order to split the news into independent subjects. This model fits a type of news where the anchorperson and camera position do not change much. The Informedia's work is very impressive, in which speech recognition and image analysis were combined to extract content information and to build indexes and abstracts of news video [9, 10]. Lately, many researchers adopted the idea that image, audio and speech analysis are integrated in video content analysis [11, 12].

The rest of the paper is organized as following. Section 2 proposes a hierarchical video organization schema. A news video sequence is segmented into shots and news items based on audiovisual features in Section 3. A novel mining tool, news video abstract based on key frames, is introduced in Section 4. Section 5 concludes this paper.

2 Hierarchical Video Organizing Schema

For the video with content structure, video data usually bear hierarchy in both content and structure. Accordingly, a hierarchical video organizing schema is introduced and an independent object identifier can be assigned to every video object.

2.1 Hierarchical Video Organizing Model

The original video, with content structure, can be organized in five levels: video event, episode, scene, shot and frame image. All but the shortest video are made up of a number of distinct scenes, each of which can be further broken into individual shot depicting a single view, conversation or action. A shot designates a continuous sequence of frames, which are bottom level of the model and are corresponding to the temporal image sequence of the original video. Using high level semantics, some scenes (neighboring or not) can be combined into episode. Episode makes up the semantic unit of video and depicts a story or an action. In the same episode, the content of scenes is relevant, but they can be separated in temporal order.

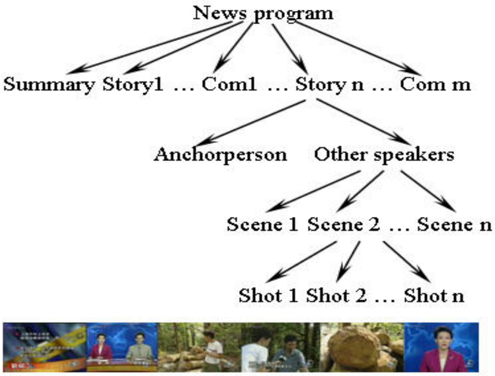

News program
Summary Story1 … Com1 … Story n … Com m
Anchorperson Other speakers
Scene 1 Scene 2 … Scene n
Shot 1 Shot 2 … Shot n

Fig. 1. Hierarchical organization of news video

2.2 News Video Organizing Schema

We can use hierarchical model to organize news video. Fig.1 shows the hierarchical organization of news video. A typical national news program (e.g., CCTV news) consists of news and commercials. News is made up of several headline stories, each of which is usually introduced and summarized by the anchor prior to and following the detailed report by correspondents, quotes, and interviews from newsmaker. Commercials are usually found between different news stories. Based on this hierarchy of news video, we propose a prototype of news video database system, with which continuous news video can be automatically analyzed based on content by utilizing different cues of audiovisual information.

3 Content-Based Analysis of News Video

To retrieve or mine news video data, one of the most important tasks is to the transform the original video sequence into a hierarchical dataset according to the model depicted in Fig. 1. To facilitate this goal, we adopt some video processing techniques to analyze the news video. Users have different needs when mining news data. For instance, some users may want to directly retrieve a news story; some may like to listen to the news summary of the day in order to decide which story sounds interesting before choosing what to watch further. In order to satisfy different requirements, a segmentation mechanism is needed that partitioning news video data in different ways so that direct indices to the events at different levels of abstraction can be automatically established.

3.1 Feature Extraction

Digital video is a dynamic sequence, which contains image and audio signal simultaneously. Compared with the text, static image and audio, the video's content is much more complicated and abundant. However, restricted by the present computer technology and artificial intelligence, high-level semantics cannot be directly extracted from video and low-level features should be extracted firstly. Three sources of information can be used for video processing. They are color, audio and motion. Most of the existing video segmentation algorithms are based on visual information, such as color histogram, edge feature, etc. Combined audio and visual approaches have been considered only recent years. Here, we propose a new feature extracting approach, which is based on Microsoft DirectShow SDK system. Special audio-visual feature extractors, which are custom filters in DirectShow, are designed to extract features in real time. Extracted features are listed in the following [12, 13]:

Audio features: NSR (non-silence-ratio), ZCR (zero crossing rate), STE (short-time energy), VDR (volume dynamic range), VU (volume undulation), FC (frequency centroid) and BW (bandwidth);

Color features: DC (dominant color), DCStd (standard deviation of dominant color), PDC (percentage of dominant color), Color Histogram, DCH (mean of difference between the color histograms);

Motion features: DMX (X component of dominant motion), DMY (Y component of dominant motion), PDM (percentage of dominant motion), ME (mean motion energy).

3.2 Anchorperson Frame Detection

The main content of news video is a series of news story, so exactly detecting the boundary between news stories is very important for the content-based mining news data. It is not difficult to find that the anchorperson frame is usually the beginning or ending frame of individual news story. Just because of this, automatically detecting the anchorperson frame can help recover the news stories. Most of the existing approaches to this problem are either based on face detection or based on speaker identification, so their computational complexity is very great. Here, a completely new and simplified method was proposed. The detailed detection process can be explained as follow:

Step 1: Establishing the DC template of anchorperson frame. For most of news video, the first anchorperson frame usually appears after the theme music in a relatively rigid time interval, the template frame can be automatically chose through the detection of theme music, which has been discussed in [14]. Suppose M frames are chose to be template frame, the DC template can be computed by equations 1 and 2.

$$DCStd_T = \frac{1}{M}\sum_{i=1}^{M} DCStd_T(i) \tag{1}$$

$$PDC_T = \frac{1}{M}\sum_{i=1}^{M} PDC_T(i) \tag{2}$$

Where $DCTStd_T(i)$ and $PDC_T(i)$ indicate the standard deviation and percentage of dominant color in i^{th} template frame. $DCTStd_T$ and PDC_T stand for the template features.

Step 2: Computing the DCStd and PDC features of every frame image in the news sequence.

Step 3: Template matching. We compute the difference, *D(i)* ,between the template feature and responding feature in i^{th} frame by equation 3. If *D(i)* is less than the predefined threshold, which can be adjusted adaptively, the frame can be identified to be anchorperson frame and it can be marked automatically.

$$D(i)=\sqrt{C_1(DCStd_i-DCStd_T)^2+C_2(PDC_i-PDC_T)^2}\quad(1\le i\le N) \tag{3}$$

Where C_1 and C_2 indicate the weight of DCStd and PDC feature, N refers to the reframe number in the news sequence. If in RGB color model, we can compute the DCStd and PDC in red, green and blue separately.

3.3 News Story Segmentation

In the news video, a period of news program usually consists of several news stories, and the news story is made up of some scenes or shots. Therefore, the approach to parsing shot and scene can be used here. To detect news story boundaries, robust anchorperson frame detection is needed, because the frame with anchorperson is often appeared at the beginning of news story. The silence between stories is also important cue. Just based on these considerations, approach based audio-visual information is proposed [14, 15]. The entire segmenting process can be divided into two steps, one is to search candidate boundary points, and the other is to verify the candidate boundary points.

Step 1: Searching candidate boundary points. Continuous news video sequence consists of two types of clip, one is anchorperson frame chip, and the other is non-anchorperson frame chip. Using the approach discussed in Section 3.2, we can find the candidate boundary points conveniently. Fig.2 gives illustration of 30 minutes news video, in which SC1, SC2, … , and SC12 stand for candidate points.. Among the candidate points, some are not exactly the boundary points of news story, and we call them false points. They maybe belong to the same news story. On the other hand, some true boundary point might be ignored if no anchorperson frame exists in some news story, or one story ends by anchorperson frame and the next neighboring story begins with anchorperson frame. Therefore, we have to verify the candidate point in the next step.

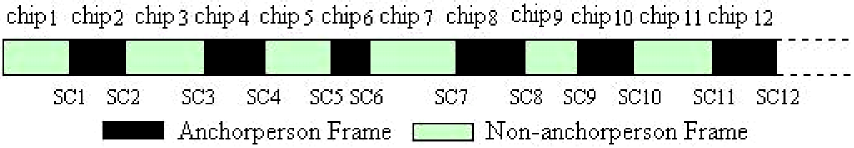

chip1 chip2 chip3 chip4 chip5 chip6 chip7 chip8 chip9 chip10 chip11 chip12
SC1 SC2 SC3 SC4 SC5 SC6 SC7 SC8 SC9 SC10 SC11 SC12
Anchorperson Frame
Non-anchorperson Frame

Fig. 2. Candidate boundary points of news stories

Step 2: Verifying the candidate boundary points. The objective of this step is to delete false points and supplement ignored ones. Here, the silence clip between the news stories was utilized. The short-time zero (STZ) rate is used to detect the silence chip, Fig. 3 gives a silence detecting result.

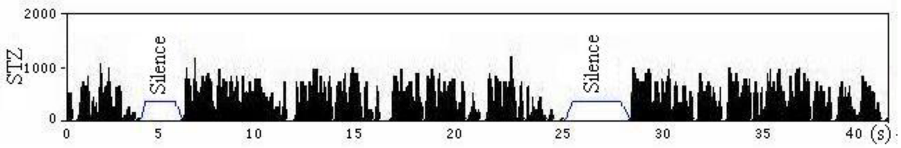


Fig. 3. Detection result of silence chip between news stories

For convenience, we can use a dualistic group to express the detected silence chip like equation 4.

$$SG(i) =< s_i, e_i >, i, s_i, e_i = 1,2,\cdots, and\ s_i \le e_i \tag{4}$$

Where, s_i and e_i indicate the starting and ending frame number of i^{th} silence chip. Two theorems are used to finish the verifying process as followed.

Theorem 1: If no silence chip exists in one candidate boundary point, this is false point and should be deleted.

Theorem 2: If one silence chip, $SG(j)=<s_j, e_j>$, exists in one anchorperson frame chip, there is a boundary point at this silence chip and it should be supplemented. Suppose this supplemented point to be $SB(j)$, it can be computed by equation 5.

$$SB(j) = (s_j + e_j)/2 \tag{5}$$

Table 1. The experimental results of news story segmentation

News video	Actual	Detected	False	Missed
News reports	44	45	1	0
Night news	38	38	1	1
Total	82	83	2	1

Through the above two steps, all the true boundary points have been identified, and using them we can segment a continuous news video into a series of news stories. Table 1 gives the experimental results of news story segmentation for CCTV news reports and night news. The results have revealed the effectiveness of segmentation algorithm.

4 News Video Abstract Based on Key Frame

The summarization of video content provides an effective way to speed up video browser and assist for video mining and retrieval, a novel method, video abstract, is proposed for automatically summarizing news video content. A video abstract is a compact representation of video content. It is defined to be a sequence of moving images, extracted from a longer video, much shorter than the original, and preserving the essential message of the original [16]. Content-based video abstract is an important type of content-based video mining tool. To concisely and informatively summarize news video content to maximum extent, key-frame-based abstract is put forward in this paper.

4.1 Key Frame Extraction Based on Caption and Visual Information

Key frame is the frame which can represent the salient content of the shot and summarizes contents of a video sequence. Key frame selection is an important method of summarizing a long video program. Key frames are often arranged as storyboards, in which the key frames represent shots and sequences to summarize the story. Depending on the content complexity of the shot, one or more key frame can be extracted from a single shot. In the news video, frame with caption or text is usually contain the main idea of a news story. Therefore, an algorithm based on caption and image information is introduced to extract key frame from news video. The detailed algorithm can be referred in [14].

4.2 News Video Abstract

Based on the extracted key frames, a news video abstract prototype [14] is developed using DirectShow architecture and COM criterion. Users can use this system not only to analyze news video, but also to mine and browse interested news story. Depending on user's different demand, this system can afford different searching depth. For example, you can just browse a mosaic picture of key frames firstly, and then you can choose your interested news to watch the details. Experiments reveal that news video abstract is a very effective mining tool news video database.

5 Conclusion

In this paper, we have addressed video mining techniques for efficient video organization, management and retrieval. To achieve this objective, a hierarchical video organizing model is proposed. A news video content structure mining scheme is introduced for parsing the news video into a series news stories. Both visual and audio features are real-timely extracted and utilized to analyze news video. A novel video mining tool, key-frame-based video abstract is introduced to summarize and browse the content of news video. Experimental results demonstrate the efficiency of our framework and strategies for video data mining. However, research of video mining is still primitive and much room remains for improvement. Our future work will include the video indexing techniques based on multimodal information.

References

1. U. Fayyad: Data Mining and Knowledge Discovery in Database: Implications for Scientific databases. Proceeding of Ninth International Conference on Scientific and Statistical Database Management (1997) 2-11
2. M.S. Chen, J. Han and P.S. Yu: Data mining: An Overview from a Database Perspective. IEEE Transactions on Knowledge and Data Engineering (1996), 8(6):866-883
3. J.-Y. Pan, C. Faloutsos: VideoCube: A New Tool for Video Mining and Classification, ICADL 12(2002)
4. Xingquan Zhu, Walid G. Aref, Jianping Fan, Ann Christine Catlin, Ahmed K. Elmagarmid: Medical Video Mining for Efficient Database Indexing, Management and Access. ICDE (2003) 569-580
5. Xingquan Zhu and Xiaodong Wu: Sequential Association Mining for Video Summarization. Proceedings of International Conference on Multimedia and Expo, (2003) 333-336
6. R Mohan: Text-based Search of TV News Stories. SPIE Proc. of Multimedia and Archiving Systems, NJ: SPIE Press (1996) 2-13
7. L Chen, P Faudemay: Multi-Criteria Video Segmentation for TV News. Proc. of 1st Multimedia Signal Processing Workshop, NJ: IEEE Press (1997) 319-324
8. H J Zhang, S Y Tan, S W Smoliar: Automatic Parsing and Indexing of News Video. Multimedia Systems, 1995, (2): 256-266.
9. Hauptmann A, Witbrock M.: Story Segmentation and Detection of Commercials in Broadcast News Video. http://www.ieee.org/ieeexplore, October 2000.
10. A G Hauptman, M Smith: Text, Speech and Vision for Video Segmentation: The Informedia Project. Working Notes of IJCAL Workshop on Intelligent Multimedia Information Retrieval, NJ: IEEE Press (1995) 17-22
11. J Huang, Z Liu, Y Wang, Y Chen: Integration of Multimedia Features for Video Classification Based on HMM. http://vision.poly.edu, October 2000.
12. Z Liu, Q Huang, A Rosenberg: Automated Generation of News Content Hierarchy by Intetrating Audio, Video, and Text Information. ICASSP-1999, NJ: IEEE Press (1999) 3025-3028
13. J Huang, Z Liu, Y Wang: Integration of Audio and Visual Information for Content-based Video Segmentation and Classification. Journal of VLSI Signal Processing System for Signal, Image, and Video Technology (1998), 20(2): 61-79.
14. Yu Junqing: Research of Content-Based Video Abstract. Wuhan: Wuhan University (2002)
15. Wang Weiqiang, Gao Wen: Automatic Parsing of News Video Using Multimodal Analysis. Journal of Software (2001), 12(9): 1271-1278
16. Rainer Lienhart, Silvia Pfeiffer and Wolfgang Effelsberg: Video Abstracting. Communications of the ACM (1997), 40(12): 55~62

Automatic Image Registration via Clustering and Convex Hull Vertices Matching

Xiangyu Yu and Hong Sun

Signal Processing Lab, School of Electronics Information,
Wuhan University, 430072 Wuhan, China
yuxywh@tom.com, hongsun@whu.edu.cn
http://dsp.whu.edu.cn

Abstract: A coarse-to-fine automatic point-based image registration method is proposed in this paper. At the first stage, clustering is used to determine the scale parameter and the rotational parameter candidates between images. Convex hull vertices correlation is applied subsequently to determine the correct rotational parameter. With the coordinates of matched point pairs and the above parameters, the translational parameter and the coarse registration result can be determined. At the second stage, control point pairs, which determine parameters of mapping polynomial, are formed by iterative convex hull vertices matching. Thus the registration result is refined. Experiments indicate that this approach can automatically align images in different resolutions.

1 Introduction

Image registration, an important task in image processing, is widely used in computer vision, medical image analysis and remote sensing data processing [1,2]. The purpose of image registration is to establish the correspondence and determine the geometric transformation between images.

Image registration techniques can be classified into manual and automatic methods. Manual registration is tedious, time-consuming and prone to error, so it is necessary to develop automatic image registration algorithm. Existing automatic image registration techniques fall into two categories: the intensity-based and the feature-based methods [2]. Feature-base methods, which extract common structure from both images, are more suitable for multi-sensor image registration.

Traditionally, feature-based automatic image registration consists of three independent steps [2]:

1) Obtain corresponding features;

2) Match corresponding features;

3) Geometric transform of sensed image by means of mapping function.

Point feature is one of the features used most frequently in image registration. There are many point features available [2]: line intersections, road crossing, centroids of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 439–445, 2005.

regions, local curvature maximum points, etc. Most point matching algorithms are based on affine model. Relaxation approach was used by Ton *et al.* to match centroids of enclosed regions [3]. Stockman *et al.* first used clustering technique to align images with affine distortion [4]. Chang *et al.* [5] and Seedahmed *et al.* [6] used similar idea to achieve point pattern matching. Shekhar *et al.* generalized it to quasi-affine model [7]. The affine model is given by Eqn. (1).

$$\begin{pmatrix} m \\ n \end{pmatrix} = s\begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix}\begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix} \tag{1}$$

where $(m,n)^T$ and $(x,y)^T$ are corresponding points in two images, s is the scale parameter, β is the rotational parameter, and t_x and t_y are translational parameters in two directions. Using the notation $p=(x,y)^T$, this transform can be expressed as a sequence of four independent stages:

$$T(p)=T_{tx}(T_{ty}(T_s(T_\beta(p)))) \tag{2}$$

where

$$T_\beta(p) = \begin{pmatrix} \cos\beta & -\sin\beta \\ \sin\beta & \cos\beta \end{pmatrix} p \tag{3}$$

$$T_s(p)=sp \tag{4}$$

$$T_{tx}(p) = p + \begin{pmatrix} t_x \\ 0 \end{pmatrix} \tag{5}$$

$$T_{ty}(p) = p + \begin{pmatrix} 0 \\ t_y \end{pmatrix} \tag{6}$$

Similar to Hough transform, clustering approach maps information into different parameter space. Overlapped bins in parameter space are formed. The occurrence in each bin is counted. For feature points, each two different points on one image are connected, and the angle difference and scale between a pair of line sections from respective images are recorded. In both parameter spaces, the center of the bin which contains the most counts is the desired parameter.

In order to improve the efficiency of clustering, Goshtasby *et al.* took convex hull vertices instead of all points in clustering [8]. The convex hull of a point set is the smallest convex set that includes all the points and can be used as an effective description of point feature. When two point sets are matched, their convex hulls are in pair too. Convex hull is not affected by inner noise points, and each noise point outside only affects its neighborhood.

There are some problem occur when implementing above methods. Firstly, in Stockman's approach, all line sections have been assigned a direction, which depends on the property of points. But in general cases, the points extracted have no priori knowledge, so both directions are considered, that's why in [7], there are two candidates of rotational parameter with 180° in difference. Secondly, there is no obvious peak when determining the translational parameter via clustering, so it is necessary to look for a new approach for translational parameter. Thirdly, if there are some noise points in [8], correct parameters can not be determined.

Consequently, an approach is proposed in this paper to achieve feature points-based registration. At the first stage of this method, convex hull vertices correlation is applied on clustering result to indicate the correct value from the two translational parameter candidates. At the refinement stage, the iterative convex hull vertices matching are used to determine the control point pairs, the later in turn form the parameters of mapping polynomial. After all points processed, the final mapping polynomial is determined and exact registration is achieved.

2 Coarse-to-Fine Point-Based Image Registration Approach

The method proposed will be discussed in details below. The first stage of the approach gives the coarse result and refinement is finished in the next stage.

2.1 Coarse Registration by Clustering and Convex Hull Vertices Correlation

After feature points are extracted from both images, each two points on the same image are connected. All the line sections from the first image are compared with those from the second one by the orientation difference and the length ratio. In recording the orientation difference, the range is set to [0, 360°) and the resolution is 0.1°. After processing each line section of both images, the center of the two bins contain most counts of angle values are the candidate rotational parameters, which have 180° in difference. Meanwhile, record the length ratio of each line of reference image to sensed image with a resolution 0.1, and the center of the bin which contains most counts is the desired scale parameter.

Convex hull vertices correlation is used to determine the true rotational parameter from the two candidates. The sensed image is scaled according to the scale parameter and rotated according to the two candidate angles. Determine the convex hull of both point sets at both angle. Two vertices from each hull are overlapped and whether there is a matched point for the other vertices is determined. If there is a point within the neighborhood of the corresponding position of a given vertex, then this vertex has a counterpart. Record the number of matched pair at each case. The angle which contains the most number of count is the desired rotational angle, while the translation can be determined by substituting the coordinates of the vertices pairs, the rotational and scale parameters into Eq.(1). Thus all affine parameters have been determined.

2.2 Registration Refinement by Iterative Convex Hull Vertices Matching

The affine parameters determined above lead to the coarse registration result. In order to achieve high precision registration, it is better to refine the coarse result. The convex hull vertices matching algorithm is used for this purpose. Affine parameters determined above are used as initial parameter. The transform parameter is updated by the matched point pair determined during the procedure. There are two sets in the procedure, one is called reference and the other is called current set. Both point sets are set to be the reference set alternately, and only those vertices on both hulls are considered. At each step, current processed point set is transformed according to the updated parameter, its convex hull is determined, and its convex hull vertices are compared with those of reference set. Those vertices that satisfy discard criterion are discarded while those satisfying matching criterion are recorded in the matched points list. All matched points are used to update the transform parameter, and point sets are updated by deleting those discarded points in the current set and matched vertices in both sets. Then reference and current sets are exchanged and above procedure are repeated until all points are processed. The diagram of the procedure is in Fig.1.

For each vertex, the discard and acceptance criteria are listed below with a given acceptance and discard threshold:

1) If there is a vertex in acceptance threshold and no points in discard threshold, current vertex is matched with the vertex in acceptance threshold. Record this pair.

2) If there is point in discard threshold and no points in the acceptance threshold, keep current vertex and give it a flag; if its correspondence can't be decided for three times, delete it.

3) Assume there is no point in discard threshold. If current vertex is inside reference convex hull, keep it without recording; otherwise, if it is outside reference convex hull, delete current vertex and update convex hull.

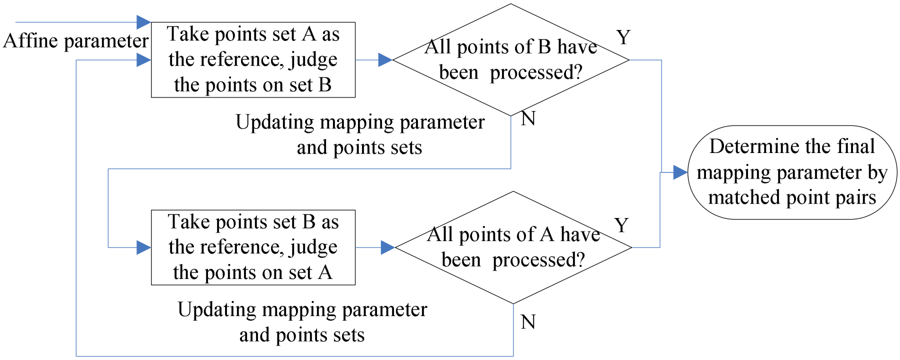

Affine parameter
Take points set A as the reference, judge the points on set B
All points of B have been processed?
Y
N
Updating mapping parameter and points sets
Determine the final mapping parameter by matched point pairs
Take points set B as the reference, judge the points on set A
All points of A have been processed?
Y
N
Updating mapping parameter and points sets

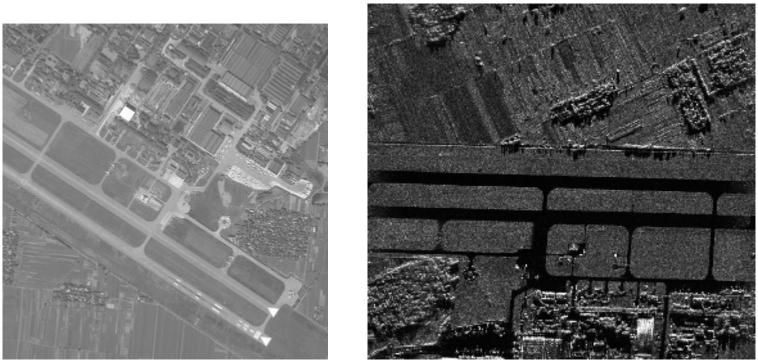
Fig. 1. Diagram of convex hull vertices matching

After all points having been processed, all matched points are used to determine the mapping polynomial parameter by solving the normal equation. Since the image difference is not much far from affine transform, bi-polynomial of order one, which is given in Eq.(7), is used.

$$\begin{aligned} m &= a_0 + a_1 x + a_2 y + a_3 xy \\ n &= b_0 + b_1 x + b_2 y + b_3 xy \end{aligned} \tag{7}$$

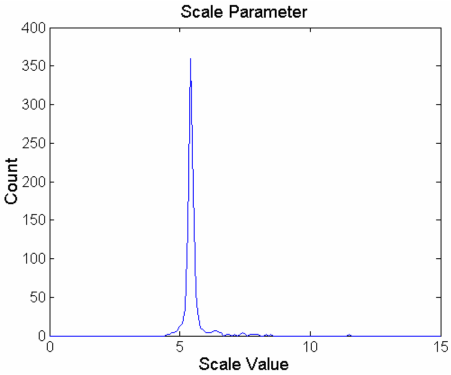
3 Experiment Results

A large scale experiments have verified this algorithm, for the limitation of the scope, one result is presented in this paper. In this experiment, two images of an airport in Shaanxi Province are used to test this algorithm, which are shown in Fig.2. One of which is Synthetic Aperture Radar (SAR) image, its resolution is 3 m/pixel; the other is Quickbird optical image, whose resolution is 0.61m/pixel. A/G algorithm is used to detect the runway in SAR image[9]. Threshold algorithm is applied on optical image. Contour tracking is used to determine the contour of the runway. Corners of contour are selected as feature points. In order to eliminate the interference among points, if the distance of two feature points is less than 3 pixels, both points are deleted before processing. After that, there are 89 feature points in optical image and 103 points in SAR images.



(a) (b)

Fig. 2. Experiment images: (a)optical image, (b)SAR image



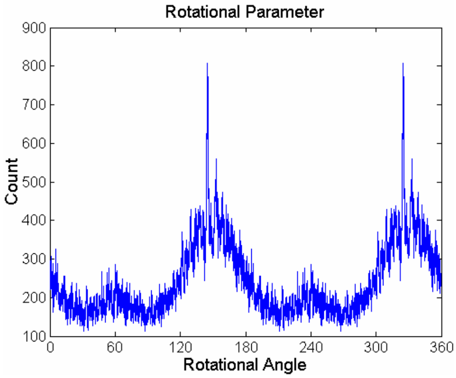


Fig. 3. Result of scale parameter clustering **Fig. 4.** Result of rotational parameter clustering

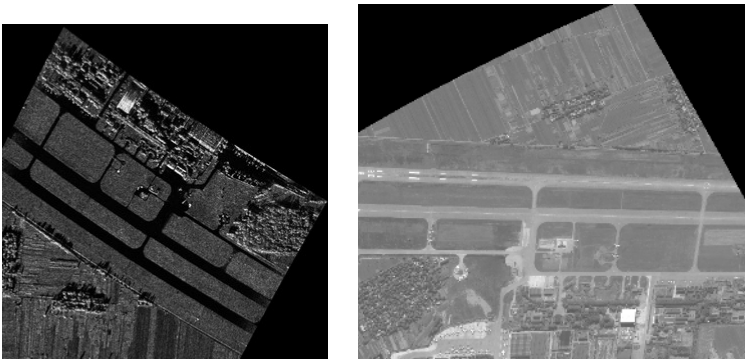
The results of clustering are show in Fig. 3 and 4, it can be seen from which that there is only a peak at 5.5 in scale parameter space, which means that the resolution ratio between optical image and SAR image is 5.5. There are two peaks at rotational result: 148.5° and 328.5°, which have 180° in difference. From convex hull vertices correlation it can be inferred that 148.5° is the correct value, which means if we rotate the optical image by 148.5° clockwise, its direction will approximate the SAR one.

Initially, there are 11 points in SAR image which mapped outside optical image and 5 points in optical image mapped outside SAR image, they are deleted and the procedure of iterative vertice matching begins. The acceptance threshold here is set to 3 and discard threshold is 5. The matching result at each step is listed in Table 1.

It can be seen that there are 65 matched pairs. From the 65 pairs of control points, the mapping parameter can be determined. The Root Mean Square Error(RMSE) is 1.1469. The final registration result is shown in Fig. 5.

Table 1. Matching detail at each step

Step	Reference	Number of Matched Pair	Number of Delete Points
1	Optical	5	6
2	SAR	8	3
3	Optical	8	4
4	SAR	6	3
5	Optical	5	6
6	SAR	5	8
7	Optical	7	8
8	SAR	6	5
9	Optical	8	2
10	SAR	4	0
11	Optical	3	1



(a) (b)

Fig. 5. Registration result: (a)the result of SAR image with optical image as reference, (b) the result of optical image with SAR image as reference

4 Conclusion

In this paper, an automatic image registration method which combines clustering and convex hull matching is proposed. This algorithm uses affine transform as the coarse transform model and applies clustering to determine the scale parameter and rotational parameter candidates. The problem that there are two peaks in rotational parameter space is overcome by convex hull vertices correlation and the correct rotational parameter is determined. With all parameters determined above and the corresponding vertices coordinates, the translational parameter is determined subsequently. Control point matching is achieved by iterative convex hull vertices matching. Experiment results verify that this method can align affine distortion for feature point-based images registration. In addition to remote sensing image processing, this method can be applied into other fields such as medical image processing and computer vision.

References

1. Brown, L.: A Survey of Image Registration Techniques. ACM Computer Surveys, Vol. 24. (1992) 325-376
2. Zitová, B., Flusser, J.: Image Registration Methods: a Survey. Image and Vision Computing, Vol. 21. (2003) 977-1000
3. Ton, J., Jain, A.: Registering Landsat Images by Point Matching. IEEE Trans. Geosci. Remote Sensing, Vol. 27. (1989) 642-651
4. Stockman, G., Kopstein, S., Benett, S.: Matching Images to Models for Registration and Object Detection via Clustering. IEEE Trans. PAMI, Vol. 4. (1982) 229-241
5. Chang, S., Cheng, F., Hsu, W., Wu, G.: Fast Algorithm for Point Pattern Matching: Invariant to Translation, Rotation and Scale Changes. Pattern Recognition, Vol. 30. (1997) 311-320
6. Seedahamed, G., Martueei, L.: Automated Image Registration Using Geometrically Invariant Parameter Space Clustering. ISPRS Photogrammetric Computer Vision Commission III, Symposium (2002) 318-323
7. Shekhar, C., Govindu, V., Chellappa, R.: Multisensor Image Registration by Feature Consensus. Pattern Recognition, Vol. 31. (1999) 39-52
8. Goshtasby, A., Stockman, G.: Point Pattern Matching Using Convex Hull Edges. IEEE Trans. Systems, Man, and Cybernetics, Vol. 15. (1985) 631-637
9. He, Y., Xu, X., Sun, H., Yang, W.: Detection of Airport Runways in Airborne SAR Images. Journal of Wuhan University(Natural Science), Vol. 50. (2004) 393-396

Fingerprint Image Segmentation Based on Gaussian-Hermite Moments

Lin Wang[1], Hongmin Suo[2], and Mo Dai[1]

[1] Institute EGID-Bordeaux 3,
University of Michel de Montaigne - Bordeaux 3,
1 Allée Daguin 33607 Pessac cedex, France
wang@egid.u-bordeaux.fr, dai@egid.u-bordeaux.fr
[2] Department of Mathematics,
Guizhou University for Ethnic Minorities,
Guiyang, 550025, China
gzmysxx88@sina.com

Abstract. An important step in automatic fingerprint recognition systems is the segmentation of fingerprint images. In this paper, we present an adaptive algorithm based on Gaussian-Hermite moments for non-uniform background removing in fingerprint image segmentation. Gaussian-Hermite moments can better separate image features based on different modes. We use Gaussian-Hermite moments of different orders to separate background and foreground of fingerprint image. Experimental results show that the use of Gaussian-Hermite moments makes a significant improvement for the segmentation of fingerprint images.

1 Introduction

An important step in automatic fingerprint recognition systems is the segmentation of fingerprint image. A fingerprint image usually consists of a region of interest (ridges and valleys of fingerprint impression) in the printed rectangular bounding box, smudgy patches, blurred areas of the pattern and the background. We need to segment the fingerprint area (foreground) to avoid false feature extraction due to noisy areas of the fingerprint and the background areas. Accurate segmentation is especially important for the reliable extraction of features like minutiae and singular points.

Several approaches to the segmentation of fingerprint image are known in the literatures. In [1], the fingerprint image is partitioned in blocks of 16×16 pixels. Then each block is classified according to the distribution of the gray level gradients in the block. In [2], this method is extended by excluding blocks with a gray level variance lower than a threshold. In [3], the gray level variance in the direction orthogonal to the orientation of the ridges is used to classify each 16×16 block. In [4], the output of a set of Gabor filters is used as input to a clustering algorithm that constructs spatially compact clusters. In [5], fingerprint images are segmented based on the coherence, while morphology is used to obtain smooth regions. In [6], this method is extended by use of the coherence, the mean and the variance, and an optimal linear classifier is trained for the classification of each pixel.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 446–454, 2005.

In many segmentation algorithms, features extracted cannot completely represent the characteristics of pixels, so they can only identify the blank areas but cannot distinguish foreground from noisy or blurred areas. Though the segmentation algorithms in [5][6] can identify noisy areas, the threshold for coherence is difficult to determine. Moreover, while a meaningful segmentation of fingerprint consists in compact clusters, it is not easy to obtain these compact clusters. In [7], the energy of Gaussian-Hermite moments is first applied in fingerprint image segmentation. In this paper, we present an adaptive algorithm based on the energy and the distribution of Gaussian-Hermite moments for non-uniform background removing in fingerprint image segmentation.

Our paper is organized as follows. First, Section II introduces the Gaussian-Hermite moments and analyzes their behavior. Then, Section III presents the algorithm based on Gaussian-Hermite moments for the segmentation of fingerprint image and Section IV presents some experimental results. Finally, Section V concludes this paper.

2 Gaussian-Hermite Moments

Moments, such as geometric moments and orthogonal moments, are widely used in pattern recognition, image processing, computer vision and multiresolution analysis. In order to better represent local characteristics in noisy images, smoothed orthogonal Gaussian-Hermite moments were proposed [8][9]. Unlike commonly used geometric moments, orthogonal moments use orthogonal polynomes or more complicated orthogonal functions as transform kernels, which produces minimal information redundancy. A detailed study on the different moments and their behavior evaluation can be found in [8][9].

2D orthogonal Gaussian-Hermite moments of order *(p, q)* of an input image $I(x, y)$ can be defined:

$$M_{p,q}(x,y,I(x,y))=\iint_{-\infty}^{\infty}G(t,v,\sigma)H_{p,q}(t/\sigma,v/\sigma)I(x+t,y+v)dtdv \qquad (1)$$

where $G(t,v,\sigma)$ is the 2D Gaussian function, and $H_{p,q}(t/\sigma,v/\sigma)$, the scaled 2D Hermite polynomial of order *(p, q)*, with

$$\begin{aligned}&H_{p,q}(t/\sigma,v/\sigma)=H_p(t/\sigma)H_q(v/\sigma)\\&=\left[(-1)^p\exp(t^2/\sigma^2)(d^p/dt^p)\exp(-t^2/\sigma^2)\right]\times\left[(-1)^q\exp(v^2/\sigma^2)(d^q/dv^q)\exp(-v^2/\sigma^2)\right]\end{aligned}$$

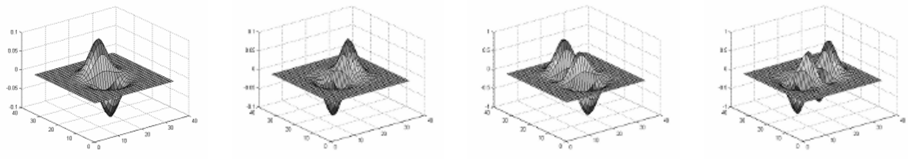


Fig. 1. 2D Gaussian-Hermite base functions (orders: (0,1), (1, 0), (0, 3) and (3, 0))

Obviously, 2D orthogonal Gaussian-Hermite moments are separable, so the recursive algorithm in 1D cases can be applied for their calculation. Fig. 1 shows the spatial responses of the bidimensional Gaussian-Hermite moment kernels of different orders.

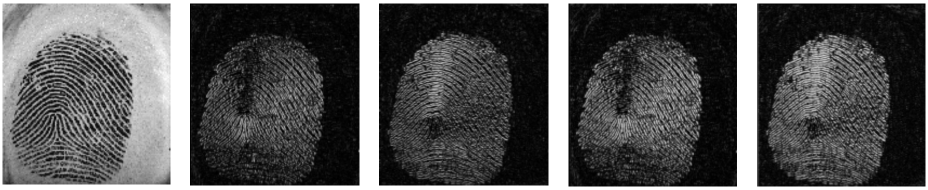
3 Segmentation Based on Gaussian-Hermite Moments (GHM)

3.1 Energy of Gaussian-Hermite Moments (GHM Energy) of a Fingerprint Image

The local intensity surface in fingerprint images is comprised of ridges and valleys whose directions vary smoothly, which constitutes an oriented texture. The gray level in the fingerprint foreground region alternates between black and white. So the Gaussian-Hermite moments in the foreground would vary much more than in the background. In order to characterize this feature of the fingerprint image by the Gaussian-Hermite Moments (GHM), the GHM energies of fingerprint image are defined as:

$$E_{p,q}(x,y)=(M_{p,q}(x,y,I(x,y))^2 \tag{2}$$

where $I(x, y)$ is input fingerprint image and $M_{p, q}$, the Gaussian-Hermite moments of order (p, q) of $I(x, y)$. When p or q is odd, the foreground (fingerprint area) exhibits a very high GHM energy and the background, a very low one. In Fig. 2, we show the GHM energies of a fingerprint image, where $E_{p, q}$=0 is visualized as black, while the maximum GHM Energy, as white.



$I(x, y)$ $E_{1.0}(x, y)$ $E_{0.1}(x, y)$ $E_{3. 0}(x, y)$ $E_{0.3}(x, y)$

Fig. 2. GHM energy images $E_{p, q}(x, y)$ for the fingerprint image $I(x,y)$

Using the orthogonality of the Gaussian-Hermite moments, we take the GHM energy of the Gaussian-Hermite moments $M_{0, 1}$, $M_{1, 0}$, $M_{0, 3}$ and $M_{3, 0}$ to represent the fingerprint image. The major steps are as follows:

- Given a fingerprint image $I(x, y)$ of size $M{\times}N$, calculate the Gaussian-Hermite moments $M_{p,q}(x,y,I(x,y))$ of order (p, q) of the fingerprint image by (1). In our algorithm, we use the moments $M_{0, 1}$, $M_{1, 0}$, $M_{0, 3}$ and $M_{3, 0}$.
- Calculate GHM energies of $I(x, y)$ by (2), which gives $E_{0,1}$, $E_{1,0}$, $E_{0,3}$ and $E_{3,0}$.

- Integrate the GHM energies of different orders to obtain the whole GHM energy:

$$E(x, y)= E_{0,1}(x, y)+ E_{1,0}(x, y)+E_{0,3}(x, y)+ E_{3,0}(x, y) \tag{3}$$

- A low-pass filter is applied for smoothing $E(x, y)$.

In Fig. 3, we show some results of the GHM energy, where black represents $E'=0$, while white, the maximum of E'. We see that in general, E' is very high in the foreground. The histogram of the filtered energy E' is bimodal, so the foreground and the background can be easily segmented.

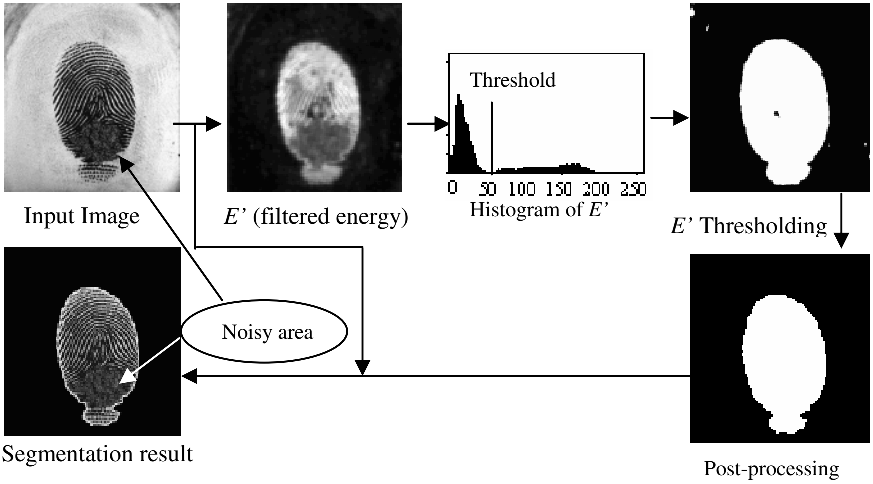


Fig. 3. Segmentation result by the direct use of GHM energy

However the noisy or blurred areas could also have high GHM energy. So even the direct use of the GHM energy can effectively identify the background area, but it sometimes cannot distinguish the foreground from noisy or blurred areas (show Fig. 3). To overcome this problem, we need to add another feature based on the distribution of GHM for the segmentation of noisy or blurred areas. This new feature is the "*coherence*", which will be discussed in the following.

3.2 Coherence of Fingerprint Image Base on GHM

Since in a local region, fingerprint mainly consists of parallel ridge and valley structures, the coherence in a region of interest (ridges and valleys of fingerprint impressions) would be higher than that in noisy areas and blank areas. We define:

$$\begin{cases} M_u=\lambda M_{1,0}+(1-\lambda)M_{3,0} \\ M_v=\lambda M_{0,1}+(1-\lambda)M_{0,3} \end{cases} \tag{4}$$

where λ is the weight associated with the Gaussian-Hermite moments of different orders of the fingerprint image. For each pixel (x, y) of the fingerprint image, we thus obtain a characteristic vector $[M_u, M_v]^T$ by (4). Fig. 4 shows the distribution of $[M_u, M_v]^T$ in a region of interest, a noisy area and a blank area respectively (window size 16×16).

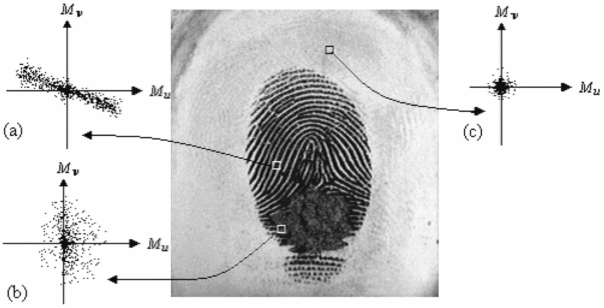


Fig. 4. Distribution of $[M_u, M_v]^T$ in a region of interest (a), a noisy area (b) and a blank area (c). (Window size 16×16)

As is shown in Fig. 4, in a region of interest, the distribution of $[M_u, M_v]^T$ is along the direction orthogonal to the local ridge orientation. However, in a noisy area or a blank area, the distribution of $[M_u, M_v]^T$ is almost a uniform distribution over all directions. So using this behavior of the distribution of $[M_u, M_v]^T$, we can distinguish fingerprint signal area from noisy areas. We use the principal component analysis (PCA) to analyze the distribution of $[M_u, M_v]^T$. The estimate of the covariance matrix $\mathbf{C_M}$ of the vectors $[M_u, M_v]^T$ is given by:

$$\mathbf{C_M}=\begin{bmatrix} \sum_W (M_u-m_u)^2 & \sum_W (M_u-m_u)(M_v-m_v) \\ \sum_W (M_u-m_u)(M_v-m_v) & \sum_W (M_v-m_v)^2 \end{bmatrix} \quad (5)$$

with

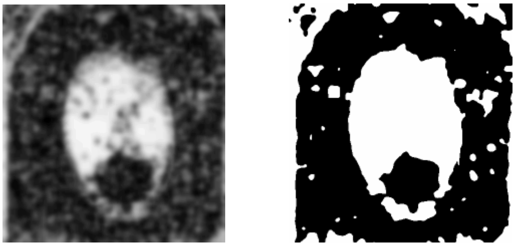
$$m_u=\frac{1}{n\times n}\sum_W M_u \text{ and } m_v=\frac{1}{n\times n}\sum_W M_v$$

where $n\times n$ is the size of the window W.

Let λ_1 and λ_2 ($\lambda_1 \geq \lambda_2$) be the eigenvalues of the covariance matrix $\mathbf{C_M}$. Therefore we can define *coherence* as follows:

$$coherence=\frac{\lambda_1-\lambda_2}{\lambda_1+\lambda_2}=\frac{\sqrt{\left(\sum_W (M_u-m_u)^2-\sum_W (M_v-m_v)^2\right)^2+4\left(\sum_W (M_u-m_u)(M_v-m_v)\right)^2}}{\sum_W (M_u-m_u)^2+\sum_W (M_v-m_v)^2} \quad (6)$$

Fig. 5(a) shows *coherence* estimate calculated from Fig. 4. It can be clearly seen that the coherence is close to 1 in the significant area of the fingerprint, while it is close to 0 in noisy areas. However, in blank area corresponding to the background, there still exist some patches due to the noise. Fig. 5(b) shows the segmentation result of the coherence image after applying an appropriate threshold. Obviously, this segmentation result with a lot of noisy patches in the background is not satisfactory.



(a) Coherence (b) Thresholded coherence

Fig. 5. Coherence image and segmentation result by coherence thresholding for the fingerprint image in Fig. 4

As a brief summary of the analysis above, we see that on one hand, the GHM energy is efficient to separate the regions where the fingerprint is present and the background region, but its direct use cannot well distinguish noisy regions from the meaningful ones (section 3.1); on the other hand, the coherence allows well detecting the noisy patches in the regions where the fingerprint is present, but it suffers from the existence of small patches in the background. That's why we propose the method combining both the GHM energy and the coherence together to improve the segmentation of fingerprint image, which will be explained in the following.

3.3 Segmentation of Fingerprint Image Based on the GHM Energy and the Coherence

In section 3.1 and section 3.2, we discussed on the GHM energy and *coherence* of fingerprint images. Table 1 summarizes the properties of different areas in terms of the GHM energy and *coherence*.

Table 1. Properties different areas in terms of the GHM energy and *coherence*

	GHM energy	Coherence	Coherence × GHM energy
Ridge and valley areas	Very high	Closer to 1	High
Blank area	Very low	<1	Low
Noisy area	High or low	Closer to 0	Low

We can consider the product of the GHM energy and *coherence*. From Table 1, we see that the product of the GHM energy and *coherence* is high in the ridge and valley areas that are not very noisy, while it is low in the blank area and the noisy areas. Using this property, we can easily segment the foreground (ridge and valley areas that are not very noisy) and the background (blank area and noisy areas) of fingerprint images.

We define therefore $CE(x, y)$, the product of the GHM energy and *coherence* as follows:

$$CE(x,y)=coherence(x,y)\times E_N(x,y) \qquad (7)$$

where *coherence*(*x, y*) is defined by (6), with 0≤ *coherence*(*x, y*) ≤ 1 and $E_N(x, y)$, the normalized filtered GHM energy *E* which is defined by (3), with 0≤ $E_N(x, y)$≤255. Because *CE* in the foreground is high and that of the background is low, the histogram of *CE* is bimodal. An adaptive optimal threshold is easily applied to segmenting the foreground and the background. Fig. 6 shows the segmentation result by *CE* thresholding. We see that the noisy fingerprint image of Fig.4 is segmented correctly.

In order to further improve the segmentation result, a post-processing, such as removing small areas and filling small interior holes, can be applied.

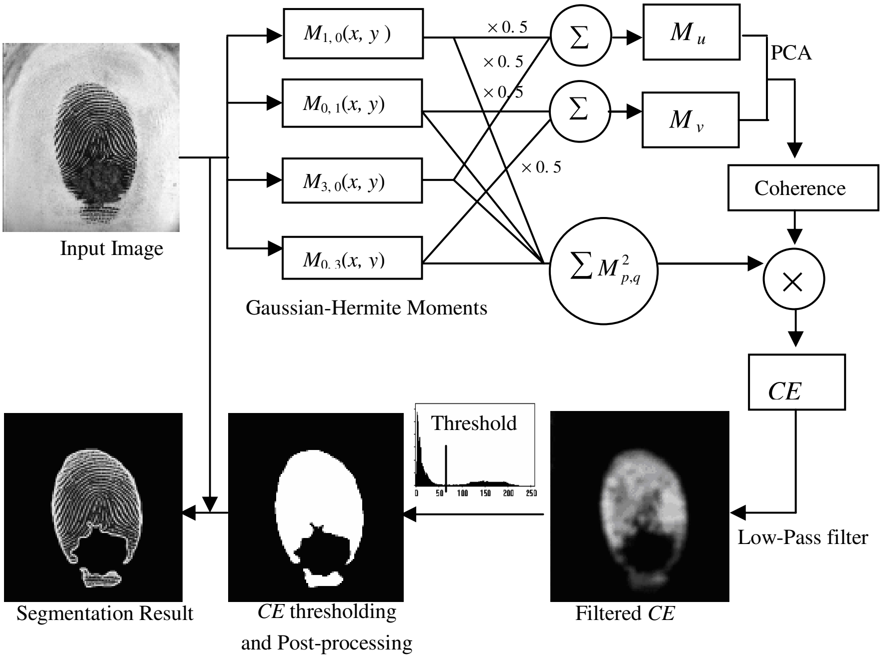


Fig. 6. Flow diagram of our segmentation algorithm

4 Experimental Results

In this section, we present some experimental results of our fingerprint image segmentation algorithm and a comparison with other segmentation algorithms. First, in order to validate the performance of our algorithm, it is tested with Database 3 of the Fingerprint Verification Competition (FVC2000) [10], FVC2002 database 3 and NIST 4 database [11]. In Fig. 7 are shown the segmentation results for three fingerprint images from these Databases. Human inspection can confirm that our algorithm provides satisfactory results.

Now we present the comparison of our algorithm with the segmentation algorithm based on gray level variance in the direction orthogonal to the ridge orientation [3] and the segmentation algorithm based on coherence [5]. We randomly selected 500 fingerprint images in the FVC2000, FVC2002 and NIST 4 databases respectively and used the three algorithms to detect the foreground and the background. The experimental results showed that our algorithm is better than the other two algorithms and is robust in the segmentation of fingerprint image. Fig. 8 shows the resultants for two fingerprint images.

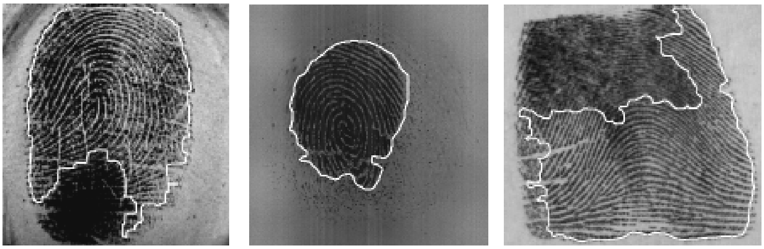


Fig. 7. Segmentation results of some fingerprint images

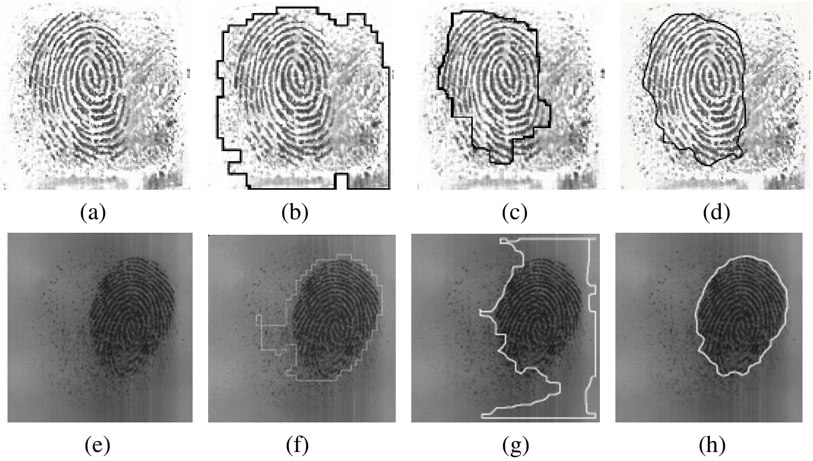


Fig. 8. (a),(e) Original fingerprint images; (b),(f) Segmentation result of algorithm [3]; (c),(d) Segmentation result of algorithm [5]; (d),(h) Segmentation result of our algorithm

In Table 2, we show a brief summary of the performance comparison of the three algorithms.

Table 2. Performance comparison of three algorithms

	Adaptive threshold	Post-processing	Segmentation of ridge/valley areas	Segmentation of blank areas	Segmentation of noisy areas
Our algorithm	Yes	Easy	Good	Good	Good
Algorithm [3]	No	Normal	Good	Good	Bad
Algorithm [5]	No	Difficult	Good	Bad	Good

5 Conclusion

In this paper, a new method based on Gaussian-Hermite moments (GHM) is presented for fingerprint image segmentation. First, the GHM energy is used to distinguish fingerprint areas from the background, allowing to well separating the fingerprint regions and the blank background regions. Second, a PCA-based method estimating the distribution of Gaussian-Hermite moments of different orders is proposed to represent the coherence in a fingerprint image, allowing separating noisy regions. Finally, to integrate the advantages of both indexes, we define the overall feature *CE*, combining the GHM energy and the coherence, to well characterize the fingerprint image, and the segmentation thus consists in the threshold of the *CE* image. By use of the *CE* image, the regions where significant ridge/valley information is present, considered as the foreground, are well distinguished from blank regions and noisy regions, considered as the background. The result of segmentation is thus much improved. Human inspection on experimental results for real fingerprint images shows that the method proposed provides accurate high-resolution segmentation results, which would facilitate the fingerprint image feature extraction and classification afterwards. Comparison between our method and some methods known in the literatures for fingerprint image segmentation shows that, our method not only has a good performance of segmentation but also reduces the influence of noise.

References

1. Mehtre, B.M., Murthy, N.N., Kapoor, S., and Chatterjee, B.: Segmentation fingerprint images using the directional image. Pattern Recognition, Vol. 20 (4) (1987) 429–435
2. Mehtre, B.M. and Chatterjee, B.: Segmentation of fingerprint images - a composite method, Pattern Recognition. Vol. 22 (4) (1989) 381–385
3. Ratha, N., Chen, S., and Jain, A.: Adaptive flow orientation based feature extraction in fingerprint images, Pattern Recognition, Vol. 28 (11) (1995) 1657–1672
4. Jain, A.K. and Ratha, N.K.: Object detection using Gabor filters. Pattern Recognition, Vol. 30 (2) (1997) 295–309
5. Bazen, A.M., and Gerez, S.H.: Directional field computation for fingerprints based on the principal component analysis of local gradients. in: Proceedings of ProRISC2000, 11th Annual Workshop on Circuits, Systems and Signal Processing, Veldhoven, Netherlands, 2000
6. Bazen, A.M., and Gerez, S.H.: Segmentation of Fingerprint Images. in: Proc. RISC 2001 Workshop on Circuits, Systems and Signal Processing, Veldhoven, Netherlands, 2001
7. Wang, L., and Dai, M.: Fingerprint Image Segmentation of Gaussian-Hermite Moments. In: Proc. Advances in Biometric Person Authentication: Sinobiometrics2004, Guangzhou, China, 2004.
8. Shen, J.: Orthogonal Gaussian-Hermite Moments for Image Characterization. in: Proc. SPIE, Intelligent Robots and Computer Vision XVI: Algorithms Techniques, Active Vision, and Materials Handling, Pittsburgh, USA, 1997
9. Shen, J., Shen, W., and Shen, D.F.: On geometric and orthogonal moments. International Journal of Pattern Recognition and Artificial Intelligence, Vol. 14 (7) (2000) 875-894
10. Maio, D., Maltoni, Cappelli, D.R., Wayman, J.L., and Jain, A.K.: FVC2000: Fingerprint verification competition. IEEE Trans. Pattern Anal. Mach. Intell. Vol. 24 (3) (2002) 402-412
11. Watson, C.I. and Wilson, C.L.: NIST Special Database 4, Fingerprint Database. National Institute of Standards and Technology, March, 1992.

HVSM: A New Sequential Pattern Mining Algorithm Using Bitmap Representation

Shijie Song[1,2], Huaping Hu[1], and Shiyao Jin[1]

[1] School of Computer Science, National University of Defense Technology,
Changsha , P.R. China, 410073
[2] The Academy of Equipment Command & Technology,
Beijing, P.R. China, 101416
songshijie87@vip.sina.com
hphu@nudt.edu.cn

Abstract. Sequential pattern mining is an important problem for data mining with broad applications. This paper presents a first-Horizontal-last-Vertical scanning database Sequential pattern Mining algorithm (HVSM). HVSM considers a database as a vertical bitmap. The algorithm first extends itemsets horizontally, and digs out all one-large-sequence itemsets. It then extends the sequence vertically and generates candidate large sequence. The candidate large sequence is generated by taking brother-nodes as child-nodes. The algorithm counts the support by recording the first TID mark (1st-TID). Experiments show that HVSM algorithm can find frequent sequences faster than SPAM algorithm in mining the large transaction databases.

1 Introduction

Finding sequential patterns in large transaction databases is an important problem for data mining. Agrawal introduced the sequential pattern mining problem in [1]. Many methods, which are based on the Apriori property, have been proposed for mining sequential patterns[2, 3, 4, 5, 6, 7]. Apriori principle states the fact that any superseqeuence of a non-frequent sequence must not be frequent. SPADE[8] and PrefixSpan[2] are two algorithms for mining sequential patterns faster than AprioriAll[7].

Jay Ayres and Johannes Gehrke presented SPAM[9] algorithm to find all frequent sequences in a list of transactions quickly. SPAM is a first depth-first search strategy for mining sequential patterns. This algorithm uses a vertical bitmap data layout allowing for simple, efficient counting. S-step/I-step traversal, and S-step/I-step pruning all contribute to the run-time. The mining speed outperforms SPADE and PrefixSpan on large transaction datasets by over an order of magnitude.

We present a first-Horizontal-last-Vertical scanning database Sequential pattern Mining algorithm (HVSM). The algorithm defines the length of sequence as the number of itemset instead of item in a sequence. The improvements include: 1) We first search for SL_1:L_1…L_k by Itemset-extended instead of I-step traversal in SPAM, each element of SL_1 is a "container". Then we extend the "container" by Sequential-extended instead of S-step traversal in SPAM. The method scans database faster than

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 455–463, 2005.

SPAM. 2) We generate candidate large sequences by taking brother-nodes as child-nodes. The pruned large itemsets in the previous layer no longer appeared in the next layer. So, our method is more efficient than S-step/I-step pruning in SPAM. 3) We count support of candidate sequences by scanning bitmap and recording 1st-TID, which reduces the number of scanning and improves counting efficiency.

2 Definition

Nonempty itemset is composed of all items. Sequence is a row of ordered itemsets. Database is a set of tuples (CID, TID, Itemset), where CID is customer-id, TID is transaction-id based on the transaction time. All the transactions with the same CID can be viewed as a sequence of itemsets ordered by increasing TID. Support of a sequence can be represented as the number of CID that contains the sequence. Different from SPAM, we redefine other sequential pattern terms as following:

- *Length of a sequence*: the number of itemset in a sequence.
- *K-large-sequence SL_k*: the sequence whose support is greater than the given minimum support and sequence length is k. SL_k is also called frequent k sequence.
- *Candidate k-large-sequence SC_k*: the candidate sequence of SL_k.
- *One-large-sequence-k-itemset L_k*: which is also called k-large-itemset. It is a one-large-sequence whose itemset size is k.
- *Itemset-extended*: the process of finding L_k.
- *Itemset-node*: it is the base unit for Itemset-extended, which is composed of one-large-itemsets in L_1.
- *Sequence-extended*: the process of finding SL_k.
- *Sequence-node*: it is the basic unit for Sequence-extend, which is composed of one-large-sequences in SL_1.

3 HVSM Algorithm

In this section, we will describe the process of mining lexicographic tree of sequences upon which our algorithm is based. Take the example of Table 1.

Table 1. Raw Database

CID	TID	Itemset
1	1	{a,b,d}
1	2	{b,c,d}
1	3	{b,c,d}
2	1	{b}
2	2	{a,b,c}
2	3	{b,c,d}
3	1	{a,b}
3	2	{a,b,c}
3	3	{b,c,d}

To allow for efficient counting, our algorithm represents the data as a vertical bitmap, which is shown in Table 2.

Table 2. BM-C_1

CID	TID	Itemset			
		{a}	{b}	{c}	{d}
1	1	1	1	0	1
	2	0	1	1	1
	3	0	1	1	1
2	1	0	1	0	0
	2	1	1	1	0
	3	0	1	1	1
3	1	1	1	0	0
	2	1	1	1	0
	3	0	1	1	1

Suppose min_support=2, HVSM algorithm includes two steps:

1) Searching for one-large-sequence SL_1 patterns

The data in Table 2 is just candidate one-large-sequence-one-itemset C_1. By combining L_{k-1} and counting C_k, we obtain SL_1:L_1…L_k, which are the input for mining SL_2…SL_k.

2) Searching for k-large-sequence SL_k patterns

By combining SL_{k-1} and counting SC_k, we obtain SL_2…SL_k.

3.1 Searching for One-Large-Sequence SL_1 Patterns

Counting and generating one-large-sequence-one-itemse: $C_1 \rightarrow L_1$

1) Use a vertical bitmap representation of the Database, the data in Table 2 BM-C_1 is just C_1. Let CID=1, scan itemset {a} of BM-C_1 vertically. If the first 1 appears, record the TID value 1st-TID. If 1st-TID {a}>0, and then support-{a} add 1.
2) Increase CID till end, if support-{a}>=min_support, then {a}$\in$ L_1 and store {a} in Table 3 BM-SL_1, otherwise {a}$\notin$ L_1. Continue with the same process, we will obtain all L_1:{a},{b},{c},{d}.

Generating candidate one-large-sequence-two-itemset $L_1 \rightarrow C_2$

1) C_2 is composed of two items, the 1st-item includes all the Itemset-nodes of L_1, which have the same parent-node *root*. Each node of 1st-item takes its brother-nodes as its child-nodes, which are composed of 2nd-item. E.g. 1st-item {a}'s parent-node *root* has 4 child-nodes {a},{b},{c} and {d}, we take them as child-nodes of 1st-item {a}, which are composed of 2nd-item.
2) We must combine two L_1 by gen() in lexicographic order. E.g., the 1st-item {a} must be ahead of {b}, thus, itemset {b,a} can not be generated by gen(). We obtain all C_2: {a,b}, {a,c}, {a,d}, {b,c}, {b,d}, {c,d}, which are shown in Fig. 1.

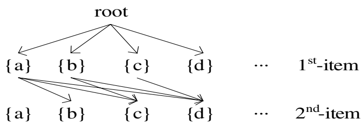


Fig. 1. Candidate one-large-sequence-two-itemset C_2

Counting and generating one-large-sequence-two-itemset $C_2 \rightarrow L_2$

1) Select an itemset from C_2, e.g. {a,b}. Let CID=1, we AND bitmap{a} with bitmap{b} in L_1 of Table 3, and then scan the resultant bitmap{a,b}. If the first 1 appears, record its TID value. If $1^{st}\text{-TID}_{\{a,b\}}>0$, then support-{a,b} add 1.
2) Increase CID till end, if support-{a,b}>=min_support, then $\{a,b\} \in L_2$ and store {a,b} in L_2 of Table 3 BM-SL$_1$, otherwise $\{a,b\} \notin L_2$. Continue with the same process, we obtain all L_2: {a,b}, {a,c}, {b,c}, {b,d}, {c,d}, in which $\{a,d\} \notin L_2$.

Generating candidate one-large-sequence-three-itemset $L_2 \rightarrow C_3$

1) Each node of 2nd-item takes its brother-nodes as its child-nodes, which are composed of 3rd-item.
2) We must combine two L_2 by gen() in lexicographic order. E.g., the 2nd-item {b}'s parent-node 1st-item {a} has 2 child-node {b},{c}, take them as child-nodes of 2nd-item {b}, which are composed of 3rd-item. 2nd-item {b} only has child-node {c} in lexicographic order. We obtain all C_3: {a,b,c}, {b,c,d}, which is shown in Fig. 2.

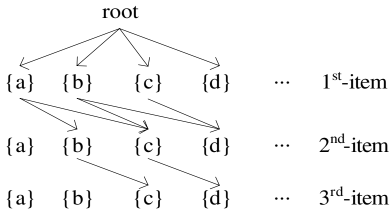


Fig. 2. Candidate one-large-sequence-three-itemset C_3

Counting and generating one-large-sequence-three-itemset $C_3 \rightarrow L_3$

1) Select an itemset from C_3, e.g. {a,b,c}. Let CID=1, we AND bitmap{a,b} with bitmap{a,c} in L_2 of Table 3, then scan the resultant bitmap{a,b,c}. If the first 1 appears, record its TID value. If $1^{st}\text{-TID}_{\{a,b,c\}}>0$, then support-{a,b,c} add 1.
2) Increase CID till end, if support-{a,b,c}>=min_support, then $\{a,b,c\} \in L_3$ and store {a,b,c} in L_3 of Table 3 BM-SL$_1$, otherwise $\{a,b,c\} \notin L_3$. Continue with the same process, we obtain all L_3: {a,b,c}, {b,c,d}.

Generating candidate one-large-sequence-k-itemset $L_{k-1} \rightarrow C_k$

1) Each node of k-1th-item takes its brother-nodes as its child-nodes, which are composed of kth-item.
2) We must combine two L_{k-1} by gen() in lexicographic order.

Counting and generating one-large-sequence-k-itemset $C_k \rightarrow L_k$

1) Select an itemset from C_k. Let CID=1, we AND two corresponding bitmaps itemset (which have the same former k-2 item and could combine two k-1th-items to form C_k), and then scan the resultant bitmap C_k. If the first 1 appears, record its TID value. If 1st-TID$_{Ck}$>0, then support-C_k add 1.
2) Increase CID till end, if support-C_k >=min_support, then $C_k \in L_k$ and store C_k in L_k of Table 3 BM-SL_1, otherwise $C_k \notin L_k$. Continue with the same process, we obtain all L_k.
3) Reserve all the bitmaps and 1st-TIDs of $L_1 \ldots L_k$ in Table 3 BM-SL_1, and take them as input for mining k-large-sequence SL_k.

Table 3. One-large-sequence bitmap BM-SL_1

CID	TID	SL_1										
		L_1				L_2					L_3	
		{a}	{b}	{c}	{d}	{a,b}	{a,c}	{b,c}	{b,d}	{c,d}	{a,b,c}	{b,c,d}
1	1	1	1	0	1	1	0	0	1	0	0	0
	2	0	1	1	1	0	0	1	1	1	0	1
	3	0	1	1	1	0	0	1	1	1	0	1
	1st-TID	1	1	2	1	1	0	2	1	2	0	2
2	1	0	1	0	0	0	0	0	0	0	0	0
	2	1	1	1	0	1	1	1	0	0	1	0
	3	0	1	1	1	0	0	1	1	1	0	1
	1st-TID	2	1	2	0	2	2	2	3	3	2	3
3	1	1	1	0	0	1	0	0	0	0	0	0
	2	1	1	1	0	1	1	1	0	1	1	0
	3	0	1	1	1	0	0	1	1	1	0	1
	1st-TID	1	1	2	2	1	2	2	3	3	2	3

3.2 Searching for k-Large-Sequence Patterns

Generating candidate two-large-sequence $SL_1 \rightarrow SC_2$

1) SC_2 is composed of two level Sequence-nodes, the 1st-level includes all the Sequence-node of SL_1, which have the same parent-node *root*. Each node of 1st-level takes its brother-nodes as itself child-nodes, which are composed of 2nd-level. E.g. 1st-level {a}'s parent-node *root* has 11 child-nodes {a}, {b}, {c}, {d}, {a,b}, {a,c}, {b,c}, {b,d}, {c,d}, {a,b,c} and {b,c,d}, we take them as child-nodes of 1st-level {a}, which are composed of 2nd-level. In which {a,d}$\notin$ SL_1, {a,d} will not appears in the latter level.
2) We must combine two SL_1 by seqgen() in lexicographic order. The result is totally ordered, which is different from the former combination. We obtain all SC_2, which is shown in Fig. 3.

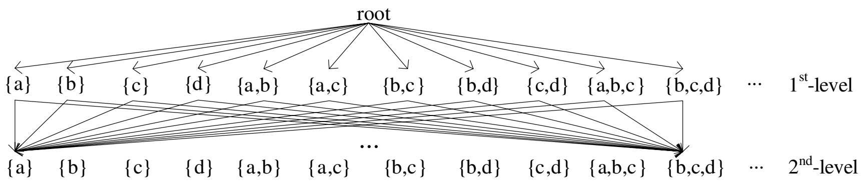


Fig. 3. Candidate two-large-sequence SC_2

Counting and generating two-large-sequence SC_2→SL_2

1) Select a sequence from SC_2, e.g. ({b,c,d}{b,d}). Let CID=1, we take out 1st-$TID_{\{b,c,d\}}$, and then vertically scan the bitmap {b,d} from the location of 1st-$TID_{\{b,c,d\}}$. If the first 1 appears, stop scanning and record its TID value 1st-$TID_{(\{b,c,d\}\{b,d\})}$, then support-{a,b} add 1.
2) Increase CID till end, if support-({b,c,d}{b,d})>=min_support, then ({b,c,d}{b,d})$\in$ SL_2, otherwise ({b,c,d}{b,d})$\notin$ SL_2. Continue with the same process, we obtain all SL_2. Take the example of Fig. 4. Note that {b} and {b,c,d} have different nodes.

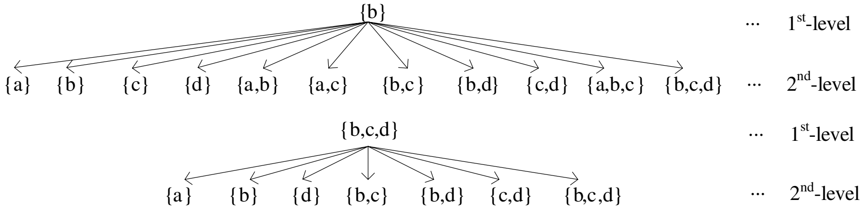


Fig. 4. Two examples of two-large-sequence SL_2

Generating candidate k-large-sequence SL_{k-1}→SC_k

1) Each node of k-1th-level takes its brother-nodes as itself child-nodes, which are composed of kth-level.
2) We must combine two SL_{k-1} by seqgen() in lexicographic order. The result is totally ordered. We obtain all SC_k in the same way.

Counting and generating k-lager-sequence SC_k→SL_k

1) Select a sequence from SC_k. Let CID=1, we take out a SL_{k-1} which has the same former k-1 itemset as SC_k, and then vertically scan the bitmap kth-itemset of SC_k in Table 3 BM-SL_1 from the location of 1st-TID $_{SLk-1}$. If the first 1 appears, stop scanning and record its TID value 1st-TID_{SCk}, and then support-SC_k add 1.
2) Increase CID till end, if support-SC_k >=min_support, then $SC_k\in$ SL_k, here 1st-TID_{SCk} is 1st-TID_{SLk}, otherwise SC_k $\notin$ SL_k. Continue with the same process, we obtain all SL_k.

4 Experimental Evaluation

All the experiments were performed on a 1.0GHz AMD PC machine with 1 gigabyte main memory, running Microsoft Windows 2000. To test our algorithm, we generated numerous synthetic datasets using the IBM AssocGen program[1]. There are several factors that we considered while comparing HVSM against SPAM. These factors are listed in Table 4.

Table 4. Parameters used in dataset generation

Symbol	Meaning
D	Number of customers in the dataset
C	Average number of transactions per customer
T	Average number of items per transaction
S	Average length of maximal sequences
I	Average length of transactions within the maximal sequences

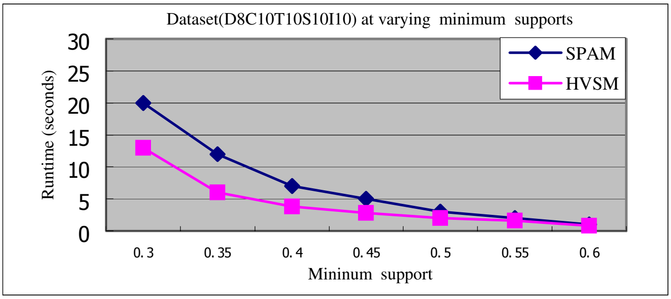


Fig. 5. Varying support for medium-sized dataset

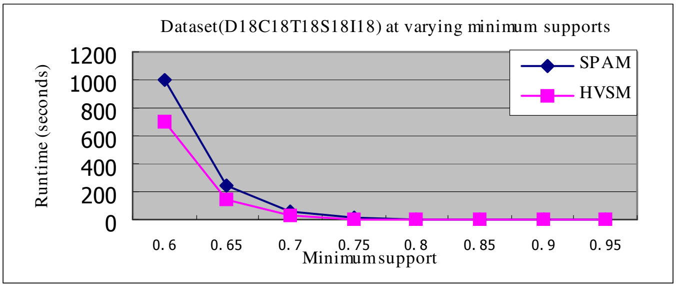


Fig. 6. Varying support for large dataset-

When we increase Number of customers in the dataset D, Average number of transactions per customer C, Average number of items per transaction T, Average length of maximal sequences S, Average length of transactions within the maximal sequences I, the performance of HVSM increases more significantly than SPAM. Fig.5 and Fig.6 show the experimental result in detail, from these figures, we can see that our HVSM algorithm outperforms SPAM algorithm for large transaction database.

There are 4 reasons that lead to the results:

1) Our algorithm defines the length of sequence as the number of itemset instead of item in a sequence. The definition extends the granularity of sequential pattern, which helps to increase mining speed and is convenient for application of sequential pattern mining algorithm.
2) We first search for SL_1:L_1…L_k by Itemset-extended, each element of SL_1 is a "container". Then we extend a "container" instead of an item each time by sequential-extended, which increases mining speed of SL_2…SL_k.
3) Comparing with SPAM, we add a step of generating candidate large sequence in HVSM. We generate SC_k by combining two SL_{k-1}, all the large itemsets pruned in k-1th-level no longer appeared in kth-level.
4) We count support of candidate sequences by scanning bitmap. We only need to vertically scan the bitmap kth-itemset of SC_k in Table 3 BM-SL_1 from the location of 1st-TID_{SLk-1}. We decrease number of scanning and improve counting efficiency with the method.

5 Conclusion

This paper presents a fast algorithm for searching frequent sequence in a large transaction database. It has been applied to Misuse Intrusion Detection[10]. The disadvantage of HVSM is that it consumes more memory than SPAM. Let the number of large itemset in each transaction of database is N, HVSM requires (D*C*N)/8 bytes to store all of the data, while SPAM requires (D*C*T)/8 bytes to store all of the data. We can improve our algorithm by the method of cutting off SC_k whose subsequences do not belong to SL_k after we generate SC_k and before we count them.

References

1. R. Agrawal and R. Srikant.: Mining Sequential Patterns. In: ICDE 1995, Taipei, Taiwan, Mar. 1995.
2. J. Pei, J. Han, B. Mortazavi-Asl, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu.: PrefixSpan mining sequential patterns efficiently by prefix projected pattern growth. In: ICDE 2001, 215–226, Heidelberg, Germany, Apr. 2001.
3. M. Garofalakis, R. Rastogi, and K. Shim. SPIRIT: Sequential pattern mining with regular expression constraints. In: VLDB 1999, 223–234, San Francisco, Morgan Kaufmann, Sept. 1999.
4. C. Bettini, X. S. Wang, and S. Jajodia.: Mining temporal relationships with multiple granularities in time sequences. Data Engineering Bulletin, 21(1) 32–38, 1998.
5. J. Han, G. Dong, and Y. Yin.: Efficient mining of partial periodic patterns in time series database. In: ICDE 1999, 106–115, Sydney, Australia, Mar. 1999.

6. H. Mannila, H. Toivonen, and A. I. Verkamo.: Discovering frequent episodes in sequences. In: KDD 1995, 210–215, Montreal, Quebec, Canada, 1995.
7. R. Agrawal and R. Srikant.: Fast Algorithms for Mining Association Rules in Large Databases. In: Proceedings of the Twentieth International Conference on Very Large Databases, 487–499, Santiago, Chile, 1994.
8. M. J. Zaki. Spade: An efficient algorithm for mining frequent sequences. Machine Learning, 42(1/2) 31–60, 2001.
9. J. Ayres, J. Flannick, J. Gehrke, and T. Yiu.: Sequential pattern mining using a bitmap representation. In: SIGKDD, 429-435, 2002.
10. SONG ShiJie, H.ZunGuo, H.Hua Ping, J.ShiYao.: A Sequential Pattern Mining Algorithm for Misuse Intrusion Detection. In: GCC 2004, 458-465. WUHAN, China, 2004.

HGA-COFFEE : Aligning Multiple Sequences by Hybrid Genetic Algorithm

Li-fang Liu, Hong-wei Huo, and Bao-shu Wang

School of Computer Science and Technology,
Xidian University, Xi'an 710071, P.R. China

Abstract. For multiple sequence alignment problem in molecular biological sequence analysis, a hybrid genetic algorithm and an associated software package called HGA-COFFEE are presented. The COFFEE function is used to measure individual fitness, and five novel genetic operators are designed, a selection operator, two crossover operators and two mutation operators. One of the mutation operators is designed based on the COFFEE's consistency information that can improve the global search ability, and another is realized by dynamic programming method that can improve individuals locally. Experimental results of the 144 benchmarks from the BAliBASE show that the proposed algorithm is feasible, and for datasets in twilight zone and comprising N/C terminal extensions, HGA-COFFEE generates better alignment as compared to other methods studied in this paper.

1 Introduction

Multiple sequence alignment is the most common task in molecular sequence analysis. It has traditionally been used to find characteristic motifs and conserved regions in protein families, in the determination of evolutionary linkage and in the improved prediction of secondary and tertiary structure, so there is on doubt that multiple sequence alignment will remain central to sequence-based biology [1]. Considering their most obvious properties, it is convenient to classify existing multiple alignment algorithms in three main categories: exact, progressive and iterative. The most classical multiple alignment algorithm is multidimensional Needlman-Wunsch algorithm [2], but it is only possible for a maximum of three sequences. A clever algorithm for reducing the volume of the multidimensional dynamic programming matrix that needs to be examined was described by Carrillo and Lipman [3]. This algorithm was implemented in the multiple alignment program MAS [4] that makes it possible to align up to ten closely related sequences. The progressive algorithm was initially described by Hogeweg [5] and later re-invented by Feng [6] and Taylor [7]. ClustalW [8](ClustalX with window interface of ClustalW)is one of the most widely used multiple alignment package that is base on an implementation of progressive algorithm. Over the last years, the increasing use of iterative algorithms such as simulated annealing, genetic algorithm, hidden Markov model and Gibbs sampling, is the trend. The SAGA

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 464–473, 2005.

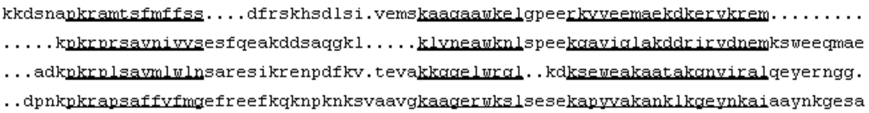
[9] package is based on the genetic algorithm, it uses a total of 22 operators, including semi-hill climbing ones, and applies a dynamic scheduling for selection operators. The PHGA [10] package is base on the parallel hybrid genetic algorithm, it does not solve the multiple sequence alignment problems directly, but converted problem of finding the shortest path in a weighted directed acyclic k-dimension graph.

Under the consistency based objective function for alignment evaluation (COFFEE)[11] model, this paper presents a new hybrid genetic algorithm based method for multiple sequence alignment, five different operators is designed. The efficiency (CPU time and accuracy) of the method was tested by computer simulations and the BAliBASE[12] benchmark tests in comparison with several existing methods. The test result shows that the proposed method is feasible, and the quality of the alignment is better than SAGA, and is comparable to ClustalX and PHGA, especially for datasets in twilight zone and comprising N/C terminal extensions, HGA-COFFEE generates better alignment as compared to other methods studied in this paper.

2 Problem Description

2.1 Multiple Sequence Alignment

Given a finite alphabet set Σ and a family $S = (s_1, s_2, \cdots, s_n)$ of n sequences of various length l_1 to l_n: $s_i = s_{i1}s_{i2}\cdots s_{il_i}$, $s_{ij} \in \Sigma, 1 \leq j \leq l_i, 1 \leq i \leq n$, where for DNA sequences, Σ consists of 4 characters {A,C,G,T}, and for protein sequences, Σ consists of 20 characters of amino acids. An alignment of S is a matrix $A = (a_{ij})$, where $(1 \leq i \leq n)$, $(1 \leq j \leq l)$, $max(l_i) \leq l \leq \sum_{i=1}^{n} l_i$ and satisfying: (1) $a_{ij} \in \Sigma \cup \{.\}$, here "." denoting the gap letters; (2) each row $a_i = a_{i1}a_{i2}\cdots a_{il}(1 \leq i \leq n)$ of A is exactly the corresponding sequence s_i if we eliminate all gap letters; (3) A has no column which only contains gaps. Figure 1 shows an example of a multiple sequence alignment of four protein sequences.



```
kkdsnapkramtsfmffss....dfrskhsdlsi.vemskaagaawkelgpeerkvveemaekdkervkrem.........
.....kpkrprsavnivysesfqeakddsaqgkl.....klvneawknlspeekqaviqlakddrirvdnemksweeqmae
...adkpkrplsavmlwlnsaresikrenpdfkv.tevakkggelwrql..kdksewéakaatakqnviralqeyerngg.
..dpnkpkrapsaffvfmgefreefkqknpknksvaavgkaagerwkslsesekapyvakanklkgeynkaiaaynkgesa
```

Fig. 1. An example of multiple sequence alignment

2.2 Objective Function

Evaluation of the alignments is made using an objective function (OF) which is simply a measure of multiple alignment quality, the most widely used OFs are: WSP (weighted sums of pairs with affine gap penalties) score [2,3,7,9], Hidden

Markov Models (HMMs)[13] and COFFEE [11] score etc. The main limitation of WSP score is that it relies on very general substitution matrices. HMMs describe multiple sequence alignment in a statistical context, using a Bayesian approach. The main drawback of HMMs is that to be general enough, the model requires large numbers of sequences. The COFFEE score reflects the level of consistency between a multiple sequence alignment and a library containing pairwise alignments of the same sequences, the pairwise library can be constituted using the existing pairwise alignment methods. The OF used in this paper is COFFEE score. The COFFEE score works by first generating the pairwise library of the sequences in the alignment and then it calculates the level of identity between the current multiple alignment and the pairwise library. The global score measuring the quality of the alignment is computed by the following formula:

$$COST(A) = [\sum_{i=2}^{n}\sum_{j=1}^{i-1} w_{ij} SCORE(A_{ij})] / [\sum_{i=2}^{n}\sum_{j=1}^{i-1} w_{ij} LEN(A_{ij})] \tag{1}$$

Where n is the number of sequences, A_{ij} is the pairwise projection of sequences s_i and s_j, $LEN(A_{ij})$ is the length of this alignment, w_{ij} is the percent identity between the two aligned sequences s_i and s_j, $SCORE(A_{ij})$ is the overall consistency (level of identity) between A_{ij} and the corresponding pairwise alignment in the library. If an alignment A' satisfying: $COST(A') = max_A COST(A)$, then A' is the optimal alignment. Since the problem of computing optimal multiple alignments according to the COFFEE score is NP-complete[14], usually in practice heuristic methods are used. Paper [15] has a description of recent progresses in multiple sequence alignment. In this paper, the overall approach is to use a measure of multiple alignment quality (the COFFEE score) and to optimize it using a hybrid genetic algorithm.

3 Methods and Algorithm

3.1 Population Initialization

Since an alignment is represented as a matrix, the problem can be solved directly, chromosome is presented as a matrix. A population of the initial parents alignment matrices is produced by the following steps: 1) A random permutation of the numbers $1, 2, \cdots, n$ (representing n sequences) is produced, combine the first $\lfloor (n-1)/2 \rfloor \times 2$ numbers by twos sequentially, and get the corresponding pairwise alignment from the pairwise library, these pairwise alignment are merged sequentially; 2) A random offset is chosen for all the remaining ($\lfloor (n-1)/2 \rfloor \times 2 + 1 \sim n$) sequences and each sequence is moved to the right according to its offset; 3) The sequence are padded with null signs in order to have the same length. The length of each chromosome is limited to $w = \lceil 1.2 \times l_{max} \rceil$,$l_{max} = max(l_1, l_2, \cdots, l_n)$ when population initialization. During the cycles, the length is limited to $w = \lceil 2 \times l_{max} \rceil$, if one individual's length exceed the limitation, then discard this individual. The choice of 1.2 as a scaling factor was based on the observation that solutions to common alignment problems rarely contained more than 20 percent gaps.

3.2 Crossover

Crossover is responsible for combining two different alignments into a new one. We implemented two different types of crossover: one-point and two-point. The one-point crossover combines two parent alignments through a single exchange. Figure 2 outlines this mechanism. The two-point crossover combines two parent alignments through two exchanges. Figure 3 outlines this mechanism. The two newborn children are ranked according to their fitness, only the best of the two children produced that way is kept, if it is not identical to any of the children already generated, it will be put into the new generation, otherwise, it will simply be discarded. This technique helps maintain a high level of diversity in a population of small size, but causes slower convergence.

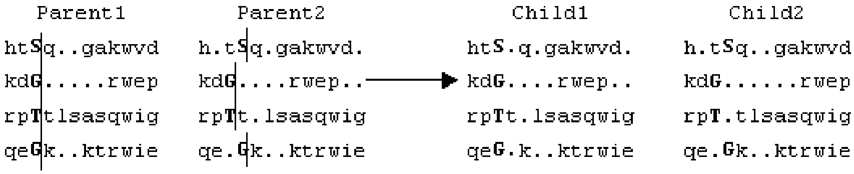


Fig. 2. One-point crossover

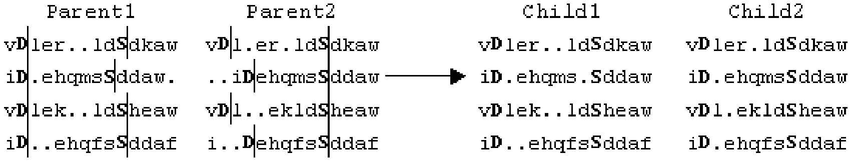
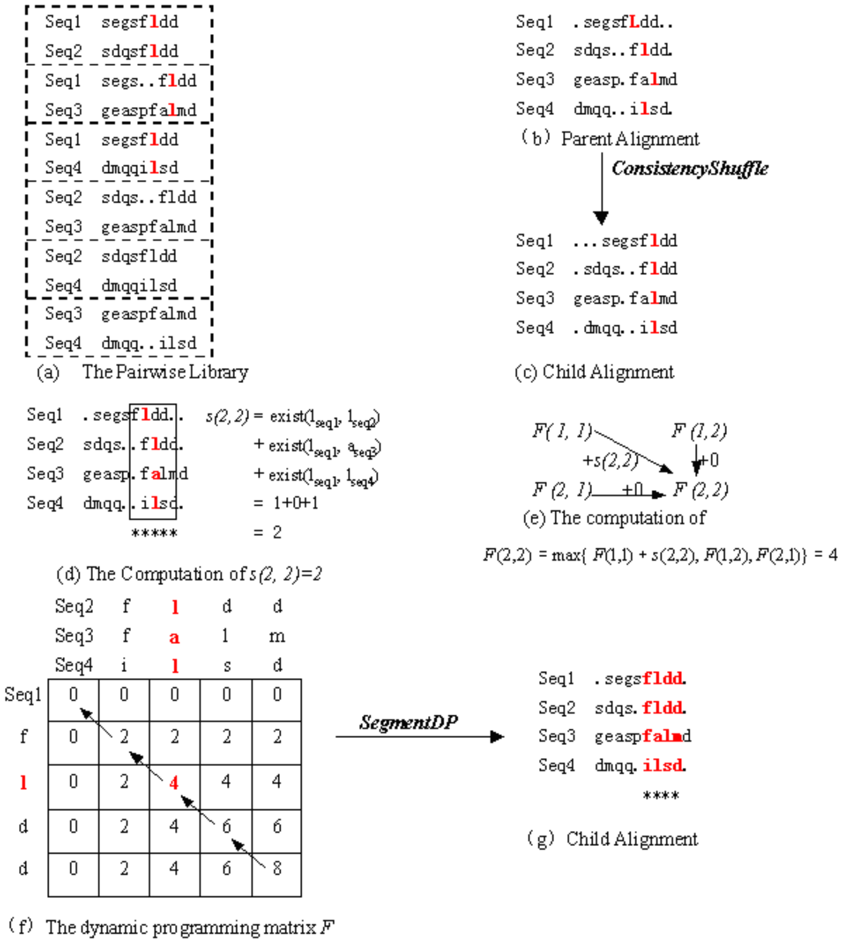


Fig. 3. Two-point crossover

3.3 Mutation

The mutation operator should slightly alter the parent to introduce new genetic information. Based on the idea of using consistency information in a multiple sequence alignment context, two particular mutation operators are designed, namely, [illegible] and [illegible], below is a brief description of the two mutation operators:

The [illegible] operator selects one of the sequences s_i at random from the alignment A , and selects one symbol a_{ij} at random in this sequence, based on the consistency information between A and pairwise library, the other sequences shift to the left or right in order to align with a_{ij}. This process is outlined in Figure 4.



(a) The Pairwise Library

(b) Parent Alignment

(c) Child Alignment

(d) The Computation of $s(2, 2)=2$

(e) The computation of

$F(2,2) = \max\{ F(1,1) + s(2,2), F(1,2), F(2,1)\} = 4$

(f) The dynamic programming matrix F

(g) Child Alignment

(a) The pairwise library. 4 sequences, 6 sequence pairs. (b) Parent alignment for the 4 sequences. (c) The child alignment after ***ConsistencyShuffle*** mutation. The sixth symbol L of sequence 1 is selected, the other sequences shift to the left or right in order to align with L. (d) The computation of s(i, j). The 6~10th columns of the parent alignment is selected, as the asterisks (***) below the alignment, to do ***SegmentDP*** mutation, the 4 sequences are divided into 2 groups, one includes sequence 1, the other includes sequences 2~3. A pair receives a score of 0 if it does not appear in the library or a score of 1 in which it occurs in the pairwise library. (e) The computation of F(i ,j). (f) The dynamic programming matrix F, with arrows indicating traceback pointers. The maximum alignment score is F(4,4). (g) The child alignment after ***SegmentDP*** mutation.

Fig. 4. *ConsistencyShuffle* and *SegmentDP* operators

The [illegible] operator selects a short segment at random from the alignment A, the length of the segment is limited to $2 \leq l \leq 60$. Next, the segment is randomly divided into two groups, with one group having one or two sequences.

That can greatly reduce the computing time when the number of sequences is large. Then, columns with only gap characters are removed from each group, and the dynamic programming algorithm [2] is used to re-align these two groups to a new segment of alignment, the dynamic programming matrix is computed by the following equation

$$F(i, j) = max\{F(i-1, j-1) + s(i, j), F(i-1, j), F(i, j-1)\}. \tag{2}$$

Finally, the new segment is connected to two terminal segments of the parent to complete the offspring, if the newborn child is not identical to any of the children already generated, it will be put into the new generation, otherwise, it will simply be discarded. Since the length of the short segment is limited to $2 \leq l \leq 60$, the computational time for the mutation is bound by a constant, not dependent on the length of sequences of the problem. This process is outlined in Figure 4.

3.4 The Selection

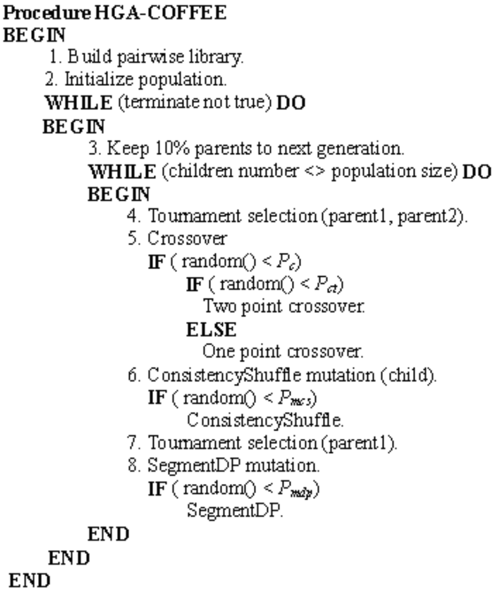
We use two different types of selection: roulette wheel selection and elitist model selection. First, the new generation is directly filled with the fittest individuals from the previous generation (typically 10%), Next the remaining 90% of the individuals in the new generation are created by roulette wheel selecting parents and modifying them.

3.5 The Generation of Pairwise Library

The pairwise library contains a set of pair-wise alignments between all of the sequences to be aligned. The pairwise library is constructed using Needlman-Wunsch[2] algorithm on the sequences, two at a time. For protein sequences, the BLOSUM matrices[16] are selected as the substitute matrix depending on the distance between the two sequences. The range used with the BLOSUM series is: $80 \sim 100\%$(BLOSUM80), $60 \sim 79\%$(BLOSUM62), $30 \sim 59\%$(BLOSUM45), $0 \sim 29\%$(BLOSUM30).

3.6 The Hybrid Genetic Algorithm

The 5 genetic operators are scheduled as the following: Parents are selected based on tournament selection. Crossover is applied with probability Pc using one of the two mentioned crossover operators. Afterward, [illegible] mutation is applied with probability P_{mcs} on the accepted child. Further, selects another parent, [illegible] mutation is applied with probability P_{mdp}. Following this simple process, the fitness of the population is increased until no more improvement can be made or the number of generations exceeded the maximum. All these steps can be summarized in Fig. 5.



```
Procedure HGA-COFFEE
BEGIN
    1. Build pairwise library.
    2. Initialize population.
    WHILE (terminate not true) DO
    BEGIN
        3. Keep 10% parents to next generation.
        WHILE (children number <> population size) DO
        BEGIN
            4. Tournament selection (parent1, parent2).
            5. Crossover
               IF ( random() < $P_c$)
                   IF ( random() < $P_{ct}$)
                     Two point crossover.
                   ELSE
                     One point crossover.
            6. ConsistencyShuffle mutation (child).
               IF ( random() < $P_{mcs}$)
                   ConsistencyShuffle.
            7. Tournament selection (parent1).
            8. SegmentDP mutation.
               IF ( random() < $P_{mdp}$)
                   SegmentDP.
        END
    END
END
```

Fig. 5. Procedure HGA-COFFEE

4 Experimental Results

HGA-COFFEE is implemented in C on a Windows 2000 system. It can be used to align DNA, RNA and protein sequences. The 144 benchmarks from BAliBASE1.0 are used for a comparison with ClustalX1.83, SAGA0.95 and PHGA. The 144 benchmarks are categorized into five different types of references. The description of each reference set is shown in table 1.

The sum-of-pairs score (SPS) and the column score (CS) are used to evaluate the quality of solution from the biological point of view, and BaliScore [12] program is used to calculate SPS and CS score. The SPS score indicates the ratio of pairs correctly aligned while CS score shows the ratio of columns correctly

Table 1. BAliBASE reference sets

Reference	Description	Number
Ref.1	Equidistant sequences of similar length, V1 ($<$ 25% identity),V2 (20 $\sim$ 40% identity) V3 ($>$ 35% identity)	82
Ref.2	Family versus orphans.	23
Ref.3	Equidistant divergent families.	12
Ref.4	N/C-terminal extensions.	15
Ref.5	Insertions.	12

Table 2. SPS and CS scores of various alignment methods for the BAliBASE benchmark tests (SPS/CS)

Methods	HGA-COFFEE	ClustalX	SAGA	PHGA
Ref.1 (82)	0.873/0.812	0.866/0.796	0.767/0.666	0.854/0.779
Ref.2 (23)	0.832/0.346	0.857/0.404	0.786/0.223	0.842/0.364
Ref.3 (12)	0.721/0.457	0.724/0.499	0.607/0.275	0.737/0.448
Ref.4 (15)	0.770/0.504	0.731/0.480	0.546/0.175	0.760/0.415
Ref.5 (12)	0.839/0.670	0.837/0.621	0.758/0.573	0.871/0.746
Avg.1	0.840/0.664	0.836/0.661	0.737/0.510	0.835/0.649
Avg.2	0.807/0.558	0.803/0.560	0.693/0.382	0.813/0.550

aligned. The scores for ClustalX are calculated by executing its software on our system with default parameters, and the scores for SAGA and PHGA are taken from H.D Nguyen's paper[10]. The average results are given in Table 2.

The Avg.1 row shows the values over all 144 benchmarks while the Avg.2 row shows the average values over 5 references.

Reference 1 is also categorized into 3 categories. The detailed results are given in Table 3.

For SAGA and PHGA, the detailed results did not shown in H.D Nguyen's paper.

The results shows that the accuracy of HGA-COFFEE is comparable with that of ClustalX and PHGA, and performs considerably better than SAGA, especially for datasets in twilight zone (< 25% identity) and comprising N/C terminal extensions, HGA-COFFEE generates better alignment as compared to other methods.

For the above 4 alignment methods, ClustalX is based on progressive alignment algorithm, HGA-COFFEE, SAGA and PHGA are base on stochastic iterative algorithms and all base on genetic algorithm (GA). A big disadvantage of

Table 3. SPS and CS scores for the 3 categories in Ref.1(SPS/CS)

Method	V1(23)	V2(31)	V3(28)
	(< 25% identity)	(20 ∼40% identity)	(> 35% identity)
HGA-COFFEE	0.657/0.514	0.947/0.912	0.970/0.947
ClustalX	0.651/0.495	0.928/0.876	0.975/0.955

Table 4. CPU time of GA based methods

Method	HGA-COFFEE (144)	SAGA (141)	PHGA (141)
CPU time (s)	18,244	207,700	31,540
Machine type	A Intel Pentium 4 1 2.8GHz CPU 512M memory	A Sun Ultra 80 4 450-MHz CPUs 1GB memory	A Sun Ultra 80 4 450-MHz CPUs 1GB memory

the GA based methods is the heavy time penalty incurred. The total running time of all benchmarks is shown in Table 4. The data for SAGA and PHGA are taken from H.D Nguyen's paper.

5 Conclusions

HGA-COFFEE is a new GA based method for sequence alignment. The two mutation operators are designed base on the pairwise library and consistency aim of COFFEE, so the global search ability and local search ability are greatly improved.

There is still vast room for improvement in the HGA-COFFEE software. The accuracy of HGA-COFFEE heavily depends on the accuracy of the pairwise library, so with improved accuracy of pairwise library, the accuracy of HGA-COFFEE can also be improved. There are several issues for future work. First, the local information of pairwise sequences will be combined in order to improve the accuracy of the pairwise library, and the computing method in [illegible] operator will be changed due to the global and local information. Second, extend the method to parallel style in order to reduce the computing time.

References

1. T K Attwood, D J Parry-Smith(translated by Luo JingChu)(2002). Introduction to Bioinformatics(in chinese). BeijingPeking University Press2002.
2. Saul B.Needleman and Christlan D.Wunsch(1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J.Mol. Biol*, 48(3)443-453.
3. Carrillo H and Lipman DJ(1988). The multiple sequence alignment problem in biology. *SIAM Appl. Math*, 48(5)1073-1082.
4. Lipman,D. , Altschul,S. , and Kececioglu,J(1989). A tool for multiple sequence alignment. *Proc. Natl. Acad. Sci. USA*, 864412-4415.
5. Hogeweg P and Hesper B(1984). The alignment of sets of sequences and the construction of phylogenetic trees: An integrated method. *J.Mol. Evol*, 20(2)175-186
6. Feng D.F and Doolittle R.F(1987). Progressive sequence alignment as a prerequisite to correct phylogenetic trees. *J.Mol. Evol*, 25(4)351-360.
7. Taylor WR(1988). A flexible method to align large numbers of biological sequences. *J.Mol. Evol*, 28(1-2)161-169.
8. Julie D.Thompson, Desmond G.Higgins, Toby J.Gibson(1994). ACLUSTAL Wimproving the sensitivity of progressive multiple sequence algnment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22(22)4673-4680.
9. Cedric Notredame , Desmond G.Higgins(1996). PSAGAsequence alignment by genetic algorithm. *Nucleic Acids Research*, 24(8)1515-1524.
10. Hung Dinh Nguyen, Ikuo Yoshihara(2002). Aligning multiple protein sequences by parallel hybrid genetic algorithm[A].Genome Informatics 2002[C], Tokyo, Japan: Universal Academy Press, 2002. 123-132.
11. Cedric Notredame , Liisa Holm, Desmond G.Higgins(1998). COFFEEan objective function for multiple sequence alignment. *BIOINFORMATICS*, 14(5)407-422.

12. Julie D.Thompson, Frederic Plewniak, Olivier Poch(1999). A comprehensive comparison of multiple sequence alignment programs. *Nucleic Acids Research*, 27(13)2682-2690.
13. Eddy, S(1995). Multiple alignment using hidden Markov models[A]. Proc. Int. Conf. on Intelligent Systems for Molecular Biology[C], Cambridge, England: AAAI/MIT Press, 114-120..
14. Wang, L and Jiang, T(1994). On the complexity of multiple sequence alignment. *J. Comp. Biol*, 1(4)337-348.
15. Cedric Notredame(2002). Recent progresses in multiple sequence alignmenta survey. *Pharmacogenomics*, 3(1) 131-144.
16. Henikoff, S.and Henikoff, J.G(1992). Amino acid substitution matrices from protein blocks[A]. Proceedings of the National Academy of Sciences of the USA[C], Washington, USA: National Academy of Sciences, 10915-10919.

Independent Component Analysis for Clustering Multivariate Time Series Data

Edmond H.C. Wu and Philip L.H. Yu

Department of Statistics & Actuarial Science, The University of Hong Kong,
Pokfulam Road, Hong Kong
hcwu@hkusua.hku.hk, plhyu@hku.hk

Abstract. Independent Component Analysis (ICA) is a useful statistical method for separating mixed data sources into statistically independent patterns. In this paper, we apply ICA to transform multivariate time series data into independent components (ICs), and then propose a clustering algorithm called ICACLUS to group underlying data series according to the ICs found. This clustering algorithm can be used to identify stocks with similar stock price movement. The experiments show that this method is effective and efficient, which also outperforms other comparable clustering methods, such as K-means.

Keywords: Clustering, Independent component analysis, Statistics, Time series.

1 Introduction

The goal of clustering is to find an intrinsic structure in a set of unlabeled data. As a common data mining technique, clustering is useful in finding suitable groupings and representatives for homogeneous groups, and in detecting unusual data objects. A cluster is therefore a collection of objects which are 'similar' among them and are 'dissimilar' to the data objects belonging to other clusters. For instance, we could be interested in finding groups of stocks with similar return performance from a large database of historical stock prices. Also, we need to consider the scalability and robustness of the clustering algorithms that can deal with high-dimensional data with noises and outliers.

Many clustering algorithms have been developed in the literature. Clustering algorithms can be classified as partitioning methods, hierarchical methods, density-based methods, grid-based methods etc. For partitioning methods, MacQueen [6] first proposed the well-known K-means clustering algorithm. The K-means algorithm first randomly selects k of the objects as cluster centers. The remaining objects will be assigned to a cluster to which it is the most similar by computing the distance between the object and the cluster mean. Then, it recomputes the new mean for each cluster and reassigns the objects to the new clusters. This process will iterate until certain criterion function converges. However, a disadvantage of K-means is that the clustering results will be influenced by noises or outliers in the data.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 474–482, 2005.

Many statistical techniques have been used in various data mining algorithms. A recently developed linear transformation method is the Independent Component Analysis (ICA) [3], in which the desired representation is the one that minimizes the statistical dependence of the components of the representation. Such a representation can capture the essential structure of the data in many potential applications. In this paper, we investigate ICA for clustering applications.

The rest of this paper is organized as follows: In Section 2, we will introduce the independent component analysis technique. Then, in Section 3, we will propose a clustering model for time series data by using independent component analysis. In Section 4, we give some experimental results on testing the effectiveness and scalability of this clustering model with artificial time series data and real financial datasets. Finally, we give some conclusions in Section 5.

2 Independent Component Analysis

ICA [1,3,5] is a statistical method aiming to express the observed data in terms of a linear combination of underlying latent variables. The latent variables are assumed to be non-Gaussian and mutually independent. The task is to identify both the latent variables and the mixing process. A typical ICA model is:

$$X = AS \tag{1}$$

where $X = (x_1, ..., x_m)$ is the vector of observed variables, $S = (s_1, ..., s_m)$ is the vector of statistically independent latent variables called the independent components, and A is an unknown constant mixing matrix. The independent components S in the ICA model (1) are found by searching for a matrix W such that $S = WX$ up to some indeterminacies.

The FastICA algorithm [2,4] is a computationally efficient and robust fixed-point type algorithm for independent component analysis and blind source separation. The iterative fixed-point algorithm for finding one unit is:

$$\tilde{w}_{n+1} = E\{x(w_n x) * g(|w_n x|^2)\} - E\{g(|w_n x|^2) + |w_n x|^2 g'(|w_n x|^2)\}w_n \tag{2}$$

where $w_{n+1} = \frac{\tilde{w}_{n+1}}{\|\tilde{w}_{n+1}\|}$. Getting the estimate of w, we can obtain an IC by $s = wx$.

The above algorithm can be extended to the estimation of the whole ICA transformation $S = WX$. To prevent converging to the same ICs, the outputs $w_1 x, ..., w_n x$ are decorrelated after every iteration. When we have estimated n independent components, or n vectors $w_1, ..., w_n$, we run the one-unit fixed-point algorithm for w_{n+1}, and after every iteration step subtract from w_{n+1} the projections of the previously estimated n vectors, and then renormalize w_{n+1}:

$$\tilde{w}_{n+1} = \tilde{w}_{n+1} - \sum_{j=1}^{n} w_j w_j' \tilde{w}_{n+1}. \tag{3}$$

where $w_{n+1} = \frac{\tilde{w}_{n+1}}{\|\tilde{w}_{n+1}\|}$. The above decorrelation scheme is suitable for deflationary separation of the ICs. Using FastICA, we can estimate A and S from observations X, where $A = W^{-1}$. The vectors $w_1, ..., w_n$ compose W, i.e., $W = [w_1; ...; w_n]$.

3 The Clustering Model

3.1 The Idea of Applying ICA for Time Series Clustering

Although ICA has successful applications in some areas, such as signal processing, there is little work on applying ICA methods for clustering time series data.

First of all, ICA model is suitable for time series analysis. We can regard time series as observed signals from different sources. Given a set of n time series $\{x_i(t)\}$, $i = 1, ..., n$, at each time step t, we can assume that each time series follows a mixing process $x_i(t) = a_{ij}s_j(t)$, $j = 1, ..., m$. The sources $\{s_j(t)\}$ are statistically independent. This denotation is exactly the ICA model $X(t) = AS(t)$. Let us see whether the mixing matrix A and the ICs reveal some useful information of the underlying structures of the time series by an example below.

The first row of Fig 1 lists three different time series sources with 300 observations: Sine $S_1(t) = 0.5 + 0.4 \times sin(2 \times pi \times t/20)$, Random noise within the range [0,1] $S_2(t) = rand(0, 1)$, Ramp $S_3(t) = 0.5 + .4 \times mod((2 \times pi \times t), 1)$. The second row is the mixing time series: $X_1 = 0.8 \times S_1 + 0.2 \times S_2$, $X_2 = 0.8 \times S_1 + 0.2 \times S_3$, $X_3 = 0.8 \times S_3 + 0.2 \times S_1$. After whitening X_1, X_2 and X_3, we obtain $\bar{X}_1$, $\bar{X}_2$, $\bar{X}_3$ in the third row. The last row shows the ICs $\bar{S}_1$, $\bar{S}_2$, $\bar{S}_3$ found by using ICA. It can be seen that the shapes of original sources can be well recovered, so $\bar{S}_1$, $\bar{S}_2$, $\bar{S}_3$ are good estimates of S_1, S_2 and S_3, respectively.

We further check the values in A (see the 3×3 matrix below). For instance, $\bar{X}_1 = 0.962 \times \bar{S}_1 + 0.27 \times \bar{S}_2 - 0.0234 \times \bar{S}_3$. From the loadings (weighting), we can see that $\bar{S}_1$ is the most dominant IC of $\bar{X}_1$, which is consistent with the fact that S_1 is the most important driving force of X_1. Similar results can be found in $\bar{X}_2$. We notice that time series X_1 and X_2 are much similar while X_3 is not. It is interesting that the difference of time series patterns can be reflected by the loadings of ICs. This key finding motivates us to apply ICA in cluster analysis. Besides noises, we observe that ICA is also robust to outliers or scales in data.

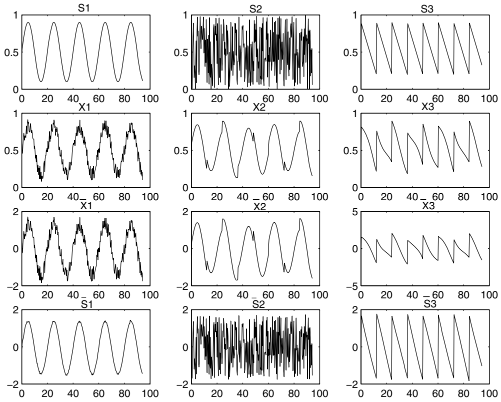


Fig. 1. Clustering Time Series with Noises

$$\begin{pmatrix} \bar{X}_1 \\ \bar{X}_2 \\ \bar{X}_3 \end{pmatrix} = \begin{pmatrix} 0.962 & 0.270 & -0.0234 \\ 0.988 & 0.0304 & 0.154 \\ 0.381 & 0.0300 & 0.924 \end{pmatrix} \begin{pmatrix} \bar{S}_1 \\ \bar{S}_2 \\ \bar{S}_3 \end{pmatrix}$$

3.2 The Algorithm

In this article, we propose an algorithm called ICACLUS for clustering time series data by using ICA. Our goal is to group different time series according to their unique structures expressed by the corresponding loadings of ICs. First, we will use FastICA to find the ICs from the time series. We set the default number of ICs to be found equals to the number of time series. After finding the ICs, we select some of the ICs that are dominant to each time series. The selection criterion is based on the loadings of these ICs to the time series. Since the values of loadings can be positive and negative, we choose a certain percentage (e.g., 15%) of positive loadings and negative loadings with the largest absolute values, respectively. Then, the corresponding ICs will be the dominant ICs for a particular time series. After finding the dominant ICs for all the time series, we can make a comparison of the dominant ICs of these time series and perform clustering based on their similarity (e.g., all the dominant ICs matched should be in one cluster). Finally, we can group all the time series to different clusters.

We summarize the main steps of ICACLUS algorithm as follows:

1. Import n time series $X_1, ..., X_n$ with m observations.
2. Use FastICA to compute k ICs and corresponding mixing matrix A.
3. For each X_i, sort its ICs by the loadings and select $C\%$ of ICs with highest postive loadings and $C\%$ of ICs with most negative loadings ($0 < C\% \leq 50\%$) and then determine the $C\% \times k$ positive dominant ICs and $C\% \times k$ negative dominant ICs.
4. Set a similarity threshold $t \leq 2C\% \times k$, if X_i and X_j $(i \neq j)$ has at least t dominant ICs matched, then X_i and X_j group in a cluster. (Note: varying the threshold t can control the number of clusters generated, any X_i can belong to multiple clusters.)
5. Output the clustering results.

4 Experimental Results

4.1 Clustering Performance Analysis

In order to validate the performance of the ICA-based clustering method, we design a cluster validation experiment as follow: first, we use three data sources ([illegible], [illegible], ([illegible]), and [illegible] math functions) to generate different kinds of time series with each 300 observations in the ranges from -1 to 1. We called these sources S_1, S_2, S_3. For each source, we generate 100 time series by adding noises which follow [-0.25,0.25] uniform distribution, 100 time series by adding random outliers ranging from -2.5 to 2.5, 100 time series by adding different scales ranging from -0.25 to 0.25, respectively. The 900 time series are denoted by $X_1, X_2, ..., X_{900}$. Based on these time series, we also generate 300 mixture time series by adding time series from *Sine*, *Ramp* and *Atan* with equal weights, i.e., $X_{Mixture} = (X_{Sine} + X_{Ramp} + X_{Atan})/3$. Finally, we obtained 1,200 time series which can be classified into four groups: C_{Sine}, C_{Ramp}, C_{Atan} and $C_{Mixture}$. Thus, we have known the cluster labels of the total 1,200 time series.

In this experiment, we use two frequently used clustering performance measures *Recall* and *Precision* for validation. They are defined as follows:

$$Recall = \frac{a}{a+c} \tag{4}$$

$$Precision = \frac{a}{a+b} \tag{5}$$

where a is the number of correctly predicted objects in the predicted cluster, $a+b$ is the number of objects in the predicted cluster, $a+c$ is the number of objects in the actual cluster.

As a combination of *Recall* and *Precision*, *F-measure* is defined as:

$$F_{measure} = \frac{2 \times Recall \times Precision}{Recall + Precision} \tag{6}$$

A higher value of F-measure implies a better overall performance. We employ these three measures to validate and evaluate the effectiveness of ICACLUS by comparing with K-means. Here, we set K=4 for K-means. For ICACLUS, we set the number of ICs to be found is 300 and the number of dominant ICs selected is 20 (10 for Top 10 positive loadings and 10 for Top 10 negative loadings). The criterion that two objects in one cluster is that they have at least 15 dominant ICs matched. Four non-overlapping clusters are also found at this setting.

Table 1 shows the clustering evaluation results. Comparing with *Recall*, *Precision* and *F-measure*, we find that ICACLUS always outperforms K-means, especially in the time series from mixture sources. It can be explained that ICA can recover the underlying data structures while K-means only focuses on the distance measure (we use the Squared Euclidean distance for K-means). In the cases there are noises or outliers in data, the performance of K-means will be greatly affected. However, ICA can detect the similar patterns with noise, outliers and even with different scales. Therefore, we can see that ICACLUS is very effective for clustering time series data according to their underlying patterns.

Table 1. Clustering Evaluation Results for 1,200 Time Series from Four Clusters

Methods	ICACLUS	K-means	ICACLUS	K-means	ICACLUS	K-means
Clusters	**Recall**	**Recall**	**Precision**	**Precision**	**F-measure**	**F-measure**
Sine (300)	100%	76.3%	100%	69.6%	100%	72.8%
Ramp (300)	100%	80.0%	100%	70.6%	100%	75.0%
Atan (300)	100%	100%	100%	79.6%	100%	88.6%
Mixture (300)	100%	33.3%	100%	64.9%	100%	44.1%

4.2 Scalability Analysis

In the following experiments, we test the scalability of ICACLUS. We first generate a 500 × 500 data matrix which represents 500 time series, each has 500 observations. Then, we use this synthetic dataset to compare the performance of the ICACLUS with K-means by varying the data sizes. For each specified data size, we carry out the datasets 10 times and record the average running time. We fix the length of time series or number of observations at 250 in each experiment. Here, we set K=10 for K-means and the number of ICs equals to number of time series n for ICACLUS. Also, $C = 20\%$ and $t = 0.8 \times C \times n$.

Fig 2 shows the average running time for various length of time series. In Fig 2, we notice that when the number of observations is larger than 200, ICACLUS is faster than K-means. Fig 3 shows the average running time for various number of time series. We can see that when the number of objects increase, the increase of running time in K-means clustering grows exponentially whereas ICACLUS is much more computationally efficient, especially when the number of data objects is large. From above results, we can conclude that the scalability of ICACLUS is satisfactory, thus this technique can be used in large-scale clustering applications.

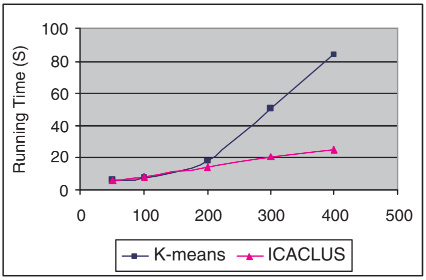


Fig. 2. Increasing Length of Time Series

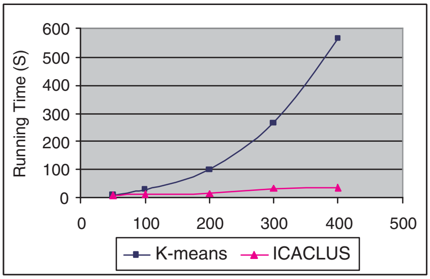


Fig. 3. Increasing # of Data Objects

4.3 A Real Application in Financial Market Analysis

In this research, we use the historical stock prices obtained from Hong Kong Exchange and Cleaning Ltd for clustering stocks. We selected 8 years daily stock prices of 26 HSI Constituent Stocks in Table 2 (Note: the datasets are available at http://finance.yahoo.com/). The daily return $r_i(t)$ are calculated by $r_i(t) = log(p_i(t)) - log(p_i(t-1))$, where $p_i(t)$ is the closing price of stock i on the trading day t. These stocks can be classified into [illegible], [illegible], [illegible], and [illegible] & [illegible] sectors according to Hang Seng Index criterion (refer to http://www.hsi.com.hk/). We would like to identify the underlying stock grouping structures based on the stock returns.

First, we use the daily returns data of the 26 stocks during the four-year period from Jan 3, 1994 to Dec 30, 1997, so we obtain 26 time series, each with around 1,000 observations. For each stock, 3 dominant ICs with most

Table 2. List of the 26 Constituent Stocks of Hang Seng Index

No.	Abbrev.	FINANCE	No.	Abbrev.	COMMERCE & INDUSTRY
5	HSBC	HSBC Holdings	4	WH	Wharf (Holdings) Ltd.
8	HSB	Hang Seng Bank Ltd.	7	PCCW	Pacific Century CyberWorks Ltd.
14	BEA	Bank of East Asia, Ltd.	10	HW	Hutchison Whampoa Ltd.
No.	**Abbrev.**	**UTILITIES**	12	SP	Swire Pacific Ltd.
2	CLP	CLP Holdings Ltd.	17	CMH	China Merchants Holdings Co. Ltd.
3	GAS	HK & China Gas Co. Ltd.	18	JEH	Johnson Electric Holdings Ltd.
6	HKE	HK Electric Holdings Ltd.	19	DM	Denway Motors Ltd.
No.	**Abbrev.**	**PROPERTIES**	20	CP	CITIC Pacific Ltd.
1	CK	Cheung Kong (Holdings) Ltd.	21	CRE	China Resources Enterprise, Ltd.
9	HLD	Henderson LD Co. Ltd.	22	CPA	Cathay Pacific Airways Ltd.
11	SHK	Sun Hung Kai Properties Ltd.	23	EH	Esprit Holdings Ltd.
13	WLC	Wheelock and Co. Ltd.	24	LF	Li & Fung Ltd.
15	HI	Henderson Investment Ltd.	25	YYI	Yue Yuen Industrial (Holdings) Ltd.
16	HLP	Hang Lung Properties Ltd.	26	LG	Lenovo Group Ltd.

positive loadings and 3 dominant ICs with most negative loadings are selected from the total 26 ICs found. The clustering results by using ICACLUS are shown in Fig 4. The most similar stocks (6 ICs matched) are labeled as ’×’, and the stocks with 5 ICs matched are labeled as ’.’. In Fig 5, we find some interesting results. The stocks in a square form a cluster which represents these stocks have 6 ICs matched. Two squares with lines connected form larger clusters which represent the stocks in these squares have at least 5 ICs matched. We notice that banking stocks HSBC(#5) and HSB(#8) are in one cluster. Because HSB is a group member of HSBC, so this fact may explain why these two companies have more similarity. Let us see the cluster including HLD(#9), HW(#10), SHK(#11), WLC(#13), CK(#1), BEA(#14) and HI(#15). Although HW and BEA are not classified into properties stocks, these two companies do own many property related business (e.g., shopping centers, property mortgage). Therefore, we can regard this cluster as a properties-related cluster. Another cluster includes CLP(#2), GAS(#3), WH(#4), HKE(#6), HLD(#9), HW(#10), SHK(#11) and WLC(#13). CLP, GAS and HKE are the utilities and also energy stocks. From the company profiles, we discover that the other companies in the cluster own a large proportion of business in public transportation, logistics and infrastructure, no matter in Hong Kong or in China. This can be explained that such kinds of business heavily rely on oil and other energy resources. Therefore, we can regard this cluster as a energy related cluster. We also note that the stocks HLD, HW, SHK and WLC are the overlaps of the two clusters.

From the clustering results, we can see that PCCW (#7), the largest IT stock in HK, is not so similar to other stocks. ICACLUS also reveals that this stock has unique stock price movement at the time periods which is consistent with the facts. From the clustering results, we validate that the companies geared in the same business or with mutual business interdependence show certain tendency

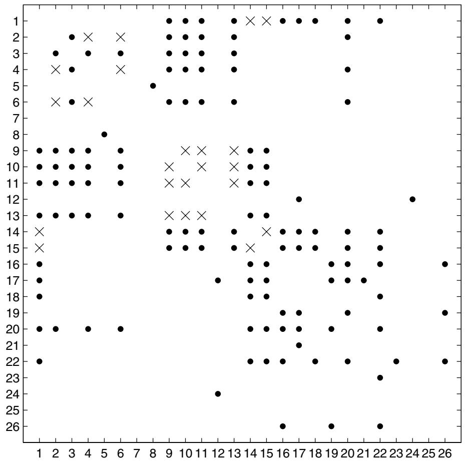


Fig. 4. Clustering Results Using ICA

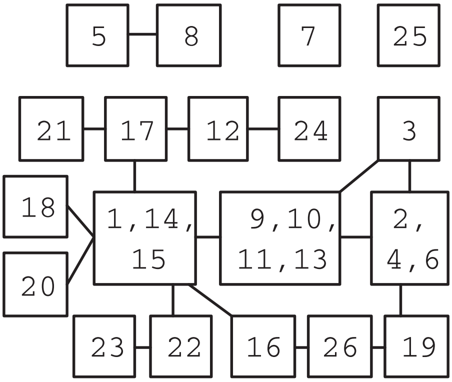


Fig. 5. Representation of Clusters

Table 3. Clustering Results Using K-means

Clusters	C1	C2	C3	C4	C5	C6	C7
K=2	1–6, 8–16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26	7					
K=3	1–6, 8–16, 17, 18, 20, 21, 22, 23, 24, 25, 26	19	7				
K=4	1–6, 8–16, 18, 20, 22, 23, 24, 25	17, 21, 26	19	7			
K=5	1–6, 8–16, 18, 20, 22, 23, 24, 25	17, 21	19	26	7		
K=6	1–6, 8–16, 20, 22	18, 23, 24, 25	19	26	17, 21	7	
K=7	1–6, 8–16, 20, 22	18, 24, 25	19	26	17, 21	23	7

to cluster in homogeneous groups. Table 3 shows the clustering results using K-means. Comparing with ICACLUS, we find that K-means can not well identify the sector information and the interdependence of stocks because many stocks are grouped in a single cluster. We remark that there is no a priori assumptions or domain knowledge on the grouping criterion during the ICA clustering process, so the clusters are objectively generated.

5 Conclusions

In this paper, we explore a new approach for clustering by using independent component analysis. The proposed ICACLUS algorithm can generate overlapping clusters, which is also more effective and efficient for clustering large time series datasets than traditional clustering methods, such as K-means. The experimental results show that ICA is a potentially powerful and robust technique for clustering multivariate time series and providing fresh insights of the underlying driving mechanisms of the data.

References

1. P. Comon, Independent component analysis: a new concept?" Signal Processing 36, 287-314, 1994.
2. A. Hyvarinen 1999,Fast and robust fixed-point algorithms for independent component analysis, IEEE Transactions on Neural Networks 10(3), 626-634.
3. A. Hyvarinen, J. Karhunen, E. Oja, *Independent Component Analysis*, New York, J. Wiley, 2001.
4. A. Hyvarinen and E. Oja, A fast fixed-point algorithm for independent component analysis, Neural Computation 9, 1483-1492, 1997.
5. C. Jutten and J. Herault, Blind separation of sources, part I: An adaptive algorithm based on neuromimetic architecture, Signal Processing 24, 1-10, 1991.
6. J. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. 5th Berkeley Symp. Math. Statist, Prob.*, 1:281-297, 1967.

Applying Fuzzy Neural Network to Intrusion Detection Based on Sequences of System Calls*

Guiling Zhang and Jizhou Sun

Department of Electronic Information Engineering, Tianjin University
glzhang808@sohu.com

Abstract. Short sequences of system calls have been proven to be a good signature description for anomalous intrusion detection. The signature provides clear separation between different kinds of programs. This paper extends these works by applying fuzzy neural network (FNN) to solve the sharp boundary problem and decide whether a sequence is "normal" or "abnormal". By using threat level of system calls to label the sequences the proposed FNN improves the accuracy of anomaly detection.

1 Introduction

An intrusion detection system (IDS) is an important component of computer and information security framework. Some of the earliest works in intrusion detection were performed [12] in the early 1980s. Intrusion detection techniques are generally classified into two categories: anomaly detection and misuse detection.

Anomaly detection suffers from accuracy problems [11], as building an accurate model (avoiding false negatives) may not fully reflect the complex dynamic nature of computer systems (leading to false positives).

S. Forrest and coworkers reported that short sequences of system calls in running processes generate a stable signature for normal behavior [4,5,6]. The signature provides clear separation between different kinds of programs. These results are significant because most prior published work on intrusion detection is relied on either a much more complex definition of normal behavior or on prior knowledge about specific form of intrusion.

W. Lee groups formulated *Machine Learning* tasks on operating system call sequences of normal and abnormal (intrusion) executions of the Unix *sendmail* program [7,8]. Their results indicate that machine learning can play an important role in intrusion detection services.

Z. Liu and et al. present a comparison of two different encoding methods of sequences of system calls for three types of neural networks [3]. They demonstrated that neural networks are able to analyze sequences of system calls, and they can be

* This research was supported by the National High Technology Development 863 program of China under Grant No. 2002AA142010 and Tianjin Key Science Item.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 483–490, 2005.

used to deploy an intrusion detection system. In paper [10], they reported the results of a comparative study of various different methods of detecting anomalies in sequences of system calls by neural networks.

In [1,2,10], the authors applied neural network to intrusion detection with many useful results.

In [9], the authors showed that fuzzy logic is appropriate for the intrusion detection problem because security itself involves fuzziness.

Fuzzy logic and neural networks are complementary technologies. The hybrid approach in this paper is to combine them into an integrated system which will have the advantages of both neural networks and fuzzy systems. This technique will be used to learn system or user behaviors and detect anomalies.

The rest of the paper is organized as follows: In section 2, how to select the fuzzy neural network is discussed. Section 3 details the user behavior description method and introduces the classification of system calls according to their threat levels. In this section, coding method for input dataset is also presented. Some experiments and results are presented in section 4. In section 5 some conclusions are given.

2 The Structure of Fuzzy Neural Network for Intrusion Detection

2.1 The Back Propagation Network

The back propagation network has been used successfully in other intrusion detection studies [1,2,10]. The back propagation network, or *backprop*, is a standard feed-forward network. It is the most commonly used neural network. *Backprops* are well suited for applications in classification, function approximation, and prediction.

2.2 Fuzzy Neural Network

Neural networks are good at recognizing patterns and classification, but they are not good at explaining how they reach their decisions. Fuzzy logic systems, which can reason with imprecise information, are good at explaining their decisions but they cannot automatically acquire the rules they use to make those decisions. The fuzzy neural network combines fuzzy logic and neural network, which has both advantages of neural network and fuzzy logic.

The structure of FNN used in this paper is presented in fig.1. It has real weight coefficients and it's input signal is fuzzy. Input layer has k neural cells, in this paper, which represent k position features (i.e. k system calls identification number) respectively; Fuzzification layer transforms crisp input value to fuzzy subjection function value by selecting appropriate membership function, and then transmits them into network. There are many different types of membership functions for fuzzification layer [3], such as larger fuzzy distribution (S type), smaller fuzzy distribution (Z type) and medium fuzzy distribution (πtype) . In this paper, for a larger threat level number is assigned to more dangerous system call (see section 3), we adopt larger fuzzy distribution function (S type) as follows:

$$S(x:a,b)=\begin{cases}0 & x\leq a\\ 2(\frac{x-a}{b-a})^2 & a<x\leq\frac{a+b}{2}\\ 1-2(\frac{x-a}{b-a})^2 & \frac{a+b}{2}<x\leq b\\ 1 & x>b\end{cases}$$

The hidden layer is handled by using non-linear method and *Sigmoid* function. The output layer has one cell and outputs are in (0,1). In supervised learning, we set a threshold (e.g. 0.5) and if an output is bigger than the threshold it will be labeled as the corresponding "abnormal", otherwise it will be labeled as "normal".

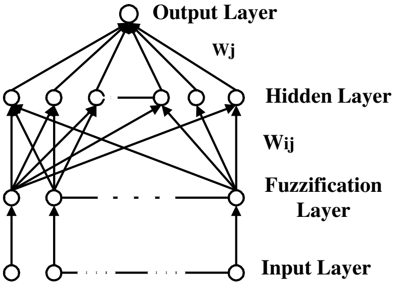


Fig. 1. The structure of fuzzy neural network

We use BP algorithm and the outputs of each layer are the following:

Output layer: $o = f(net), net = \sum_{j=0}^{m} w_j y_j$

Hidden layer: $y_j = f(net_j), net_j = \sum_{i=0}^{n} w_{ij} x_i$ Where, j=1, 2,…, 1.

In the two formulas above, the *f(x)* is the *Sigmoid* function, $f(x) = 1 \big/ 1+e^{-x}$.

The learning procedure is as follows: During training process an input sequence is put into the network and flows throw the network generating a value on the output cell. Then the actual output is compared with desired target, and a match is computed. If the output and target match (in the same range), no change is made to the net. However, if the output differs from the target (in the different range) a change must be made to some of the connections. There are some error functions that adjust weight value. This paper selects the following error function to train the FNN, for by comparing, this error function has more accurate result than others [3]:

$$E = \frac{1}{2}\sum_{k=1}^{l}(t_k - o_k)^2$$

Where, t_k is expected output and o_k is practical output. The adjusting mode is as follows:

If j is output unit: $\Delta w_j = -\eta \frac{\partial E_d}{\partial w_j} = -\eta \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_j}$

$$= (t_j - o_j)o_j(1-o_j)y_j = \eta\delta_j y_j \quad \text{Where, } \delta_j = (t_j - o_j)o_j(1-o_j)\,.$$

If j is hidden unit: $\Delta w_{ij} = -\eta \frac{\partial E_d}{\partial w_{ij}} = -\eta \frac{\partial E_d}{\partial net_j}\frac{\partial net_j}{\partial w_{ij}}$

$$\frac{\partial E_d}{\partial net_j} = \sum_{k \in Downstream(j)} \frac{\partial E_d}{\partial net_k}\frac{\partial net_k}{\partial net_j} = \sum_{k \in Downstream(j)} -\delta_k w_{kj} o_j(1-o_j)$$

set $\delta_j = o_j(1-o_j) \sum_{k \in Downstream(j)} \delta_k \; w_{kj}$, then $\Delta w_{ij} = \eta\delta_j x_{ij}$

In the formula above,$\eta$is a learning parameter, on beginning of training it has the larger value, then it decrease quickly[3]. According to this principle, we also setη=$c_2(1\text{-}t/t_m)$, c_2 is a constant value and in [0,1], we set it 0.5, t_m is the maximal training number in advance, t is the t_{th} training.

Table 1. System Call Categories

Groups	Threat level	System calls
File system	4(401 to 415)	chmod, chown chown32, fchmod, fchown, fchown32, lchown, lchown32, link, mknod, mount, open, rename, symlink, unlink
	3(301 to 315)	close, create, dup2, flock, ftruncate, ftruncate64, ioctl, mkdir, nfsservctl, quotactl, rmdir, truncate, truncate64, umount, umount2
	2(201 to 223)	chdir, chroot, dup, fchdir, fcntl, fcntl64, fsync, llseek, lseek, newselect, poll, pread, putpmsg, pwrite, read, readv, select, sendfile, umask, utime, afs syscall, write, writev
	1(101 to 127)	Other file system calls
Process	4(416 to 434)	execve, setfsgid, setfsgid32, setfsuid, setfsuid32, setgid, setgid32, setgroups, setgroup32, setregid, setregid32, setresgid, setresgid32, setresuid, setresuid32, setreuid, setreuid32, setuid, setuid32
	3(316 to 336)	Vfork, adjtimex, brk, clone, exit, fork, ioperm, iopl, kill, modifyldt, nice, ptrace, reboot, sched_setparam, sched_setscheduler, sched_yield, setpriority, setrlimit, vhangup, vm86, vm86old
	2(224 to 231)	capset, personality, prctl, setpgid, setsid, uselib, wait4, waitpid
	1(128 to 156)	Other process calls
Network	4(435 to 440)	accept, bind, connect, listen, socket, socketpair
	3(337 to 344)	recv, recvfrom, revcmsg, sedmsg, send, sendto, setdocketopt, shutdown
	1(157 to 159)	Getpeername, getsocketname, getsocketopt
Module	4(441 to 442)	Init_module, create_module
	3(345)	deletemodule
	1(160 to 161)	Get_kernel_system, query_module
Signal	3(345 to 360)	All signal calls
Others	3(361 to 376)	Alarm, madvise, madvisel, mlock, mlockall, pivot_root, setdomainname, sethostname, setitimer, settimeofday, stime, swapoff, swapone, sysctl, syslog, ugetrlimit
	2(232 to 242)	Mincore, mmap, mmap2, modify_ldt, mprotect, mremap, munlock, munlockall, munmap, nanosleep, security
	1(162 to 180)	Break, ftime, getitimer, gettid, gettimeofday, gtty, idle, lock mpx, msyncm, pause, prof, profile, sty, sysi, time, times, ulimit, uname

3 Descriptions of User Behaviors Based on Sequences of System Calls

3.1 System Calls Classification According to Threat Level

The system calls is classified in six categories according to their functionality in Table 1 [13]. Each call category is further classified into four groups according to their threat level, i.e. 1(Harmless), 2(Used for subverting the invoking process), 3(Used for a denial of service attack), 4(Allows full control of the system). In this paper, larger threat level numbers are assigned to more dangerous system calls.

According to the table, the top threat level 4 has 42 calls, the next threat level 3 has 76 calls, the threat level 2 has 42 calls and the threat level 1 has 80 calls. We assign identifying number 401, 402,•••, 442 to each calls in threat level 4, 301,302,•••, 376 to each calls in threat level 3, 201, 202,…242 to each calls in threat level 2, 101, 102, 180 to each calls in level 1, respectively,(see table 1).

3.2 Preparing Training Dataset

S. Forrest provided us with set of traces of the *sendmail* program used in her experiments [6]. The *sendmail* program is sufficiently varied and complex to provide a good initial test, and there are several documented attacks against *sendmail* that can be used for testing, so if we are successful with *sendmail* we conjecture that we will be successful with many other privileged Unix processes. In the generating *sendmail* traces detailed in [6], each file of the trace data has two columns of integers, the first is the process ids and the second is the system call "numbers". These numbers are indexed into a lookup table of system call names. For example, the number "5" represents system call open. But we instead these "numbers" by "threat level identifying numbers" we defined in table 1, e.g. using "412" instead of "5" for system call "open". The training set of traces includes both "normal"(80%) and "abnormal" (20%) sequences [8].

In order to prepare the training datasets, we use a sliding window to scan the normal traces and to create a list of unique sequences of system calls (The size of the sliding window may be 2l+1, e.g. 7, 9, 11, 13, etc). This list is called the "normal" list. Next, we scan each of the intrusion traces. For each sequence of these system calls, we first look it up in the normal list. If an exact match can be found then the sequence is labeled as "normal". Otherwise it is labeled as "abnormal". The fuzzy neural network learns from these "normal" and "abnormal" pre-labeled system call sequences. This paper creates four groups labeled dataset according to sliding windows size 7, 9, 11 and 13, respectively.

4 Experiment Results

4.1 Training the FNN

Using the system calls threat level numbers in table 1 we encode the sequences in training datasets above. These encoded sequences are used as input data to train the fuzzy neural network. If the length of the sliding window is 7, 9, 11, 13, the FNN will have 7, 9, 11, 13 input node respectively. Then, the fuzzification layer of the FNN transforms the input sequences into fuzzy values by using S membership function. We select various numbers of hidden nodes in our experiments. Many parameters of the fuzzy neural network may be adjusted in the training procedure.

The weights of the FNN with which are usually randomly set to begin, are then adjusted by the network. Two thresholds will be selected in the learning process: One (≤0.5, e.g. 0.4) is for "normal" sequences and the other (≥0.5, e.g. 0.6) is for "abnormal". We select 0.4 and 0.6 for the two thresholds. The difference between 0.4 and 0.6 may produce more accurate weights because the S type membership function and larger identification number assigned for high threat level system calls are adopted in our system.

When using normal traces to train the FNN, we compare its outputs with the threshold 0.4 and adjust the corresponding weights until the outputs are less than or equal to 0.4. If an output is greater than 0.4 then we compute the errors. Errors are then propagated back through the system and cause the system to adjust the weights that control our network by learning parameter η.

The abnormal traces dataset has two different sort data, "normal" and "abnormal". In the training process, if the sequences are normal then the weights adjustment is similar to using normal traces dataset. Otherwise, when using "abnormal" system call sequences to train the FNN, we adjust the weights as follows. We compare the outputs with the threshold 0.6 and adjust the weights until the outputs are greater than or equal to 0.6. If an output is less than 0.6 then we compute the errors. Errors are then propagated back through the system and cause the system to adjust the weights that control our network by learning parameterη until the outputs are greater than or equal to 0.6.

The training process may spend long terms. After the FNN was trained well, we use the FNN to determine whether one sequence is the "normal" or "abnormal".

4.2 Experiment Results

After the FNN has been well trained, we prepare test datasets to examine its intrusion detection ability. The test datasets contains both "normal" and "abnormal" sequences from *sendmail* traces but *does not* include training dataset. The original *sendmail* datasets were downloaded from http://cs.unm.edu. We produce three groups dataset for the evaluation. These test datasets contain 80%, 70%, 60% "normal" and 20%, 30%, 40% "abnormal" sequences, respectively. Fig.2 and Fig.3 show us the average results of these three groups datasets by running the FNN technique.

The FNN can be used to decide whether a sequence is "abnormal" or "normal". But what the system needs to know is whether the trace being analyzed is an intrusion or not. We use Lee's post-processing scheme to detect whether a given trace is an intrusion [8]:

> Use a sliding window of length 2l+1, e.g., 7,9,11,13, etc., and a sliding (shift) step of l (in our experiments, l is selected as 3,4,5,6 respectively), scan the test data and each sequence is as input of FNN. If corresponding output is greater than or equal to 0.6 it will be labeled as "abnormal" and if corresponding output is less than or equal to 0.4 it will be labeled as "normal".

For each of the (length 2l+1) *regions* generated in Step 1, it will be labeled as "normal" or "abnormal" regions according to the FNN outputs.
If the percentage of *abnormal regions* is above a threshold value(say 2%) then the trace is an intrusion.

Figures 2 demonstrate that the error rate curves of the FNN with different hidden node numbers. Corresponding to input node length k=7, 9, 11, and 13, The FNN can get lower error rate when hidden node number is between 10 and 25. So we can obtain that the good hidden node number of the FNN is likely selected from 10 to 25.

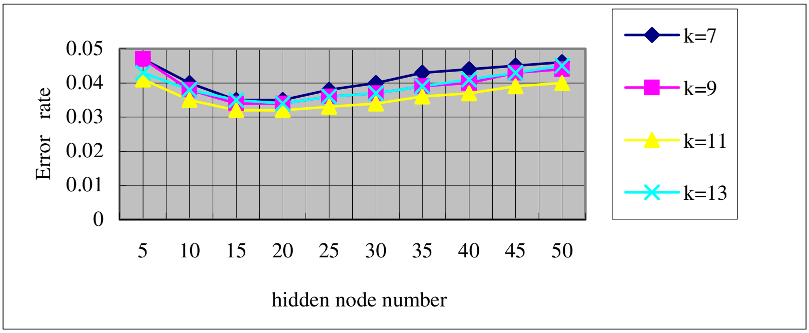


Fig. 2. The error rate curve (input node k=7, 9, 11, 13)

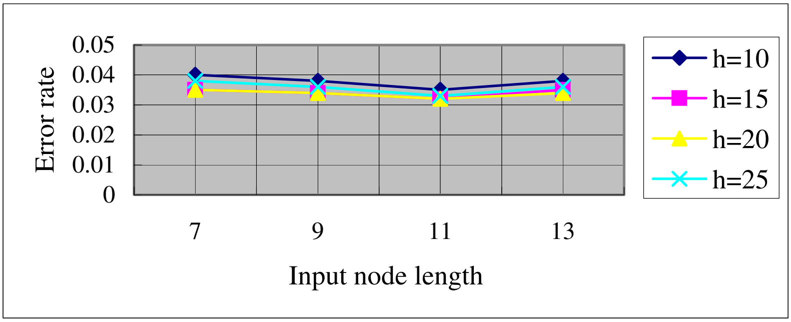


Fig. 3. The error rate curve (Hidden node number k=10, 15, 20, 25)

Figures 3 demonstrate that the error rate curves with the different input node length. Corresponding to hidden node number h=10, 15, 20, and 25, The FNN can get lower error rate when the input node length is 11.

5 Conclusions

Many evidences have demonstrated that analyzing sequences of system calls is an effective method for intrusion detection. In this paper we develop fuzzy neural network technique to intrusion detection system based on sequences of system calls.

By using classification of threat level for each system call, the new version of user behavior description based on sequences of system calls has been developed. It is used as input data coding for the fuzzy neural network intrusion detection system. When the FNN is well trained, it will decrease affection of low threat level sequences and concentrate on the high threat level sequences so as to improve the accuracy of intrusion detection. The primary experiments show that the hybrid system is much accurate if we select appreciate input node number and hidden node number.

However, there are some problems need to be studied in the future. For example, we will make comparison with other traditional algorithm in accuracy, efficiency and complexity, etc.

References

[1] A. K. Ghosh, Aaron Schwartzbart, Michael Schatz: Learning Program Behavior Profiles for Intrusion Detection, Proceedings 1st USENIX Workshop on Intrusion Detection and Network Monitoring, Santa Clara, California, 1999

[2] Z. Liu, G. Florez , and S.M. Bridges: A Comparison of Input Representations In Neural Networks: A Case Study in Intrusion detection. International Joint Conference on Neural Networks (IJCNN), Honolulu, Hawaii, 2002

[3] F. Yuan, H. Wu, and Ge Yu: Web Users' Classification Using Fuzzy Neural Network. Knowledge-Based Intelligent Information and Engineering Systems: 8th International Conference, Kes 2004, Wellington, New Zealand, September, 2004

[4] C. Warrender, S. Forrest, and B. Pearlmutter: Detecting Intrusions Using System Calls: alternative data models. IEEE Computer Society 1999.

[5] S.A. Hofmeyr, S. Forrest, and A. Somayaji: Intrusion detection using sequences of system calls. Journal of Computer Security, 1998.

[6] S. Forrest, S.A. Hofmeyr, A. Somayaji, and T.A. Longstaff: A sense of self for unix processes. In Proceedings of the 1996 IEEE Symposium on Security and Privacy, Los Alamitos, CA, 1996. IEEE Computer Society Press.

[7] Wenke Lee, Sal Stolfo, and Phil Chan: Learning patterns from Unix process execution traces from intrusion detection. AAAI Workshop: AI Approaches to Fraud Detection and risk management, July 1997.

[8] W. Lee and S. Stolfo: Data Mining Approaches for Intrusion Detection. Proc. Of the Seventh USENIX Security Symposium, January, 1998.

[9] G. Florez, SM. Bridges, Vaughn RB: An Improved Algorithm for Fuzzy Data Mining for Intrusion Detection. Annual Meeting of The North American Fuzzy Information Processing Society Proceedings, 2002.

[10] Z. Liu, S.M. Bridges, R.B. Vaughn: Classification of anomalous traces of privileged and parallel programs by neural networks. Proceedings of The 12th IEEE International Conference on Fuzzy Systems 2003.

[11] T. Verwoerd, R. Hunt: Intrusion detection techniques and approaches. Computer Communications, 25(15), SEP 15 2002.

[12] C.C. Michael: Finding the Vocabulary of Program Behavior Data for Anomaly Detection. DARPA Information Survivability Conference and Exposition, VOL 1, Proceedings, 2003.

[13] M. Xu, C. Chen, J. Ying: Anomaly Detection Based on System Call Classification. Journal of Software, China, VOL. 15 2004.

Design and Implementation of Web Mining System Based on Multi-agent

Wenbin Hu and Bo Meng

Computer Application Department, College of Computer,
Wuhan University, 430079, Hubei, China
{hwb77129, mengbo}@126.com

Abstract. Some challenges for website designers are to provide correct and useful information to individual user with different backgrounds and interests, as well as to increase user satisfaction. Most existing Web search tools work only with individual users and do not help a user benefit from previous search experience of others. In this paper, a collaborative Web Mining System, Collector Engine System is presented, a multi-agent system designed to provide post-retrieval analysis and enable across-user collaboration in web search and mining. This system allows the user to annotate search sessions and share them with other users. The prototype system and component of Collector Engine System is discussed and described, and especially designs the web Agent, the knowledge discovery of web Agent is extracted based on a combination of web usage mining and machine learning. The system model is established and realized by J2EE technology. The system's application shows that subjects' search performances are improved, compared to individual search scenarios, in which users have no access to previous searches, when they have access to a limited of earlier search session done by other users.

1 Introduction

With the increasing amount of information available online, the World Wide Web has rapidly become the main source of competitive intelligence for businesses, and consequently search engines represent invaluable decision support tools. However, disciplines like information retrieval and machine learning are facing serious difficulties in dealing with the Web in a scalable way. Researchers [1] have developed many different techniques, which address this challenging problem of locating relevant Web information efficiently. Examples of such techniques include Web search engines [2], meta-searching, post-retrieval analysis, and enhanced Web collection visualization.

A major problem with most such techniques is that they do not facilitate user collaboration, which has potential for greatly improving Web search quality and efficiency. Without collaboration, users must start from scratch every time they perform a search task, even if other users have done similar or relevant searches.

In this paper, we propose a multi-agent approach for collaborative information retrieval and Web mining, implemented in a system called Collector Engine System (CES). We also propose an integrated intelligent agent-based framework to accom-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 491 – 498, 2005.

modate the deficiency of contemporary agent software platforms such as IBM Aglets and ObjectSpace Voyager Agents, which mainly focus on multi-agent mobility and communication, the framework provides an ingenious layer to support different AI functionalities to the multi-agent applications.

2 System Architecture of CES

In order to improve search effectiveness and efficiency, CES is presented, which establishes a collaborative Web information mining and mining environment that performs in-depth post-retrieval analysis.

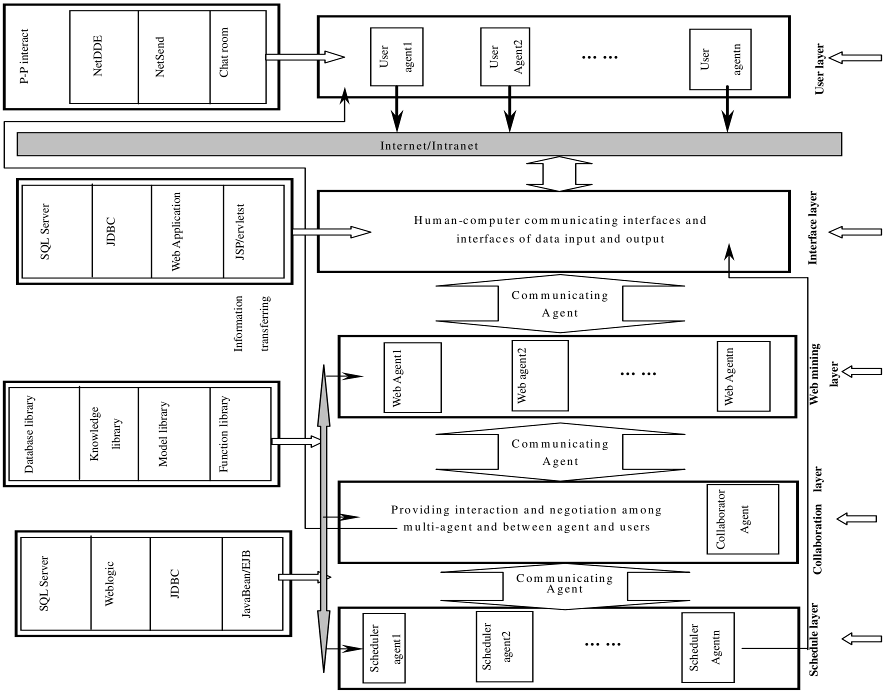


Fig. 1. The architecture of CES

The system architecture is shown in Fig.1. CES consists of three types of software agents, namely, User Agent, Collaborator Agent, Scheduler Agent and Web Agent. In a typical system setup, each individual user will have his or her own personalized User Agent. Each user group (e.g., all participants of a research project or members of a new product design team) will share one Collaborator Agent, one Scheduler Agent and one Web Agent. The User Agent is mainly responsible for retrieving pages from the Web, performing post-retrieval analysis, and interacting with the users. The Collaborator Agent facilitates the sharing of information among different User Agents. The Scheduling Agent Keep a list of monitoring tasks and is responsible for carrying out these tasks based on users' schedules. Web Agent is responsible for mining the data and filter the trashy information. This architecture differs from traditional information retrieval system or recommended system in that collaboration is based on users' searches and analysis, not Web Pages rated, a news article viewed, or items purchased.

3 Knowledge Discovery of Web Agent

A crucial issue in the design and implementation of a Web Agent is construction of its knowledge base [3]. We address this issue from the viewpoint of machine learning and data mining. In particular, we use association-mining method to find associated Web pages, and propose a new algorithm LCSA (Leader cluster algorithm, C4.5 machine learning algorithm, and Web Structure for Adaptive Website) for Web page clustering. We propose an algorithm called LCSA to generate useful knowledge for a Web Agent. Figure 2 gives a schematic description of Algorithm LCSA. The structure tree *T* is basically the physical tree structure of Web page file. It is assumed that a Website designer put pages of similar nature under the same subdirectory. The description *D* provides additional semantic information for each Web page. It is assumed that *D* is provided by a Website designer. The LCSA algorithm first constructs page clusters based on Web log data by using the Leader clustering algorithm, and then filters out the uninteresting clusters by considering semantic information provided by *T* and *D*. The rules for describing clusters are generated using the C4.5 algorithm. A similar approach has been used to discover conditional association rules. The results of LCSA are semantically related clusters of Web pages that can be used to build an adaptive Website.

```
Get: log file L, Web structure tree T, and description D for nodes of T.
Out: Semantically related clusters of Web pages.
1. Clean L to generate cleaned data CD.
2. Identify user sessions in CD to produce the set of session S.
3. Run Leader algorithm on S to generate a set C of clusters.
4. Create input for the C4.5 algorithm from clusters of the Leader algorithm:
for each cluster c in C
   for each page P in c
      Derive the complete set P' of prefixes
               From the pathname of page P;
   Use k shortest prefixes in P' as
               Condition attributes for a C4.5 training instance;
   Use the name of c as the decision attribute
   for the C4.5 training instance.
5. Run the C4.5 algorithm to generate a decision tree DT.
6. Combine decision tree DT, description D, and Web structure tree T to derive
semantically related clusters of Web page.
```

Fig. 2. LCSA Algorithm of Web Agent

4 CES Model

4.1 Establishing CES Model

CES is an evolutionary multi-agent system in which each agent in a population of peers adapts to its local information environment by learning to estimate the value of hyperlinks, while the population as a whole attempts to cover all promising areas through selective reproduction [4]. Figure 3 shows the representation of each agent running mode of CES. The agent interacts with the information environment that consists of the actual networked collection plus information kept on local data structures.

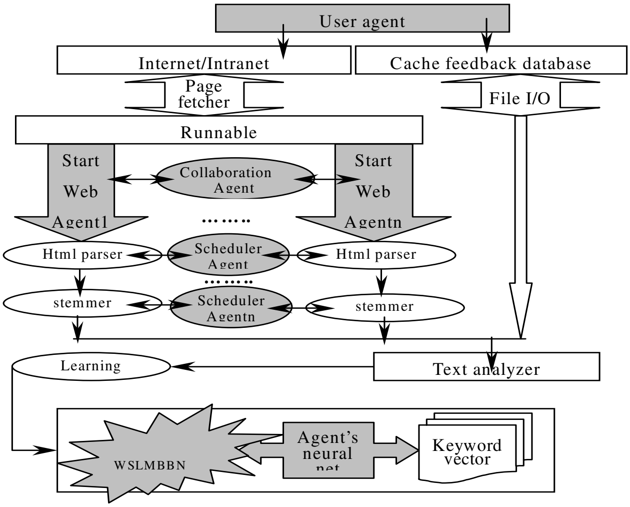


Fig. 3. Each agent running mode of CES

The adaptive representation of CES roughly consists of a list of keywords, initialized with the query terms, and of a feed-forward neural net. The keywords represent an agent's opinion of what terms best discriminate documents relevant to the user from the rest. The neural net is used to estimate links; it has an input for each keyword and a single output unit.

At each step, each agent analyzes the text of the document where it is currently situated to estimate the relevance of its information neighborhood, given by the outgoing hyperlinks in the current page [5]. In simple terms, an agent estimates each outgoing link by looking at the occurrence of query terms in the vicinity of the link. The agent then uses the link relevance estimates to choose the next document to visit. More formally, for link *P* and for each keyword *Q*, the neural net receive input *GET*:

$$GET_{Q,P} = \sum_{i:WEI(Q_i,P)\le\varepsilon} \frac{1}{WEI(Q_i,P)} \qquad (1)$$

Where Q_i is the *i*th occurrence of *k* in document *D* and $WEI(Q_i,P)$ gives more weight to keyword occurrences in the vicinity of *P*, by counting intervening links up to a maximum window size of $\pm\varepsilon$ links away. The neural network then sums activity across all of its input *GET*; each unit *j* computes a logistic activation function [6].

$$T_J = tag(BI_j + \sum_k \omega_{jk} GET_Q^P) \tag{2}$$

Where BI_j is its bias term, ω_{jk} are its incoming weights that is ensured by a new weight-ensured method: Weight Self-Learning Method Based on Bayes Net (WSLMBBN), and GET_Q^P its inputs from the lower layer. The next section will study how to ensure the neural net weight and present a new method: WSLMBBN.

4.2 Weight Self-learning Method Based on Bayes Net

(1) Establishing Bayes Network

Figure 4 is a typical Bayes Network graph according to neural net. After establishes the Bayes Network graph of the multi-object optimization problem, it should presents the "CPT" among each node (layer point of Neural Net) in the Bayes Network graph. The CPT is not probability; it is the initialized weight and Neural Net receiving input value. The CPT of Bayes Network of the CES is a two-unit structure (show in figure 5), which stores the initialized weight and input value of node to its evaluating index (parents).

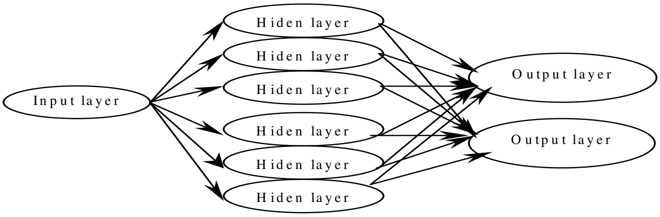


Fig. 4. Bayes network graph

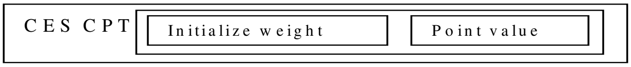


Fig. 5. CPT two structural unit of Bayes net graph

(2) Training and Learning of Bayes Network

The training of Bayes Network uses grads descending method in optimization, the goal is learning the value of CPT. It presumes that S is set with s training specimens $X_1, X_2, \cdots\cdots, X_S$, w is node's initialized weight of variable with parents in CPT. The w in each layer should satisfies $\sum w_i = 1$ in Bays network of optimization, e is an attribute in the optimizing course, which is corresponding to a point in the Bayes Network graph, $p(e \mid X_d)$ Expresses the prior probability of e point under the specimen of X_d, which is the node value of CPT in Bayes Network of optimization. w Can be taken as weight that resembles the weight of hidden unit in Neutral Network.

W Is the total set of weight. All weight are initialized as random probability value, the method of cupidity mountain climbing was adopted as grads descending strategy. It modifies these weights in any iteration until it converges a local best result.

Each possibilities of W 's value are all assumed at same probability, this method can learn the value of w. The goal is made the value of $P_W(S)=\prod_{d=1}^{s} P_W(X_d)$ biggest. The goal value computes according to the grads of $\ln P_W(S)$, which makes the problem easy. The algorithm can be deal with at the follow step at the given network structure and initialized value of w:

1) Computing grad: computing each attribute and evaluating index in the network structure:

$$\frac{\partial \ln P_W(S)}{\partial w}=\sum_{d=1}^{s}\frac{p(e \mid X_d)}{w} \tag{3}$$

2) Forward a small step at direction of grad, and uses the follow formula to update the weight,

$$w \leftarrow w+(l)\frac{\partial \ln P_w(S)}{\partial w} \tag{4}$$

l expresses the step's length, and $\frac{\partial \ln P_W(S)}{\partial w}$ is computed according to formula (3), the step's length is initialized as a small constant.

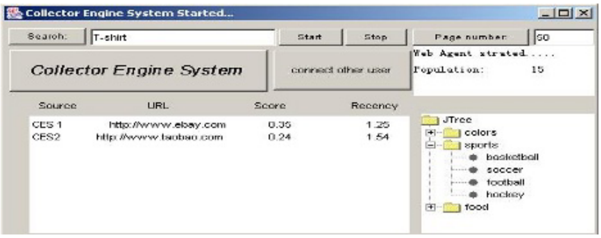
3) Regulating the weight again: because weight w is between 0.0 and 1.0, attribute's weight of each layer should satisfy $\sum w_i = 1$. After weight value is updated according to formula (4), other attribute's weight w at each layer should regulate again to guarantee this condition.

4) ε is a smallest value, when the grad computed by formula (3) satisfies $\frac{\partial \ln P_W(S)}{\partial w}<\varepsilon$, the training of Bayes Network is ended, if the grad don't satisfy $\frac{\partial \ln P_W(S)}{\partial w}<\varepsilon$, it should jump to the formula (5) for the next training.

5 CES Implementation

5.1 Developing CES

Due to the parallel nature of the CES, multithreading is expected to provide better utilization of resources compared to a single thread implementation such as described in the previous section. Since Java has built-in support for threads, we decided to implement a multithreaded version of CES as a Java applet. The multithreaded implementation allows one agent to use the network connection to retrieve documents, while another agent can use the CPU, and other can access information on the local disk. The CES is available on a public Web server.



Search: T-shirt

Page number: 50

```
Web Agent strated....
Population:       15
```

Source	URL	Score	Recency
CES 1	http://www.ebay.com	0.35	1.25
CES2	http://www.taobao.com	0.24	1.54

Fig.6. Screen shot of CES

5.2 Experimental Results

For the product database, over 200 items under eight categories were being used to construct the e-catalog [7]. These categories were: T-shirt, shoes, trousers, skirt, sweater, tablecloth, and napkins. We deliberately choose soft-good items instead of hard goods such as books or music CDs, so that it would allow more room for user requirement definition and product selection. For neural network training, all the e-catalog items were pre-trained in the sense that we had pre-defined the attribute descriptions for all these items to be fed into the neural network for production training. Thus totally, eight CES thread were constructed according to each different category of product. From the experimental point of view, the product-selecting test was conducted. In the product-selecting test, since there was no definite answer to whether a product would fit the taste of the customer or not, a sample group of 40 candidates was used to judge the effectiveness of the CES. Details are illustrated in the following sections. In the product-selecting test, each candidate would buy one product from each category according to his or her own requirement. For evaluation, they would browse around the net to choose a list of the best five choices (bfc), which fit his/her taste. In comparison with the top five recommended product items(i) given by the CES, the Fitness value (Fit) is calculated as follows:

$$Fit = \frac{\sum_{n=1}^{5} n \times i}{15} \; where \; = \begin{cases} 1 & \text{if i} \in \text{bfc} \\ 0 & \text{otherwise} \end{cases}$$

Table 1. Fitness value for product in CES

Product	Fit %
T-shirt	81
Shirt	78
Shoes	89
Trousers	88
Skirts	65
Sweater	81
Tablecloth	85
Napkins	86
Average score	81.6

In the calculation, scores of 5 to 1 were gave to correct matches of the candidate's first to fifth Best five choices with the CES's suggestion. For example, if out of the five best choices selected by the customer, products of rank no. 1,2,3 and 5 appear in the CES recommended list, the fitness value will be 73%, which is the sum of 1,2,3 and 5 divided by 15. Results under the eight different categories are shown in Table 1.

It is not difficult to predict that the performance of the CES in highly dependent on the variability of the merchandise. The higher the variety, the lower the score. As shown in Table1, skirts and shoes are typical example in which skirts score 65% and shoes scores 89%. Nevertheless, the average score is still over 81%. Note that these figure are only for illustration purpose, as human justification and product variety in actual scenarios do vary case by case.

6 Conclusions and Future Direction

In this paper, an innovative multi-agent based Web-mining application, CES, is proposed. Based on the integration of neural network and Bays Net based Web-mining technology (CES model) and intelligent data-mining technology (Web Agent) for automatic user authentication. It will hopefully provide a new era of Web-based data mining and knowledge discovery using intelligent agent-based system.

The future research plan is to perform data mining on user search activities such that user profiles can be learned automatically. Currently, users have to specify their areas of interest explicitly in order to access shared search sessions. We are currently planning to use data mining algorithm to enhance the CES by including more sophisticated content-based or collaborative-based information recommendation functionalities.

In conclusion, we believe that the experimental results are interesting and useful for related research and that the research issue identified should be further studied in other collaborative environments.

References

1. Spertus, E.: ParaSite: Mining Structure Information on the Web. In Proceedings of the Sixth Intl WWW Conference, (1997) 485-492.
2. Pitkow, J.: In Search of Reliable Usage Data on the WWW. In Proceedings of the Sixth int'l WWW Conference, (1997) 451-463.
3. F.MENCZER, R.Belew: Adaptive retrieval agents: internalizing local context and scaling up to the web, Machine Learning 39 (2-3) (2000) 203-242.
4. P. Maes: Agent that reduce work and information overload, Communications of the ACM37 (7)(1994)31-40.
5. K.Lang, NewsWeeder: learning to filter Netnews, Proceedings of the 12th International Conference on Machine Learning, San Francisco, CA, 1995.
6. Michael Chau, Danie Zeng, etal: Design and evaluation of a multi-agent collaborative web mining system, Decision Support System 35(2003)167-183.
7. Y.Y.Yao, H.J.Hamilton, and Xuewei Wang: PagePrompter: An Intelligent Web Agent Created Using Data Mining Techniques. Springer-Verlag Berlin Heidelberg 2002, 506-513.

A Novel Framework for Web Page Classification Using Two-Stage Neural Network

Yunfeng Li, Yukun Cao, Qingsheng Zhu, and Zhengyu Zhu

Department of Computer Science, Chongqing University, Chongqing, 400044, P.R.China
lyf129@126.com

Abstract. Web page classification is one of the essential techniques for Web mining. This paper presents a framework for Web page classification. It is hybrid architecture of neural network PCA (principle components analysis) and SOFM (self-organizing map). In order to perform the classification, a web page is firstly represented by a vector of features with different weights according to the term frequency and the importance of each sentence in the page. As the number of the features is big, PCA is used to select the relevant features. Finally the output of PCA is sent to SOFM for classification. To compare with the proposed framework, two conventional classifiers are used in our experiments: k-NN and Naïve Bayes. Our new method makes a significant improvement in classifications on both data sets compared with the two conventional methods.

1 Introduction

The amount of online text data has grown greatly in recent years because of the increase in popularity of the World Wide Web. As a result, there is a need to provide effective content-based retrieval, search, and filtering for these huge and unstructured online repositories. An effective text automatic categorization or classification system has several applications, such as the construction of recommendation systems or providing the ability to categorize very large libraries of text collections on the Web in an automated way.

The goal of web page categorization is to classify the information on Internet into a certain number of pre-defined categories. Text categorization is an active research area in information retrieval and machine learning. And Several text categorization have recently been proposed. For examples, there are Naïve Bayes [1], Rocchio [2] and Nearest Neighbor [3]

Furthermore, a feature selection using a hybrid case-based architecture has been proposed by Gentili et al [4] for text categorization where two multi-layer perceptrons are integrated into a case-based reasoner. Wermeter has used the document title as the vectors to be used for document categorization [5]. Ruiz and Srinivasan [6] and Calvo and Ceccatto [7] have used the X^2 measure to select the relevant features before classifying the text documents using the neural network.

A common approach for text categorization is to use unsupervised artificial neural networks. Neural networks are highly suited to textual input, being capable of identi-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 499–506, 2005.

fying structure of high dimensions within a body of natural language text. Neural networks work better than other method even when the data contains noise, has a poorly understood structure and changing characteristics. In the approach, the emphasis will be on using the hybrid neural network architecture, based on PCA and SOFM to improve the quality of clusters. The proposed method is different compared to the previous works based on the improvement of web page categorization accuracies using the PCA features selection approach and the SOFM with a combination of some conventional statistical methods. The experimental evaluation demonstrates that the proposed method provides the acceptable categorization accuracy.

The rest of this paper is organized as follows. Section 2 explains the proposed text categorization system in detail. Section 3 presents the discussion of empirical results in our experiments. The final section presents conclusions and future works.

2 The Hybrid Architecture

The proposed hybrid architecture consists of four modules as shown in Fig. 1: (1) the page-preprocessing module is used to extract textual features of a document, (2) the feature-weighting module is designed to rank the importance of features, (3) the feature-selecting module is utilized the PCA neural network to reduce the dimensionality of feature space, and (4) the page-classifying module is employed the SOFM neural network to perform the categorization.

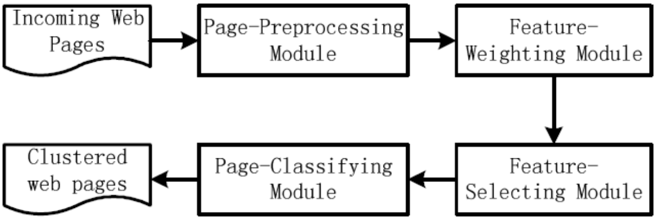


Fig. 1. The Architecture of proposed system

In the approach, each web page is represented by the term frequency-weighting scheme in the page-preprocessing module and the feature-weighting module. As the dimensionality of a feature vector in the collection set is big, the PCA has been used to reduce it into a small number of principal components in the feature-selecting module. Then the reduced feature vectors should be inputted into the page-classifying module that utilizes the SOFM to classification.

2.1 Page-Preprocessing Module

During the first stage of the proposed system, the full text of a document to be clustered must be parsed to produce a list of potential features that could serve as a basis for filtering process. The page-preprocessing module is divided into *stopping* and *stemming*. The *stemming* is a process of extracting each word from a document by reducing it to a possible root word. For example, the words ‘compares’, ‘compared’, and ‘comparing’ have similar meaning with the word ‘compare’. The *stopping* is a process of deleting the high frequent words with low content discriminating power in a document, such as ‘to’, ‘a’, ’and’, ’it’, etc. Deleting these words will save spaces for storing document contents and reduce time taken during the subsequent processes.

2.2 Feature-Weighting Module

After *stopping* and *stemming*, the rest terms in a page could be organized in a vector. The vector space model has been used as a conventional method for text representation. In the feature-weighting module, the vector obtained from the pre-processing module should be weighted using term frequency (TF) and inverted document frequency (IDF).

TF-IDF is the one that has been well studied in the information retrieval literature. This scheme is based on the assumption that terms that occur in fewer documents are better discriminators. Therefore, if two terms occur with the same frequency in a document, the term occurring less frequently in other documents will be assigned a higher value. But TF-IDF simply counts TF without considering where the term occurs. Each sentence in a document has different importance for identifying the content of the document. Thus, by assigning a different weight according to the importance of the sentence to each term, we can achieve better results. Generally, we believe that a title summarizes the important content of a document. Terms that occur in the title have higher weights. In the approach, we use WTF_i replace TF_i in TF-IDF, which is calculated as follow: (1) each time a word occurs in the title its WTF_i is increased by ten, (2) each time a word occurs in the heading its WTF_i is increased by six, (3) each time a word occurs in the boldface type its WTF_i is increased by three, and (4) each time a word occurs its WTF_i is increased by one.

Let DF_i be the frequency of occurrence of sentence s in a collection. The weight of word t_i, denoted by $W_{i,p}$, is expressed as follows:

$$W_{i,p} = \left({}^{WTF_i}\!\big/_{|P|} \right) \cdot \log \left({}^{n}\!\big/_{DF_i} \right) \tag{1}$$

where n is the number of documents in the collection, $j=1,..,n$, $|P| = \sum_j WTF_j$ is used to normalize term frequency to [0,1] in order to avoid favoring long documents over short documents.

2.3 Feature-Selecting Module

According to the length of a document, the high dimensionality of the term-weight vectors representing the document could result the difficulty to classify. To improve the accuracy of categorization, our approach uses the principal component analysis (PCA) technology, with a combination of statistical and natural language approaches, to reduce the original term-weight vectors with high dimensionality to a small number of relevant features.

The term-weight vector representing a document is considered as inputs to a neural network PCA system. PCA provides a means by which to achieve such a transformation where the feature space accounts for as much of the total variation as possible. Specifically, using the PCA procedure the original feature space is transformed into another feature space that has exactly the same dimension as the original. However, the transformation is designed in such a way that the original feature set may be represented by a reduced number of 'effective' features and yet retains most of the intrinsic information content of the data. Therefore, using PCA we achieve not only the increasing of feature variation but also decreasing of feature space dimensionality. A complete analysis of the PCA method used in this paper is given in [8,11].

Suppose that we have *A*, which is a matrix with document-terms weight as follows:

$$A = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1m} \\ x_{21} & x_{22} & \cdots & x_{2m} \\ \cdots & \cdots & \cdots & \cdots \\ x_{n1} & x_{n2} & \cdots & x_{nm} \end{pmatrix} \tag{2}$$

where x_{ij} is the terms weight that exist in the collection of documents. The definitions of *i, j, m, n* have been described in the previous paragraph. There are a few steps to be followed in order to calculate the principal components of data matrix *A*. The mean of *m* variables in data matrix *A* will be calculated as fellows:

$$\overline{x_j} = \frac{1}{n}\sum_{i=1}^{n}(x_{ij}) \tag{3}$$

After that the covariance of matrix $S = \{s_{ij}\}$ is calculated. The variance, s_{jj}^2, is given by

$$s_{jj}^2 = \frac{1}{n}\sum_{i=1}^{n}(x_{ij} - \overline{x_j})^2 \tag{4}$$

where *j=1, 2, ..., m*. The covariance, s_{kj}, is given by

$$s_{kj} = \frac{1}{n}\sum_{i=1}^{n}(x_{ik} - \overline{x_k})(x_{ij} - \overline{x_j}) \tag{5}$$

where *k=1, ..., m*. Then we determine the eigenvalue symmetric positive matrix. An eigenvalue λ and a nonzero vector e can be found such that, $Se = \lambda e$, where e is an eigenvector of S.

In order to find a nonzero vector e the characteristic equation $|S - \lambda I| = 0$ must be solved. If S is an $m \times m$ matrix of full rank, *m* eigenvalues $(\lambda_1, \lambda_2, ..., \lambda_m)$ can be found. By using

$$(S - \lambda I)e = 0 \tag{6}$$

all corresponding eigenvectors can be found. The eigenvalues and corresponding eigenvectors will be sorted so that $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_m$. The eigenvectors matrix is represented as $e = [u_1, u_2, \ldots, u_m]$. A diagonal nonzero eigenvalue matrix is represented as

$$\Lambda = \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 \\ 0 & \lambda_2 & \ldots & 0 \\ \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & \ldots & \lambda_m \end{pmatrix} \tag{7}$$

In order to get the principal components of matrix S, we will perform eigenvalue decomposition that is given by

$$\Lambda = E^T SE \tag{8}$$

Then we select the first $d \leq m$ eigenvectors where d is the desired value. The set principal components is represented as fellows:

$$Y_1 = e_1^T x, Y_2 = e_2^T x, \ldots, Y_d = e_d^T x \tag{9}$$

2.4 Page-Classifying Module

In the approach, the categorization module employs a Kohonen SOFM neural network, where the outputs of the PCA, namely the set of principal components, are the inputs of SOFM. The Self Organized Feature Map (SOFM) is one of the most widely applied ANN first introduced by Kohonen [9, 10]. It has successfully been used in a variety of applications including, data visualization, data clustering, pattern recognition and data mining. The SOFM's objective in document clustering is to group the documents, which appear similar, close to one another and place the very different ones distant from one another. The SOFM neural network combines its input layer with a competitive layer of neurons, and is trained by unsupervised learning. Typically, the competitive layer is organized as a two-dimensional square grid, and each neuron represents a class. Fig. 2 gives the Kohonen SOFM topology as the last module of the entire system. Although the gird is two-dimensional, each neuron is labeled by an index k which takes values from the set {1,2,…,m}, where *m* is the number of categories in the system.

After training, each neuron on the output layer represents a class of documents. Documents of large similarity are represented by the same neuron on the grid. Each neuron is labeled by the identity of the documents that were classified on it. More, specifically, due to the high discrimination ability of the input feature vector, the distributions of occurrences (votes) in the output neurons have small standard deviations. That is, the majority of occurrences in each neuron in the competition layer corresponds clearly to one categories. Thus, after the training stage it is easy to label each neuron with the correct class. The Kohonen SOFM organizes the neurons of the competitive layer in such a way that similarities among documents are mapped into closeness relationships of the competitive layer grid. Also, the SOFM provides advantages over classical text classification techniques because it utilizes the parallel architecture of a neural network and provides a graphical organization of document relationships.

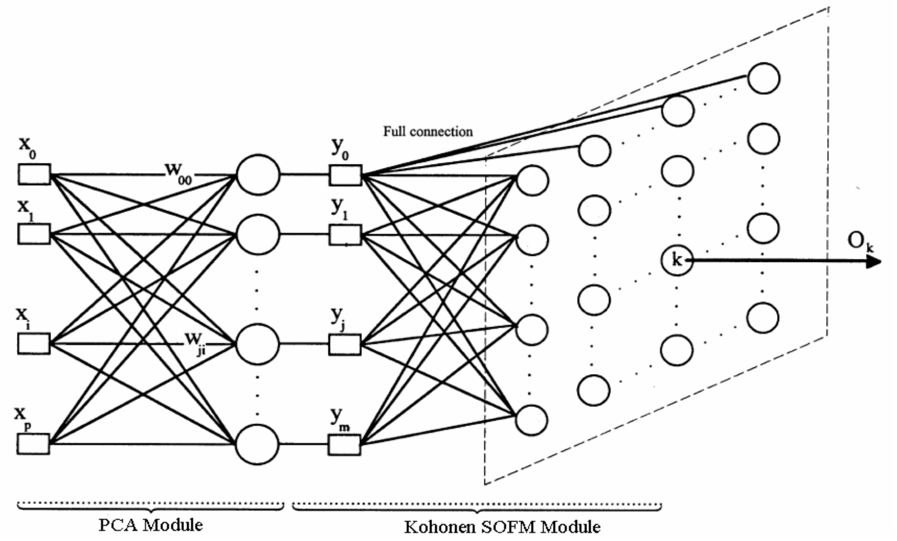


Fig. 2. The relation of the 3th module using PCA and the 4th module using SOFM

Suppose that the input $y = [y_1, y_2,..., y_m]^T$, the weight vector of the neuron *j* in SOFM is $w_j = [w_{j1}, w_{j2},..., w_{jm}]^T$. There are two basic steps in SOFM, which are the search for the Best Matching Unit (BMU) *i* of weight vector w_i and *y*, and updating the BMU *i* with it's neighbours. The BMU *i* is found by computing the Euclidean distance between the input data vector *y* (document) and the reference vector w_i (weight) as show as follows:

$$i(y) = \arg\min_j \{ \| y - w_j \| \} , \; j = 1, 2, ..., \; n \tag{10}$$

where *n* is the number of neurons in the SOFM's feature map. Once we have found the BMU we update the BMU and it's neighbouring nodes using:

$$w_i(t+1) = w_i(n) + \Lambda_{i,j}(t)[x(t) - w_i(t)] \; , \; t = 1,2,3,... \tag{11}$$

where $\Lambda_{i,j}(t)$ is neighbourhood function, *t* is discrete time constant. The neighbourhood function $\Lambda_{i,j}$ used in equation (11), is a time decreasing function which determines to which extent the neighbours of the BMU will be updated. The extent of the neighbourhood is the radius and learning rate contribution, which should both decrease monotonically with time to allow convergence. The radius is simply the maximum distance at which the nodes from the BMU are affected. A typical smooth $\Lambda_{i,j}$ is given bellow in equation (12).

$$\Lambda_{i,j}(t) = \alpha(t) \cdot \exp\left(-\frac{\| r_i - r_j \|^2}{2\sigma(t)} \right) \tag{12}$$

where $\alpha(t)$ is the learning rate function, $\sigma(t)$ is the kernel width function, $\| r_i - r_j \|^2$ is the distance of BMU *i* unit to current unit *j*. There are various functions used as the learning rate $\alpha(t)$ and the kernel width functions $\sigma(t)$. For further details about the SOFM please refer to [9] and [10].

4 Experiments

To test the proposed system, we collected a data set of sports news obtained from the Yahoo.com and Google.com, including 5,732 web pages. The types of news in the data set are tennis (718 documents), swimming (116 documents), baseball (953 documents), football (1257 documents), golf (521 documents), badminton (126 documents), boxing (374 documents), rugby (578 documents), skating (279 documents) and skiing (105 documents). Among the data set, 4500 documents (about 80%) selected randomly from different classes were used for training data, and the remaining 1232 documents (about 20%) for test data. We have used the Naïve Bayes and k-Nearest Neighbor classifiers mentioned before. Also we include the proposed approach as a comparison to the methods that are used in order to examine the applicability of the classification system. The training and test sets for all classifiers were the same. As performance measures, we employed the standard information retrieval measures of recall (r), precision (p), and F1 (F1=2rp/(r+p)) to evaluate our system comparing with Naïve Bayes and k-Nearest Neighbor classifiers. The average for precision, recall and F1 measures using the Naïve Bayes classifier are 79.09%, 84.54%, 81.67%, respectively. And the average for precision, recall and F1 measures using the K-NN classifier are 82.28%, 86.11%, 84.08%, respectively. In comparison with the proposed classifier, the precision, recall, and F1 measures are 86.24%, 88.12%, and 86.87, respectively. This indicates that if the feature vectors a selected carefully, the improvement of web sports news classification using the combination of the PCA and SOFM in the approach will increase the classification accuracy.

5 Conclusions

In this paper, we have presented a new approach for web pages categorization using a hybrid neural network. The comparison of categorization accuracy between the Naïve

Bayes classifier, K-NN classifier, and the proposed system has been presented in this paper. The experimental evaluation with different classification algorithms demonstrates that this system achieved a better performance did in all these classifiers. Although the classification accuracy using the proposed system is high in comparison with the other approaches, the time taken for training is relatively long compared with the other methods. The system could be utilized in digital library, resource discovery, local data management, and so on.

References

1. McCallum, A., & Nigam, K.: A comparison of event models for Naïve Bayes text classification. In AAAI'98 workshop on learning for text categorization (1998) 41-48
2. Lewis, D.D., Schapire, R.E., Callan, J.P., & Papka, R.: Training algorithms for linear text classifiers. In Proceedings of the 19th international conference on research and development in information retrieval (1996) 289-297
3. Yang, Y., Slattery, S., & Ghani, R.: A study of approaches to hypertext categorization. Journal of Information Systems, archive Volume 18, Issue 2-3 March-May (2002)
4. G.L. Gentili, M. Marinilli, A. Micarelli, F. Sciarrone: Text categorization in an intelligent agent for filtering information on the Web. International Journal of Pattern Recognition and Aritificial Intelligence 15 (3) (2002) 527-549
5. S. Wermeter: Neural network agents for learning semantic text classification. Information Retrieval 3 (2) (2000) 87-103
6. E.M. Ruiz, P. Srinivasan: Hierarchical text categorization using neural networks. Information Retrieval 5 (1) (2002) 87-118
7. R.A. Calvo, H.A. Ceccatto: Intelligent document classification. Intelligent Data Analysis 4 (5) (2000) 411-420
8. R.A. Calvo, H.A. Ceccatto: Intelligent document classification. Intelligent Data Analysis 4 (5) (2000) 411-420
9. Teuvo Kohonen: Self-Organizing Maps, Second Extended Edition. Springer Series in Information Sciences, Vol. 30, Berlin, Heidelberg, New York, (1997)
10. S. Haykin: Neural Networks: A Comprehensive Foundation (2nd Edition). Prentice Hall, (1999)
11. R.A. Calvo, M. Partridge, M. Jabri: A comparative study of principal components analysis techniques. In Proceedings 9th Australian Conference on Neural Networks, Brisbane, QLD (1998) 276-281
12. R.A. Johnson, W.D. Wichern: Applied Multivariate Statistical Analysis (5th edition). Prentice-Hall, USA, (2002)
13. O. Nouali, P. Blache: A semantic vector space and features-based approach for automatic information filtering. Expert Systems with Applications, 26 (2004) 171-179
14. Ali Selamat: Web page feature selection and classification using neural networks. Information Sciences 158 (2004) 69-88
15. Youngjoong Ko, Jinwoo Park, Jungyun Seo: Inproving text categorization using the importance of sentences. Information Processing and Management 40 (2004) 65-79

Fuzzy Evaluation of Hotel Websites

Rob Law

School of Hotel & Tourism Management,
The Hong Kong Polytechnic University,
Hung Hom, Kowloon, Hong Kong
hmroblaw@polyu.edu.hk

Abstract. Prior studies on hotel website performance have primarily concentrated on frequency counting, content analysis or user behavioral approaches. These studies, however, failed to offer any insight that can accurately evaluate hotel website quality based on users' assessment of attribute weights and performance ratings. This research proposes a fuzzy multicriteria analysis model which systematically integrates hotel guests' preferences and fuzzy assessments of website attributes. The fuzzy evaluation of linguistic values by respondents offers a comprehensive approach for handling incomplete and imprecise user preferences to capture the realistic evaluation process. The research output will be a set of overall performance indices representing the cumulative effect of website features, which will offer a benchmark enabling hotels to evaluate their websites' relative performance and ranking based on all relevant weighted attributes.

1 Introduction

In spite of the existence of numerous hotel websites, there is no standard measurement or ranking mechanism for assessing the performance of these websites. In academia, published research articles on hotel websites and e-business quantitative analyses rarely incorporate hotel guests' preferences and views. Since the comprehension of useful information from hotel data is a difficult and slow task [11], the provision of a formal systematic assessment method which computes the overall performance index of a hotel website certainly helps hotel guests and practitioners in their decision-making process.

Human subjective assessments are characterized by qualitative linguistic expressions that are imprecise in nature. These linguistic assessments, in turn, add to the overall uncertainty in the decision-making process. The fuzzy set theory's ability to model the qualitative human expression in linguistic terms (i.e., variables whose values are not numbers but words or sentences in a natural or artificial language) makes a fuzzy set assert factuality and strongly represent reality.

In this paper, the primary objective is to presents a method that attempts to investigate the main research question:

Is it feasible to construct a performance index of hotel websites based on users' subjective assessment?

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 507 – 514, 2005.

Related sub-issues arisen from this research question are:

i) *What quality attributes, in linguistic terms, can determine the performance of a hotel website?*
ii) *What are the weights that these determining attributes contribute to the overall performance of a hotel website?*
iii) *In terms of applications, what are the relative performances of hotel websites?*

Having introduced the research background, the remaining sections of this paper are organized as follows. First, there is a literature review section which summarizes the related work. The section after that presents the proposed methodology and research plan. The last section states the contribution of the research, and its long-term impact to the industry.

2 Related Work

In general, existing literature on website evaluations falls into two major categories: i) theoretical studies, which logically develop a concept/approach of the evaluation, and ii) empirical studies, which validate or verify an evaluation technique with experimental findings. The empirical studies are further divided into two subcategories, namely i) without user involvement and ii) with user involvement.

2. 1 Theoretical Studies

Researchers have long emphasized the importance of assessing a website's effectiveness. For instance, Lu and Yeung [13] presented a framework for e-commerce applications. In the framework, the overall usability of a website depends on the performance of functionality and usability of the site, which in turn, is a sum of many other attributes. Evans and King [5] stated that a website's performance can be determined by hit rate, log analysis, online surveys, e-mail correspondence, and expert opinions. Further, Evans and King [5] proposed an elementary tool that used objective values to assess the marketing effect of B2B websites.

2.2 Empirical Studies

Without User Involvement

Mainly using content analysis approaches, prior studies in this area can be classified into two groups. The first group concerns the evaluation of website attributes based on frequency counting and the subjective evaluation of features. Examples of frequency counting in this group included studies of hotels [14], tourism organizations [16], and general businesses [6].

The second group describes computerized approaches, which automatically evaluate the content of a website. An example of a computer system which automatically assesses hotel websites is the webLyzard developed by Wöber et al. [17].

With User Involvement

i*) Qualitative Analyses*

Numerous articles have been published that have analyzed primary data collected from users or practitioners on the importance of website performance. These studies were predominantly based on respondents' Likert-scale ratings on the relative importance of a set of selected website attributes. For example, Jung and Butler [7] investigated the views of tourism marketing managers about the most important factors and the measure of success of their websites. In a recent Omnibus Survey, Law and Wong [12] found that online travel purchasers viewed a secure payment method as the most important factor for a successful travel website. In these studies, the importance rating of factors was represented by the average scores of Likert-type scale interval values.

Although these studies have concluded many important factors that contribute to the overall performance of a website, the vital attributes that can be generalized consisted of price level, security, and user-friendliness.

ii) Quantitative Modeling

Chung and Law [3] developed an information quality model to measure the functionality performance of hotel websites. Using the equal-interval rated attributes from Hong Kong hotel managers, the model computed a mean score M_a for each attribute of the five selected dimensions, and this mean score was then transformed into a weighted score W_a, for a total of n attributes by:

$$W_a = \frac{1 + n - M_a}{\sum_{i=1}^{n} Mi} \tag{1}$$

Each attribute was then assigned a score in the range of 1 to 5, which represented the rating of this attribute. A performance score was then computed to reflect the website attribute's performance.

Au Yeung and Law [1] extended Chung and Law's [3] approach to investigate the usability performance of Hong Kong hotel websites. In addition to involving hotel managers, the researchers incorporated the views of hotel customers and IT professionals in the attribute rating process. In particular, respondents were asked to weigh the relative importance of responses from each group so that an average weighted score W_i for attribute i was calculated for its usability hazard within a dimension (for m attributes) by:

$$W_i = \frac{\sum_{j=1}^{x} C_j}{x}\frac{\sum_{k=1}^{m} P_k}{m} + \frac{\sum_{j=1}^{y} H_j}{y}\frac{\sum_{k=1}^{m} Q_k}{m} + \frac{\sum_{j=1}^{z} T_j}{z}\frac{\sum_{k=1}^{m} R_k}{m} \tag{2}$$

where C,H,T are the rating values given by customers, hoteliers, and IT professionals, respectively, for a total of x customers, y hoteliers, and z IT professionals. P, Q, R are the relative importance scores provided by customers, hoteliers, and IT professionals; $P_i + Q_i + R_i = 100\%$. The usability performance of all selected attributes for each hotel website was independently evaluated by two users using a five-point subjective

Likert-type scale. Aggregating the weighting and rating scores generated an overall performance score.

To summarize, the existing published articles, albeit offering a comprehensive overview of the success factors of business websites, have failed to provide a comprehensive systematic approach that quantitatively measures a website's overall performance based on users' subjective evaluation in linguistic terms. The fuzzy set's ability to capture the variabilities of linguistic variables appears to be a superb technique to model the reality for evaluating hotel website performance based on users' qualitative expressions. Research outputs are therefore expected to serve as a benchmark in knowledge development in the context of hotel website performance.

3 The Proposed Methodology

3. 1 The Proposed Fuzzy Set Model to Solve the Research Problem

The problem can be presented so as to subjectively evaluate the performance of a set of r hotel websites (alternatives) H_i ($i = 1,2,3,\ldots,r$) based on a set of s website attributes (criteria) A_j ($j = 1,2,3,\ldots,s$). Assessments are therefore made based on a multicriteria analysis to determine i) the weighting vector $W = (w_1,w_2,w_3,\ldots,w_s)$ of $A_{j,}$ $\forall j$ and ii) the decision matrix $P = (p_{ij};\ i = 1,2,3,..,r;\ j=1,2,3,\ldots,s)$ for hotels H_i w.r.t. A_j. Each w_i in W can be input by customers [8,18] or obtained by using pairwise comparison of analytic hierarchy process [4,10]. The main objective of the research is then to evaluate and rank the hotel websites by assigning a performance score to H_i w.r.t. $A_j,\forall i,j$.

An overall performance matrix for a specific hotel website M is then obtained by:

$$M = W \bullet P \tag{3}$$

Triangular fuzzy numbers suggested by Klir et al. [9] and Zimmermann [19] for a five-point scale linguistic variable with fuzzy membership functions $((\beta_{1l}, \beta_{1m}, \beta_{1h}), (\beta_{2l}, \beta_{2m}, \beta_{3h}), (\beta_{3l}, \beta_{3m}, \beta_{3h}), (\beta_{4l}, \beta_{4m}, \beta_{4h}), (\beta_{5l}, \beta_{5m}, \beta_{5h}))$ will be used for interval calculations. To model customers' confidence level in the fuzzy evaluation w.r.t. w_i and p_{ij}, the theory of $\propto$-cut, $0 \leq \propto \leq 1$, is integrated to determine the crisp value interval of a corresponding fuzzy number. For a specific value of $\propto$, m^{α}_{ijl} and m^{α}_{iju} represent the lower and upper bounds of the crisp intervals respectively. Hence, M in (3) will become:

$$M_{\alpha} = \begin{bmatrix} \left[m^{\alpha}_{11l}, m^{\alpha}_{11u}\right] & \left[m^{\alpha}_{12l}, m^{\alpha}_{12u}\right] & \ldots\ldots & \left[m^{\alpha}_{1sl}, m^{\alpha}_{1su}\right] \\ \left[m^{\alpha}_{21l}, m^{\alpha}_{21u}\right] & \left[m^{\alpha}_{22l}, m^{\alpha}_{22u}\right] & \ldots\ldots & \left[m^{\alpha}_{2sl}, m^{\alpha}_{2su}\right] \\ \ldots. & \ldots\ldots & \ldots\ldots & \ldots. \\ \left[m^{\alpha}_{r1l}, m^{\alpha}_{r1u}\right] & \left[m^{\alpha}_{r2l}, m^{\alpha}_{r2u}\right] & \ldots\ldots & \left[m^{\alpha}_{rsl}, m^{\alpha}_{rsu}\right] \end{bmatrix} \tag{4}$$

An index ϕ, $0 \leq \phi \leq 1$, for customers' preference between m^{α}_{ijl} and m^{α}_{iju} in (4) is added to $M_{\propto}$. $M_{\propto}$ in (4) then becomes:

$$M_{\alpha}^{\phi'} = \begin{bmatrix} m_{11\alpha}^{\phi'} & m_{12\alpha}^{\phi'} & \cdots\cdots & m_{1r\alpha}^{\phi'} \\ m_{21\alpha}^{\phi'} & m_{22\alpha}^{\phi'} & \cdots\cdots & m_{2r\alpha}^{\phi'} \\ \cdots\cdots & \cdots\cdots & \cdots\cdots & \cdots\cdots \\ m_{s1\alpha}^{\phi'} & m_{s2\alpha}^{\phi'} & \cdots\cdots & m_{sr\alpha}^{\phi'} \end{bmatrix} \tag{5}$$

where each $m_{ij\alpha}^{\phi'}$ in (5) equals ϕm_{iju}^{α} + (1- ϕ) m_{ijl}^{α}, for $i \in [1,r]$, $j \in [1,s]$.

To solve the multiattribute performance evaluation problem presented in (5), this paper will adopt the technique proposed by Chu and Tsao [2] to rate fuzzy number N with membership function f_N, and f_N is a continuous mapping in $[0,w]$ for $0 \le w \le 1$, f_N is continuously increasing in $[a,b]$, $f_N(x) = w$ for $x \in [b,c]$, and f_N is continuously decreasing in $[c,d]$. The area between centroid (x',y') and original points (0,0) is modified as

$$A(N) = x'(N)*y'(N) \tag{6}$$

$$x'(N) = \frac{\int_a^b (x f_N^L)\,dx + \int_b^c x\,dx + \int_c^d (x f_N^R)\,dx}{\int_a^b (f_N^L)\,dx + \int_b^c dx + \int_c^d (f_N^R)\,dx} \tag{7}$$

$$y'(N) = \frac{\int_0^w (y g_N^L)\,dy + \int_0^w (y g_N^R)\,dy}{\int_0^w (g_N^L)\,dy + \int_0^w (g_N^R)\,dy} \tag{8}$$

where f_N^L and f_N^R are the left and right membership functions of N; g_N^L and g_N^R are their inversed functions.

The next step is to compute the most and least representative solutions. The positive and negative ideal solutions [15], denoted by $H_{\alpha}^{\phi+}$ and $H_{\alpha}^{\phi-}$, are computed from (5) and (6) to represent the best (maximum) and worst (minimum) values for all websites (alternatives) w.r.t. each $A_j, \forall j$.

Let $m_{j\alpha}^{\phi+}$ and $m_{j\alpha}^{\phi-}$ be the performance vectors of the best and worst evaluated hotels in relation to s website attributes.

$$m_{j\alpha}^{\phi+} = \{(\max(m_{ij\alpha}^{\phi}) \mid i \in [1,r]\};\ m_{j\alpha}^{\phi-} = \{(\min(m_{ij\alpha}^{\phi}) \mid i \in [1,r]\}, j = \in [1,s] \tag{9}$$

$H_{\alpha}^{\phi+}$ and $H_{\alpha}^{\phi-}$ can be obtained by:

$$H_{\alpha}^{\phi+} = (m_{i\alpha}^{\phi+}) \text{ and } H_{\alpha}^{\phi-} = (m_{i\alpha}^{\phi-}),\ i \in [1,s] \tag{10}$$

Subsequently, the closeness of a particular website to the positive and negative ideal solutions can be computed as:

$$C_{i\alpha}^{\phi+} = \frac{A_{i\alpha}^{\phi} A_{\alpha}^{\phi+}}{\max[\left(A_{i\alpha}^{\phi}\right)^2, \left(A_{\alpha}^{\phi+}\right)^2]},\ i \in [1,r] \tag{11}$$

$$C_{i\alpha}^{\phi-} = \frac{A_{i\alpha}^{\phi} A_{\alpha}^{\phi-}}{\max[\left(A_{i\alpha}^{\phi}\right)^2, \left(A_{\alpha}^{\phi-}\right)^2]},\ i \in [1,r] \tag{12}$$

Lastly, the relative closeness of a website to the ideal solution of each hotel website, represented by a score of performance index $S_{i\alpha}^{\phi}, \forall i$, can be determined by:

$$S_{i\alpha}^{\phi} = \frac{C_{i\alpha}^{\phi+}}{C_{i\alpha}^{\phi+} + C_{i\alpha}^{\phi-}},\ i \in [1,r] \tag{13}$$

The value of $S_{i\alpha}^{\phi}$ in (13) indicates that a favored website should have a higher degree of closeness to the positive ideal solution and a smaller degree of closeness to the negative ideal solution. Ranking the values of $S_{i\alpha}^{\phi}, \forall i$, in descending order yields the relative performance indices for the selected hotel websites.

3.2 Proposed Research Plan

i) A list of key attributes (factors) which can determine the usefulness, including functionality and usability of hotel websites will be identified in this phase.
ii) This phase will cover the distribution and collection of questionnaires from qualified hotel guests.
iii) This phase will implement the formal modeling approach presented earlier.
iv) This phase will compare and contrast the findings from the previous phase with the hotel tariff category as classified by the national tourism board.
v) This phase will identify and analyze the limitations of the research.

4 Conclusions

4.1 Relevance and Significance to the Industry

In general, hotels realize the importance of integrating e-commerce into their business strategies. Hence, these hotels have established websites to facilitate the marketing and consumers' purchasing processes. However, the existing literature has a very limited number of published articles that evaluate the performance of hotel websites from a customers' perspective. The lack of prior studies is particularly true in the context of consumers' subjective assessment of qualitative linguistic variables. In other words, neither hotel practitioners nor guests have a referencing standard for ascertaining the relative performance or rating of a specific website from users' perspective. The fuzzy set model presented in this research will contribute to filling this void. The model should have a direct applicability to hotel websites in other regions.

4.2 The Long-Term Impacts of the Research Are Multifold

i) Research findings will benefit hotel guests who do not have sufficient experience, familiarity, or time to better understand, compare and contrast the various hotels where the industry does not have a formal star-rating system.

ii) Based on the performance indices, hoteliers will know the relative positions of their website features, and can therefore enhance the effectiveness of their communication to market their services/products for the appropriate level of quality, price, and target market segment.

iii) Most importantly, the approach will offer further insights into the theoretical and empirical evaluations of hotel websites, which are responsible for one of the top three online purchases [11].

Acknowledgement

This research was supported, in part, by a Hong Kong Polytechnic University research grant (under contract number: G-T863).

References

1. Au Yeung, T., Law, R.: Usability Evaluation of Hong Kong Hotel Websites. In: Frew, A., O'Connor & Hitz, M. (eds): Information and Communication Technologies in Tourism 2003. Springer-Verlag, Wien New York (2003) 261-269
2. Chu, T.C., Tsao, C.T.: Ranking Fuzzy Numbers with an Area between the Centroid Point and Original Point. Computers and Mathematics with Applications. 43 (2002) 111-117
3. Chung, T., Law, R.: Success Factors for Hong Kong Hotel Websites. In: Proceedings of the Fifth Biennial Conference on Tourism in Asia, Hong Kong (2002) 96-104
4. Duke, J.M., Aull-Hyde, R.: Identifying public preferences for land preservation using the analytic hierarchy process. Ecological Economics 42 (2002) 131-145
5. Evans, J.R., King, V.E.: Business-to-Business Marketing and the World Wide Web: Planning, Managing, and Assessing Web Sites. Industrial Marketing Management 28 (1999) 343-358
6. Huizingh, E.K.R.E.: The content and design of web sites: an empirical study. Information & Management 37 (2000) 123-134
7. Jung, T., Butler, R.: Perceptions of Marketing Managers of the Effectiveness of the Internet in Tourism and Hospitality. Information Technology & Tourism 3(3/4) (2000) 167-176

8. Kao, C., Liu, S.T.: Operations research applications in Taiwan: A linguistic approach. European Journal of Operational Research 103 (1997) 628-634
9. Klir, G.J., St. Clair, U.H., Yuan, B.: Fuzzy Set Theory: Foundations and Applications. Prentice Hall, New Jersey (1997)
10. Korpela, J., Kyläheiko, K., Lehmusvaara, A., Tuominen, M.: An analytic approach to production capacity allocation and supply chain design. International Journal of Production Economics 78 (2002) 187-195
11. Law, R.: E-Commerce Applications in the Tourism Industry. In: Conference Proceedings: International Conference on Competitive Success and Challenges in Tourism. (2004) 64-78
12. Law, R., Wong, J.: Successful Factors for a Travel Web Site: Perceptions of Online Purchasers in Hong Kong. Journal of Hospitality & Tourism Research 27(1) (2003) 118-124
13. Lu, M., Yeung, W.L.: A framework for effective commercial Web application development. Internet Research: Electronic Networking Applications and Policy 8(2) (1998) 166-173
14. O'Connor, P., Horan, P.: An Analysis of Web Reservation Facilities in the Top 50 International Hotel Chains. International Journal of Hospitality Information Technology 1(1) (1999) 77-85
15. Tsaur, S.H., Chang, T.Y., Yen, C.H.: The evaluation of airline service quality by fuzzy MCDM. Tourism Management 23 (2002) 107-15
16. Weeks, P., Crouch, I.: Sites for Sore Eyes: An Analysis of Australian Tourism and Hospitality Web Sites. Information Technology & Tourism 2(3/4) (1999) 153-172
17. Wöber, K.W., Scharl, A., Natter, M., Taudes, A.: Success Factors of European Hotel Web Sites. In: Wöber, K.W., Frew, A.J, and Hitz, M. (eds): Information and Communication Technologies in Tourism 2002. Springer-Verlag Wien New York (2002) 397-406
18. Yeh, C.H., Kuo, Y.L.: Evaluating passenger services of Asia-Pacific international airports. Transportation Research Part E. 39(1) (2003) 35-48
19. Zimmermann, H.J.: Fuzzy Set Theory and its Applications. Kluwer Academic Publishers, Massachusetts (2001)

Querying Web Images by Topic and Example Specification Methods

Ching-Cheng Lee and Rashmi Prabhakara

Mathematics and Computer Science Dept, California State Univ, East Bay,
25800 Carlos Bee Blvd, Hayward, CA 94542
cclee@mcs.csuhayward.edu

Abstract. Ever since the advent of Internet, there has been an immense growth in the amount of image data that is available on the World Wide Web. With such a magnitude of image availability, an efficient and effective image retrieval system is required to make use of this information. This research presents an image matching and indexing technique that improvises on existing integrated image retrieval methods. The proposed system integrates query by topic and query by example specification methods. The topic-based image retrieval uses the structured format of HTML documents to retrieve relevant pages and potential match images. The query by example specification performs content-based image match for the retrieval of smaller and relatively closer results of the example image. The main goal is to develop a functional image search and indexing system without using a database and to demonstrate that better retrieval results can be achieved with this proposed hybrid search technique.

1 Introduction

An exponential increase in the amount of image data on the Internet has brought the need for efficient and effective image search systems. This demand has made image retrieval a very active area of research in recent years [1][4][5][6][12]. Search technology may be the foundation of the Internet, but if one is looking for rich media content, today's text or keyword-based searches are inadequate. There are two major approaches for image matching and retrieval, namely Query-by-Text (QbT) and Query-by-Content (QbC). Query-by-Text is the traditional approach for retrieving images. In this approach, queries are texts and results set are images. Images in QbT retrieval are often annotated by words. When images are sought using these annotations, such retrieval is known as annotation-based image retrieval (ABIR). Current day search engines like Google and Alta Vista use Query-By-Text approach for image search and retrieval. Using the ABIR approach to query images is considered practical in many general settings. However, text–based description tends to be incomplete, imprecise and inconsistent in specifying visual information [8][11][13][14][15]. For example, a search for a certain shade of blue sky is very difficult with textual descriptions. Using a text-only search engine, a query to look for photographs of zebra or a tiger on the Web, would result in thousands of search

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 515 – 526, 2005.

results. And there is no easy way to ensure that the search results are relevant and worth looking at. Also, photographs, graphics, logos, audio or video, text and image descriptions provided manually to the system rarely convey accurate query criteria [17].Query-by-Content is the most popular approach for retrieving images. In this approach, queries are images and result sets are also images. In this approach, retrieval is carried out based on the image content. Such retrieval is known as content-based image retrieval (CBIR). Query-by-Content approach uses the visual characteristics of images (the specific characteristics of the pixels that form the digital image) rather than words that have been related to the image to compare and retrieve the images. This technique involves analysis of specific aspects of the visual content of an image or a picture (texture, shape, color). The user inputs a query in the form of an example picture. The data which has been derived from prior analysis of that query or the example image's visual content is matched against the data relating to the other images or pictures and the search results returned, fall within specified parameters for patterns of data. This is aimed to aid users in retrieving relevant images based on their abstracted contents. CBIR is suitable for applications such as medical diagnosis based on the comparison of X-ray pictures with past cases, finding the faces of criminals from video shots of a crowd. However, this approach is computationally intensive and applications that involve more semantic relationships cannot be dealt with using this technique, even if extensive image processing procedures are applied. For example, in the gathering of the photos regarding the 'Presidential Elections 2004', it is not clear what kind of images should be used for the querying. This is simply because visual features cannot fully represent concepts. A textual description is needed to help such queries.

1.1 Related Work

Several new techniques have been proposed in the recent years for retrieving images and among them, a key technique is the content-based (CBIR) approach that retrieves images based on various image features. However many of the existing content-based approaches are computationally intensive and are not suited for image searches on the Internet (WWW). This section briefly summarizes the various popular image search systems. QBIC [16], IBM's Query by Image Content system lets users make queries to large image databases based on visual image content -- properties such as color percentages, color layout, and textures occurring in the images [1][3][6][7][8][9][10]. Such queries use the visual properties of images that match colors, textures and their positions without describing them in words. Using image texture feature does not provide satisfactory results when used as the basis for querying images [3][6][15]. For example, a query that focuses only on texture would not be able to distinguish between a tiger and a zebra. Classical shape recognition techniques tend to require that the object be clearly segmented from the rest of the image, a process that would require a lot of computations [2][8][14]. The main thrust of research in shape recognition has been for fixed, geometric objects in controlled images such as machine parts on a white background. Classical techniques can be easily applied in such cases, but they do not prove very useful in more general settings. Segmentation is imperfect in general cases because the shape, size, and color of objects contained in such photos can vary greatly [3]. Content-based queries are often combined with text and keyword predicates to get powerful retrieval methods for image and multimedia databases. The main goal of IMEDIA [22], a project research team is to develop content-based image indexing techniques and interactive search and retrieval methods for browsing large multimedia databases by content. IKONA is prototype software that illustrates the research that is lead at IMEDIA. By default, IKONA performs "retrieval by visual similarity" in response to a query, which means that it searches all images in the database and returns a list of the most visually similar images to the query image. IKONA allows also region-based queries and has hybrid text-image retrieval mode. VIPER [23] [**V**isual **I**nformation **P**rocessing for **E**nhanced **R**etrieval], yet another group is focused on developing system that is concerned with algorithms, data structures and image representations for content-based retrieval of images and video sequences from large databases. There is a particular emphasis on the interaction of the user with the database, and the use of user-provided information (i.e. relevance feedback) for the design and incremental improvement of the database system, on a variety of time-scales. Since the World Wide Web deals with both text and images, integrating both text and content-based approaches has been promising area for image retrieval. But, the existing image search systems that use the kind of integrated approach for querying images on the Internet follow primitive or traditional approach of accepting the image specifications from the user manually and hence are not very effective. WebSeer [17] is a system that retrieved images from the Web using information from two sources: the text that relates to the image and the image itself. It uses the image content in addition to associated text to index images, presenting the user with a selection that potentially fits ones need. Since the details pertaining to the image features like size and color are accepted from the user manually, the results are not very accurate. WebSeek [25,26] – a content-based image and video catalog search tool for the World Wide Web, allows users to first narrow down their searches by selecting a category from a semi-automatically-defined hierarchy. Images are pre-assigned to categories based on textual cues such as file names and surrounding text. Next, the search is refined by content-based methods (based on color histograms) to sort the retrieved images by similarity to a selected image. AMORE [24] is similar; the content-based indexing is based on objects (regions) in the image and their shapes and colors. Instead of using an actual image as a target, users can query the database using a synthetic image, such as an image of a single ellipse. This provides the Query by content alternatives: from color histograms, which are easy to extract from the image and contain little semantic content to objects, which require more sophisticated extraction techniques and are defined by their semantic content. Powerful image retrieval systems like QBIC, VIPER and IMEDIA use a complex and computationally intensive content-based algorithm for searching and indexing images in large multimedia databases. Both WebSeek and AMORE do not provide for the user to specify the scope in which the user wants to perform the query. For Example the user wants to query images of airplane in the scope of pictures. Neither of these provide for such an input from the user.

2 Architecture and Implementation

The proposed technique follows a two-phase approach. The first phase uses the query by topic specification to perform a fast and high-level filtration of pages visited according to the conceptual description associated with each image. The second phase uses the query by example method to perform a low-level content-based image match for the retrieval of smaller and relatively closer results of the query image. The figure below indicates a overview of the proposed system.

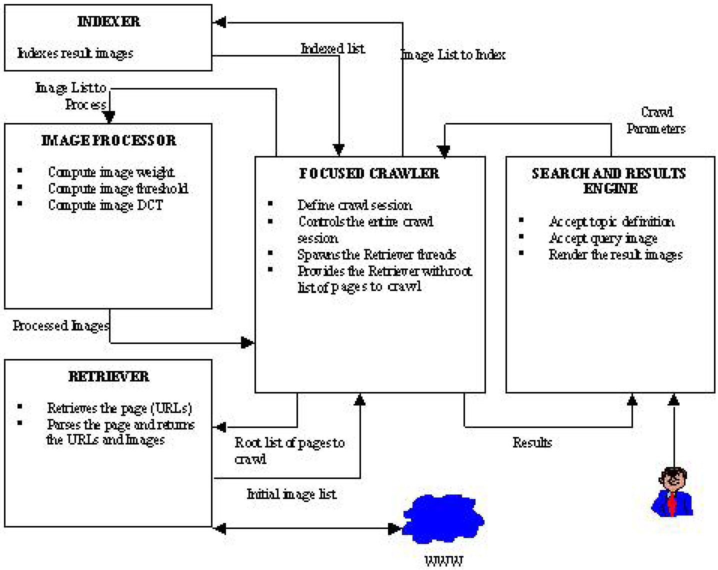


Fig. 1. Main System Components

The following are the main components of the system:

- Focused Crawler
- Retriever (Spider)
- Image Processor and Analyzer
- Indexer
- Search and Results Engine

2.1 Focused Crawler

The crawl Manager is the main entry point into the image query system. The following are the main tasks performed by this component:

- Read the crawl parameters from the properties file and accept the topic definition and query image from the user interface.
- Get Root list of URLs from a search engine (Google)
- Spawn multiple (depends on the crawl parameters) retriever threads for retrieving and parsing the pages
- Invoke the method to process the initial list of images and get the final results (images list)
- Invoke the indexer to index the result images
- Render the results to the user

2.2 Retriever

This component is responsible for the following tasks:

- Access the urls and retrieve the pages
- Parse the page (HTML document)
- Extract the links process them and if there is a match; add the links found on these pages to the queue.
- Extract the images from the potential pages and return them back to the crawler as the initial list of images
- Continue processing until all the links and sub links in the queue have been processed

2.3 Image Processor

This is the most important component of the system. It is responsible for matching and analyzing images retrieved by the Retriever. The following tasks are performed by this component:

- DCT Calculation and Comparison - Computes the DCT value of the image and compares it with the DCT values of the query image.Weight Calculation - Based on the location of the topic definition in the page (HTML document), a suitable weight is assigned to the image indicating its relevance to the query image. This is mainly used for indexing images.
- Threshold Calculation and Comparison - Computes the threshold of an image by converting it into a grayscale image and matches the image pixels with the grayscale version of query image. This is mainly used for indexing the result images.

The figure 2 illustrates the sub components of the Image Processor.

2.4 Indexer

Consists of a composite index based on image weight, threshold and DCT values. The following are the main tasks performed by this component:

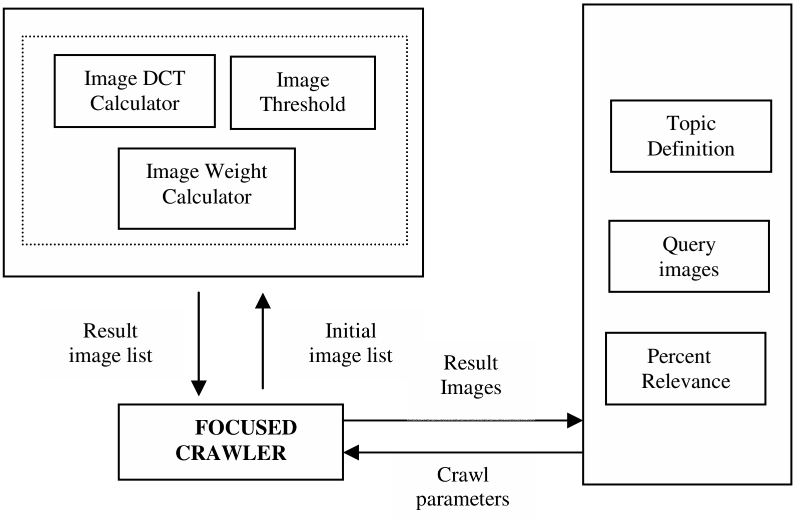


Fig. 2. Image Processor Component

- Sort images in the ascending order of image weight and DCT values and descending order of percentage relevance of threshold values
- Assign an overall ranking based on the following:

  1. Image DCT - is assigned a weight of 0.6
  2. Image weight based on occurrence of any/all of keyword/s in the image source – is assigned 0.2
  3. Image Threshold –is assigned 0.2

- Finally the images are sorted in the ascending order of the overall ranks assigned.

The following are main steps involved in querying images using the proposed system:

- Accept the topic definition, the query image and percent relevance
- Initialize the crawl session
- The crawler provides the root list of urls to the Retriever (HttpRetriever)
- Perform a topic definition match on the retrieved pages
- If there is a match, search for urls and images in that page and assign weights based on the location of the topic definition and image in the HTML document. Also computes the DCT values for each of those images.
- If the computed DCT for an image is within the user specified relevance compute the percent relevance based on the image Threshold values.
- Pass the pages (url links) to be accessed to the retriever

- Pass the images to be indexed to the Indexer
- Based on the DCT value, image weight and percent relevance index the images and display the results to the user.

3 Experiments and Results

The main aim of our research is to achieve an efficient and lightweight image search system that can be used to query images on the Internet without having to store images in a local database. A key focus of the system is to maximize the number of images relevant to the query image and minimize the number of irrelevant pages accessed and also to minimize the number of irrelevant images processed. The prototype was developed on a Pentium(R) 4, Windows XP platform. The system was implemented using Java, Struts and JSP. Note that because of the limitation around available bandwidth, we have performed "timed" experimentation below. i.e. we have run our tests for a specified amount of time(60 minutes, as opposed to running a crawl to completion). Two sets of experiments were conducted to test the efficiency and effectiveness of the proposed system. In the first experiment, we queried for airplanes and in the second experiment, we searched for flowers. Both of the experiment cases were carefully chosen to prove the scope in which the tool can be efficiently used to query the Internet for images. In experiment1, the search criteria are made more specific and percent relevance is set to a higher percent where as in experiment2 the search criteria is more relaxed. The following are the steps used in our experiments:

1. Provide a complex and a precise topic or keyword with a higher (60%) percent relevance
2. Choose the query image and submit the query. Let the crawl session run for an hour and save the results images and crawl session details
3. Provide a simple and broad topic definition or keyword with a lower (40%) percent relevance factor
4. Choose the query image and submit the query. Let the crawl session run for an hour and save the results images and crawl session details.
5. Compare and analyze the results and note the Hit ratio (number of images indexed to number of images processed); Quantity and quality of the images retrieved

Experiment 1

Keyword/Topic Definition: "AIRPLANES AND (BOEING OR AIRBUS OR JET PLANE) AND (PICTURES OR IMAGES)"

Query Image:

Percent Relevance: 60%

Results:

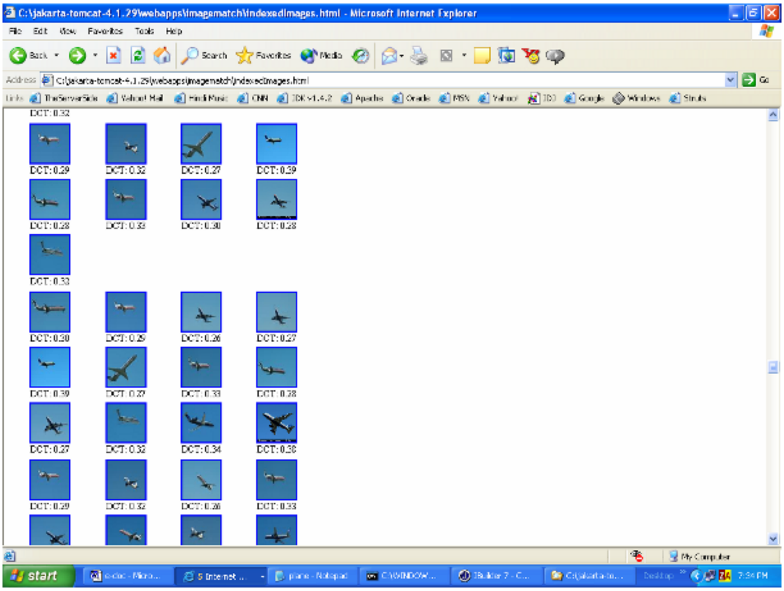


Fig. 3. Experiment 1

Analysis:

The following is the summary of the images visited, processed, indexed and discarded after running the application for about 60 minutes using 10 threads.

Number Of Pages Visited	Number Of Pages Discarded	*Number of Images Hit	Number Of Images Indexed	Number Of Images Discarded
902	471	3010	889	1932
* Note: Number of Images Hit count includes both processed as well as unprocessed images.				

Experiment 2

Keyword/Topic Definition: “FLOWERS”

Query Image:

Percent Relevance: 40%

Results:

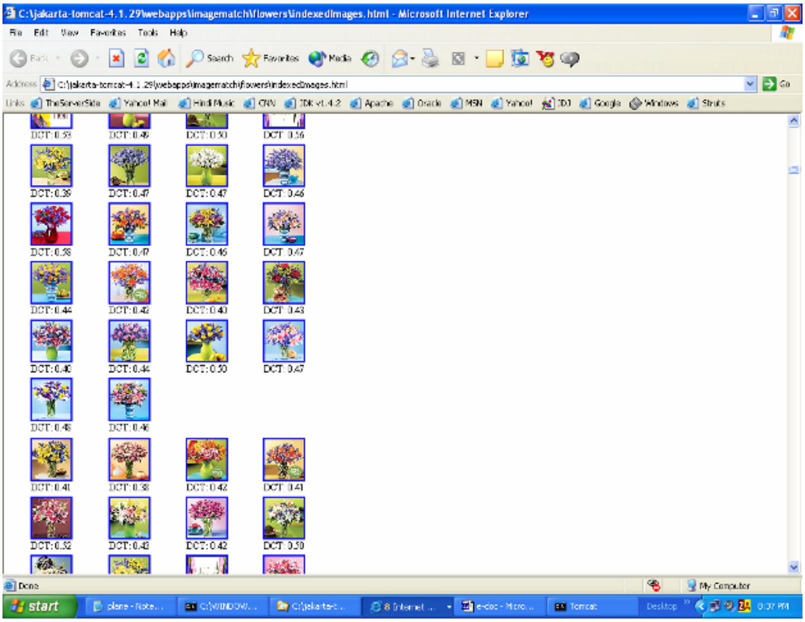


Fig. 4. Experiment 2

Analysis:

The following is the summary of the images visited, processed, indexed and discarded after running the application for about 60 minutes using 10 threads

Number Of Pages Visited	Number Of Pages Discarded	∗Number of Images Hit	Number Of Images Indexed	Number Of Images Discarded
370	31	4024	2162	1595
∗Note: Number of Images Hit count includes both processed as well as unprocessed images.				

4 Conclusions

This research presented an effective image matching and indexing technique that improvises an existing integrated image retrieval method. We used a two-phase approach, integrating query by topic and query by example specification methods. The results showed the advantage of combining the text and content-based approaches for querying real-time images on the Internet. The integrated approach produces a smaller yet, more relevant result set for a given query image. We have achieved the goal to develop a functional image search and indexing system without using a database. We also demonstrated that better retrieval results could be obtained with the proposed hybrid search technique. The following conclusions are drawn from the above two experiment cases:

Experiment1 used 60% relevance criteria to retrieve the images and about 889 good images were retrieved from 3090 images that were processed. About 1932 images were discarded as irrelevant. And in case of experiment 2, we were able to obtain 2162 result images from 4024 processed images with 40% relevance search criteria. In this case about 1595 images were considered irrelevant.

We were able to achieve this hit-ratio, because we minimized the number of images that were actually processed by discarding irrelevant pages, thus saving on computation times. The application heavily depends upon and reacts to the search criteria used. The keyword used needs to carefully chosen since it plays a major role in retrieving the relevant pages. The percent relevance criteria allow the user to control the level of relevance that he/she is interested in. Another key factor in the experiment is the query image chosen. Also, the number of threads used by the crawler affects the performance of the query process.

As seen from the above results, higher relevance, fewer and better result set. In case of experiment1 a more complex keyword or search criteria was used coupled with higher relevance (60%) resulting in smaller yet better result set. In experiment2 a simple search criteria together with lower relevance (40%) resulted in bigger result set.

5 Future Work

As an extension to this work, we propose the following –

- Automatic setting of thresholds based on user interactions: Using user inputs and learning algorithms, thresholds can be modified dynamically resulting in either finer but fewer images or coarser but many images.
- Search for other multimedia content like video and audio files: Internet is a rich source of multimedia data. Speech recognition techniques can be studied to see how they could be used to query for audio files. Challenging research problems also exits in the area of real-time video analysis and querying.

References

1. Cheng, H.D., and Sun, Y., “A Hierarchical Approach to Color Image Segmentation Using Homogeneity,” *IEEE Transactions on Image Processing*, Dec. 2001.
2. Sajjanhar, A., and Lu, G., "A grid based shape indexing and retrieval method", Special Issue of Australian Computer Journal on Multimedia Storage and Archiving Systems, November 1997, Vol.29, No.4, pp.131-140.
3. Belongie S. et al. “Color- and Texture-Based Image Segmentation Using EM and its Application to Content-Based Image Retrieval”. In *Proc. of Int. Conf. Comp. Vis*. 1998.

4. Koskela, Markrus, Laaksonen, Jonna, and Oja, Erkki, (2001). “Comparison of Techniques for Content-Based Image Retrieva”, Proceedings of the 12th Scandinavian Conference on Image Analysis (SCIA 2001), Bergen, Norway, pp. 579–586.
5. IEEE Multimedia, “The Holy Grail of Content-Based Media Analysis”, IEEE Multimedia, v.9 n.2, p.6-10, April 2002
6. Casanova, Andrea, Fraschini, Matteo, Vitulano, Sergio,, “Context: A Technique for Image Retrieval Integrating CONtour and TEXTure Information(2002)” Proceedings of the XV Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI’02)
7. Premchaiswadi, Wichian, Premchaiswadi, Nucharee, Patnasirivakin, Theianchai, Chimlek, Sutasinee and Narita, Seinosuke, Proc. “Image Indexing Technique and Its Parallel Retrieval on PVM VIIth Digital Image Computing: Techniques and Applications” 2003
8. Prasad, B. G., Gupta, S. K., and Biswas, K. K., “Color and Shape Index for Region-Based Image Retrieval” 4th International Workshop on Visual Form IWVF4, Capri, Italy, May 2001. Proceedings published in Lecture Notes in Computer Science, LNCS 2059, Springer Verlag, pp. 716 -- 725.
9. Li, Xiuqi, Chen, Shu-Ching, Shyu, Mei-Ling, and Furht, Borko, "An Effective Content-Based Visual Image Retrieval System," Proceedings of the 26th IEEE Computer Society International Computer Software and Applications Conference (COMPSAC), pp. 914-919, August 26-29, 2002, Oxford, England.
10. Ren, Jianfeng, Shen, Yuli, Guo, Lei, “A Novel Image Retrieval Based on Representative Colors” Proceedings of IVCNZ 2003
11. Lu, G., and Teng, S., "A novel image retrieval technique based on vector quantization", Proceedings of International Conference on Computational Intelligence for Modeling, Control and Automation, 17-19 Feb. 1999, Viana, Austria, pp. 36-41.
12. Lu, G. and Williams, B., "An integrated WWW image retrieval system", Australian WWW Conference, 17-20 April, 1999.
13. Furht, B., and Saksobhavivat, P., "A Fast Content-Based Multimedia Retrieval Technique Using Compressed Data," Proc. of SPIE Symposium on Multimedia Storage and Archiving Systems, Boston, MA, November 1998.
14. Ardizzone, E., Chella, A., Pirrone, R., “Shape Description for Content-based Image Retrieva”l, *Proc. of Fourth International Conference on Visual Information Systems VISUAL 2000*, November 2-4 2000, Lyon, France, Springer-Verlag, 212-222.
15. Pirrone, R., La cascia, M., “Texture Classification for Content-based Image Retrieval”, *Edoardo Ardizzone, Vito di Gesù (eds.) ICIAP 2001 11th International Conference on Image Analysis and Processing*, September 22-28 2001, Palermo, Italy, 398-403.
16. Flickner, Myron, Sawhney, Harpreet, Niblack, Wayne, Ashley, Jon, Huang, Qian, Dom, Byron, Gorkani, Monika, Hafner, Jim, Lee, Denis, Petkovic, Dragutin, Steele, David, and Yanker, Peter, “*Query by Image and Video Content: the QBIC System”* IEEE Computer 28, 9, Sep. 1995, pp. 23-32.
17. Frankel, C., Swain, M., and Athitsos, V., “WebSeer: An Image Search Engine for the World Wide Web”, , *University of Chicago Department of Computer Science Technical Report TR-96-14*, August 1996.
18. Gudivada V. N., and Raghavan, V. V., "Content-Based Image Retrieval Systems," *IEEE Computer,* 28(9), 1995, pp.18-22.
19. Faloutsos, C., Barber, R., Flickner, M., Hafner. J., Niblack, W., Petkovic, D., and Equiz, W., "Efficient and Effective Querying by Image Content," *Journal of Intelligent Information System (JIIS)*, 3(3), July 1994, pp.231-262.
20. Smith J. R., and Chang, S. F., "VisualSEEk: A Fully Automated Content-Based Image Query System," *ACM Multimedia 96,*Boston, MA, 1996.

21. Li, J., Wang, J. Z., Wiederhold, G., "Integrated Region Matching for Image Retrieval," in *Proc. of the 2000 ACM Multimedia Conf., Los Angeles, October, 2000.*
22. Boujemaa, N., Nastar, C., Content-Based Image Retrieval at the IMEDIA Group of INRIA in INRIA / Rocquencourt, BP 105, 78153 Le Chesnay, France
23. *Viper* home page: http://*viper*.unige.ch/
24. Sougata, M., Hirata, K., and Hara, Y., "AMORE: A World Wide Web image retrieval engine,"World Wide Web, vol. 2, 1999, pp. 115-132.
25. Smith, John R. and Chung, S. F., Searching for Images and Videos on the World-Wide Web, technical report #459-96-25,Department of Electrical Engineering and Center for Image Technology for New Media, Columbia University, August 19, 1996.
26. Chang, Shi-Fu, Smith, J., Beigi, M., and Ana Benitez, "Visual Information Retrieval from Large Distributed Online Repositories," Communications of the ACM, vol. 40, December 1997, pp. 63-71.
27. [Koster 1999] "The Web Robots Pages", M. Koster. 1999.
28. [Chakrabarti et al. 1999] "Focused Crawling: A New Approach to Topic-Specific Web Resource Discovery", S. Chakrabarti, M. van den Berg and B. Dom. In *Proceedings of the 8th International WWW Conference,* Toronto, Canada, May 1999.
29. Andrew McCallum, Kamal Nigam, Jason Rennie, and Kristie Seymore. 1999. "A machine learning approach to building domain-specific search engines". Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI-99), pp.662-667
30. Padmini Srinivasan, Gautam Pant, Filippo Menczer - Target Seeking Crawlers and their Topical Performance - 2002 - The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval
31. M. Diligenti, F.M. Coetzee, S. Lawrence, C.L. Giles, M. Gori, Focused Crawling Using Context Graphs (2000) - 26th International Conference on Very Large Databases, VLDB 2000
32. Soumen Chakrabartiy, Kunal Punera and Mallela Subramanyam. Accelerated Focused Crawling through Online Relevance Feedback (2002), WWW02, May 7-11, 2002, Honolulu.
33. Chakrabarti, S., Dom, B., Gibson, D., Kleinberg, J., Raghavan, P., Rajagopalan, S."Automatic resource compilation by analyzing hyperlink structure and associated text", Computer Networks and ISDN Systems, 30, 65--74, 1998.
34. Ronny Tjahyadi, Wanquan Liu, Svetha Venkatesh Department of Computer Science, Curtin University of Technology, GPO Box U1987, Perth WA 6845, Australia Application of the DCT Energy Histogram for Face Recognition

The Research on Fuzzy Data Mining Applied on Browser Records

Qingzhan Chen[1,2], Jianghong Han[1], Yungang Lai[2], Wenxiu He[2], and Keji Mao[2]

[1] School of Computer Science, Heifei University of Technology,
230009 Hefei Anhui China
[2] College of Informatin Engineering, Zhejiang University of Technology,
310032 Hangzhou Zhejiang China
qzchen@zjut.edu.cn

Abstract. With the technological advances, the Internet has been an important part of everyday life. Governmental institutions and enterprises tend to advertise and market through the internet. With the travelling records of browsers, one can analyze the preference of web pages, further understand the demands of consumers, and promote the advertising and marketing. In this study, we use Maximum Forward Reference (MFR) algorithm to find the travel pattern of browsers from web logs. Simultaneously, experts are asked to evaluate the fuzzy importance weightings for different webs. Finally, we employ fuzzy data mining technique that combines apriori algorithm with fuzzy weights to determine the association rules. From the yielded association rules, one can be accurately aware of the information consumers need and which webs they prefer. This is important to governmental institutions and enterprises. Enterprises can find the commercial opportunities and improve the design of webs by means of this study. Governmental institutions can realize the needs of people from the obtained association rules, make the promotion of policy more efficiently, and provide better services.

Keywords: Data Mining, Maximum Forward Reference, Fuzzy set theory.

1 Introduction

It is quite difficult to design a website that meets the need of consumers' browsing habits. There are some problems existing in the present webs, for example, the contents, frame and browsing path of webs are all decided according to the designers' willing instead of the browsing habits of the most users. In this study, we use the records of browsing to further understand the interests and demands of consumers and then feedback it to the decision-makers to improve the web design. From the association rules of data mining we can find out what are useful to the consumers and which pages they prefer. To the governmental institutions and enterprises, the above information is very important, because governmental institutions can know the need of people more thoroughly and take more convenient measurement, and enterprises can know how to get more commercial chances.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 527 – 535, 2005.

The paper applied fuzzy set theory on experts' weights and combined data mining algorithms to achieve the following purposes:

Find the association rules of web pages that the consumers preferred from the records of browsing;
Use the above association rules to improve the design of web to gain more chances of trading;
Improve the governmental institutions webs to make more convenience for people to get useful information.

2 Literature Review

2.1 Data Mining

Data mining combines the statistics and artificial intelligence to find out the rules that contained in the data, letters, and figures and so on by sorting and analyzing[1][6][8][12]. There are many methods of data mining including Classification, Estimation, Prediction, Affinity grouping, Clustering and Description. Among these, Affinity grouping can discover the high-frequent pattern, and discover things appeared not only frequently but simultaneously.

2.2 Fuzzy Set Theory

Fuzzy Set Theory was firstly presented by Professor L.A. Zadeh at California University in 1965[13]. It transforms the meaning and spoken description into fuzzy set instead of general set, studies and deals with subjective and undefined data with membership functions, quantifies the data and then transforms the data into useful information through systemic fuzzy operation [14].

According to the principle of fuzzy set intersection, some scholars put forward min-operand algorithm [2] in order to find out the minimum among several triangle fuzzy numbers. But the purpose of the max-operand algorithm is finding out the maximum among the several triangle fuzzy numbers. In this paper we find out the min-supported items with fuzzy set max-operand operation, and gather these filtered items as a large item sets. Next we get the large items intersection by combining each two large items with min-operand, and then take the results as the candidate item sets.

2.3 Maximum Forward Reference

When the user wants to find the interesting information, he or she shifts one page to another, and if the user backs the page, we suppose that he or she is just for convenience not for browsing. We just mine the Forward Reference Sequence of Access Patterns, and take the back browsing as the finishing of Forward Reference Path. We call this Forward Reference Sequence as Maximum Forward Reference [4] [5] [8] [9] [10].

When uses browse the web, the browsing path will be kept on the web log in the server. We can analyze uses' action on the website through the browsing path. And also we can mine the Maximum Forward Reference from web logs.

3 Methods and Steps

There are two parts in the Apriori Algorithm; the first part is scanning data base to find out the support of every item then filtering the Minimum support item according the given Minimum support; the second one is setting Minimum Confidence, the main purpose is checking the association rules. We use fuzzy max-operand to compare with two triangle fuzzy items on filtering the above Minimum support and Minimum confidence.

On the process of filtering large item sets in Apriori, we must combine each two low power large item sets to generate high power candidate item sets. The process of combination is finding the intersection of two triangle fuzzy numbers, so we use fuzzy min-operand to deal with the intersection to get a new one. The study bases on Apriori algorithm of data mining technology, and uses Maximum Forward Reference to analyze the browsing path to take out the pages users are most interested in, then combines Fuzzy Set Theory, adds the fuzzy weight, calculates the weight between each two candidate item sets with fuzzy min-operand just like using experts weight dealing with triangle fuzzy items, at last, compares the items with min-support and min-confidence with fuzzy max-operand.

Now we can find out the association rules of each pages of web. To the governmental institutions, it can advance the performance of serving; to the enterprises, it can improve the design of web and provide more useful information according to the browsing action and the interesting information of users.

The process of this web mining including: coding, extracting original data, giving weight by experts, transforming the pages into triangle fuzzy numbers according to the experts weight, calculating the support, giving minimum support, filtering candidate itemsets to get large itemsets, combining the large itemsets as candidate itemsets, giving minimum confidence, checking minimum confidence and minimum support, then forming the association rules.

4 Example

According to the above algorithm, we take the web logs of one web sever as example to explicate the whole steps of generating the association rules.

Step 1: Coding

There are seventeen pages in our website, default.htm, product.htm, enterprise_intro.htm and so on. We code A, B, C…Q as the page of the above seventeen pages.

Step 2: Extracting original data

In the web logs of server, some user's browsing path is {A,B,C,B,H,B,C,D,C,F,C,D,E,D,C,G,C,B,H,I,J,A,K,L,M,L,N,L,P,O,Q}. We get the Maximum Forward Reference Sequence in the browsing path in table 1.

Table 1. The Maximum Forward Reference Sequence of the browsing path

NO	The results of mining
1	ABC
2	ABH
3	ABCD
4	ABCF
5	ABCDE
6	ABCD
7	ABCG
8	ABC
9	ABHIJ
10	AKLM
11	AKLN
12	AKLPOQ

We sort all users' Maximum Forward Reference Sequence with above method and put them into the database.

Step 3: Giving weight by experts

We ask experts to evaluate the website, and according to the weight we divide the pages into five groups, very important, important, common, unimportant and extraordinary unimportant, as shown in table 2.

Table 2. The weight of pages

Page Weight	The NO. of page
very important	A,B,C,H,K,L
important	D,E,I,N
common	F,G
unimportant	J,M
extraordinary unimportant	O,P,Q

Step 4: Transforming the pages into triangle fuzzy numbers according to the experts' weight

We transform the pages into triangle fuzzy numbers according to the experts' weight as shown in table 3.

Table 3. The triangle fuzzy number of pages' weight

Page Weight	Triangle Fuzzy Number
very important	（0.75,1,1）
important	（0.5,0.75,1）
common	（0.25,0.5,0.75）
unimportant	（0,0.25,0.5）
extraordinary unimportant	（0,0,0.25）

Step 5: Calculating the support

We calculate every page's support according to the Maximum Forward Reference Sequence in table 1. Firstly, we count up the appearing times of every page, the results are shown in table 4.

Table 4. The appearing times of pages

Pages	Times	Pages	Times
A	12	I	1
B	9	J	1
C	7	K	3
D	3	L	3
E	1	M	1
F	1	N	1
G	1	O	1
H	2	P	1
		Q	1

Secondly, we calculate the proportion of the appearing times of every page in all records, and multiply the corresponding fuzzy weight, the result is the fuzzy support of every page. The process of transforming is shown in table 5.

Table 5. Transforming the item support into fuzzy support

Item	The proportion of item appearing times	Fuzzy weight	Fuzzy support（*SX*）
A	12/12=1	(0.75,1,1)	(0.75,1,1)
B	9/12=0.75	(0.75,1,1)	(0.563,0.75,0.75)
C	7/12=0.583	(0.75,1,1)	(0.437,0.583,0.583)
D	3/12=0.25	(0.5,0.75,1)	(0.125,0.188,0.25)
E	1/12=0.083	(0.5,0.75,1)	(0.042,0.062,0.083)
F	1/12=0.083	(0.25,0.5,0.75)	(0.02,0.042,0.062)
G	1/12=0.083	(0.25,0.5,0.75)	(0.02,0.042,0.062)
H	2/12=0.167	(0.75,1,1)	(0.125,0.167,0.167)
I	1/12=0.083	(0.5,0.75,1)	(0.042,0.062,0.083)
J	1/12=0.083	(0,0.25,0.5)	(0,0.02,0.042)
K	3/12=0.25	(0.75,1,1)	(0.188,0.25,0.25)
L	3/12=0.25	(0.75,1,1)	(0.188,0.25,0.25)
M	1/12=0.083	(0,0.25,0.5)	(0,0.02,0.042)
N	1/12=0.083	(0.5,0.75,1)	(0.042,0.062,0.083)
O	1/12=0.083	(0,0,0.25)	(0,0,0.02)
P	1/12=0.083	(0,0,0.25)	(0,0,0.02)
Q	1/12=0.083	(0,0,0.25)	(0,0,0.02)

Step 6: Giving the Minimum Support

Here we suppose the percentage of Minimum Support is 30%, and multiply it by the important triangle fuzzy number (0.5 0.75 1), the result is the Minimum Support (SMIN), so

SMIN=30 %× (0.5,0.75,1) =(0.15,0.225,0.3).

Step 7: Filtering the candidate itemsets to get the large itemsets

We take the max-operand operation of the fuzzy support from step5 and the minimum support from step 6 to get the minimum support filtered items and gather them as large itemsets. We take page A as example to explain the process of the filtering. From step 5, we can know the support of page A, SA=(0.75,1,1), the minimum support SMIN=(0.15,0.225,0.3), and according to the triangle fuzzy operation we can know $S_A > S_{MIN,.}$

From the above example we can know that page A can become one of the large itemsets by minimum support filtering. According to the above process of calculating, we can get 1- large itemset is {A, B, C}.

Step 8: Combining the large itemsets as candidate itemsets

We can get the intersection of large items by combining the each two large items with triangle fuzzy min-operand. We call the result of candidate as 1- large itemset. We take the page A and page B as examples, the intersection of SA and SB is I, I (a, b, c) = (0.563, 0.75, 0.75).

According to the above operation, we can get 2-Candidate itemset by taking triangle fuzzy min-operand operation of each two large items of 1-large itemset. Then repeat the step 7and step 8 until no other candidate itemsets can be generated; we can get all combination of large items and support as shown in table 6.

Table 6. The combination and support of large items

Single large item	Support
A	(0.75,1,1)
B	(0.563,0.75,0.75)
C	(0.437,0.583,0.583)
The combination of two large items	Support
AB	(0.563,0.75,0.75)
AC	(0.437,0.583,0.583)
BC	(0.437,0.574,0.583)
The combination of three large items	Support
ABC	(0.437,0.574,0.583)

Step 9: Giving the Minimum Confidence

Here we suppose the percentage of minimum confidence is 40%, and multiply it by the important triangle fuzzy number (0.5, 0.75 1) , the result is the minimum confidence, so C_{MIN}=40%× (0.5,0.75,1)=(0.2,0.3,0.4).

Step 10: Checking the confidence to get the association rules

During the course of checking, we use conditional probability C=*Prob* (Y|X) =P(X∩Y)/P(X). If the probability of Y ≥mini-confidence on the condition that X is appeared, an association rule is generated which accords with minimum confidence. If there is an association rule exists between X and Y, it can be expressed as If X Then Y, means that if X appears, Y will appear too. In the operation results of our example, we get the combination of {AB, AC, BC, ABC}, so there are five kinds of association rules.

IF A then B; IF A then C; IF B then C,
IF A then B and C; IF A and B then C,

Among the above association rules, take IF A then B as an example, on the condition of appearing A, the probability of B is:

Prob (B|A) =P (A∩B)/P (A) =9/12=0.75
Then we multiply the *Prob* (B|A) =0.75 by the triangle fuzzy number of the intersection of A and B calculated by fuzzy minimum operator:
0.75× (0.563, 0.75, 0.75) = (0.422, 0.5625, 0.5625)

So we get the confidence of rule IF A then B is:

(0.422, 0.5625, 0.5625)

We separately calculate the conditional probability and confidence of other combinations and check them with the minimum confidence supposed in step 9. We use fuzzy maximum operator to check them, firstly, we get the two maximum triangle numbers; secondly, we compare the similitude between maximum and triangle fuzzy numbers. The results are shown in table 7.

Table 7. The possible association rules and their confidence

Association Rule	Conditional Probability	Confidence	Minimum Confidence	Existing association rule Or not
IF A then B	0.75	(0.422,0.563,0.563)	(0.2,0.3,0.4)	Yes
IF A then C	0.583	(0.255,0.34,0.34)	(0.2,0.3,0.4)	No
IF B then C	0.778	(0.34,0.447,0.454)	(0.2,0.3,0.4	Yes
IF A then B and C	0.583	(0.255,0.335,0.34)	(0.2,0.3,0.4)	No
IF A and B then C	0.778	(0.34,0.447,0.454)	(0.2,0.3,0.4)	Yes

In this example we finally get the following three association rules:

IF A then B; IF B then C; IF A and B then C

We take “IF A then B” as an example, this rule means that if user browses page A, he or she will browse page B probably, on the other words, it means that the relation between page A and page B is very close. To the users, there are useful information existing on page A and page B, so the probability of continuing visiting page A and page B is very high. To the enterprises, if they know the rule, they must strengthen the convenience between page A and page B, in order to make more convenience for users.

5 Conclusion

If we want to run enterprises webs and governmental institution webs successfully, we must know the demands of clients. The study uses web logs in the server, puts forward a feasible fuzzy data mining algorithm, analyzes the interests and the habits of users, and helps the decision-makers to improve the webs. The process of web mining presented here includes the following steps: coding, extracting original data, giving weight by experts, transforming the pages into triangle fuzzy numbers according to the experts weight, calculating the support, giving minimum support, filtering candidate itemsets to get large itemsets, combining the large itemsets as candidate itemsets, giving minimum confidence, checking minimum confidence and minimum support, then forming the association rules.

Acknowledgement

The Project Supported by Zhejiang Provincial Science Foundation of China (601081) and Key Lab of Information Processes of Jiangsu Province (Suzhou University).

References

[1] Berry, J., Michael, A. and Linoff Gordon, S. “Data Mining Techniques: for marketing, sales and customer support,” John Wiley & Sons, Inc.,1997.

[2] Chuang, T. N. and Kung, J. Y. “A new approach for the fuzzy shortest path problem Department of Information Management,” International Conference on Fuzzy Systems and Knowledge Discovery (FSKD'02), vol. 1, pp. 381-384, Nov. 18-22, Orchid Country Club, Singapore, 2002.

[3] Drott, M. C. “Using Web Logs to Improve Site Design,” Proceedings on the Sixteenth Annual International Conference on Computer Documentation, pp.43-50, 1998.

[4] Garofalakis, M. N., Rastogi, R., Seshadri, S. and Shim, K. “Data Mining and the Web: Past, Present and Future,” Proceedings of The Second International Workshop on Web Information and Data Management, pp.43-47, 1999.

[5] Han, J. Pei, J. Mortazavi–asl B ., and Zhu, H. “Mining Access Patterns Efficiently from Web Logs,” Proc. 2000 Pacific-Asia Conf. on Knowledge Discovery and Data Mining (PAKDD’00), Kyoto, Japan, April 2000.

[6] Han, J. and Kamber, M. “Data mining concepts and techniques,” Morgan Kaufmann Publishers, 2001.

[7] Mendel, J. M. “Fuzzy logic systems for engineering: a tutorial”, Proceedings of the IEEE, 83 pp.345-377, 1995.

[8] Chen, M.-S., Han, J. and Yu, P.S. “Data Mining: An Overview from a Database Perspective”, IEEE Transactions on Knowledge and Data Engineering, Vol.8, No.6, pp.866-883, 1996.

[9] Chen, M.-S., Park, J.S. and Yu, P.S. “Data Mining for Path Traversal Patterns in a Web Environment”, IEEE Proceeding of the 16th ICDCS, pp.385-392, 1996.

[10] Chen, M.-S., Park, J.S. and Yu, P.S. “Efficient Data Mining for Path Traversal Patterns”, IEEE Transactions on Knowledge and Data Engineering, Vol.10, No.2, pp.209-221, 1998.

[11] Woon, Y.T., Jacobsen, M., Hector, G. M., and Dayal, U. “From User Access Patterns to Dynamic Hypertext Linking,” The Fifth International World Wide Web Conference, Paris, France, 1996.

[12] Ye, X. and Keane, J. A. “Mining Association Rules with Composite Items”, Systems, Man, and Cybernetics. Computational Cybernetics and Simulation. IEEE International Conference Vol. 2, pp. 1367–1372, 1997.

[13] Zadeh, L. A. “Fuzzy Sets”, Information and Control, Vol.8, pp. 338-353. 1965.

[14] Zadeh, L. A. “The concept of a linguistic variable and its application to approximate reasoning, ” Information Science, Vol. 8. 199-249 (I), 301-357 (II), 1975.

Discovering Conceptual Page Hierarchy of a Web Site from User Traversal History

Xia Chen[1], Minqiang Li[2], Wei Zhao[1], and Ding-Yi Chen[3]

[1] School of Electronic and Information Engineering
[2] School of Management,
Tianjin University, Tianjin 300072, P.R. China
[3] School of Information Technology and Electrical Engineering,
University of Queensland, QLD 4072, Australia
{jchenxia, mqli}@tju.edu.cn

Abstract. A Web site generally contains a wide range of topics which provide information for users who have different access interests and goals. This information is not randomly scattered, but well organized under a hierarchy encoded in the hyperlink structure of a Web site. It is intended to mold the user's mental models of how the information is organized. On the other hand, user traversals over hyperlinks between Web pages can reveal semantic relationships between these pages. Unfortunately, the link structure of a Web site which represent the Web designer's expectation on visitors may be quite different from the organization expected by visitors to this site. Discovering the conceptual page hierarchy from a user's angle can help web masters to have an sight into real relationships among the Web pages and refine the link structure of the Web site to facilitate effective user navigation. In this paper, we propose a method to generate a conceptual page hierarchy of a Web site on the basis of user traversal history. We use maximal forward references to model user's traversal behavior over the underlying link hierarchy of a Web site. We then build a weighted directed graph to represent the inter-relationships between Web pages. Finally we apply a "*Maximum Spanning Tree*" (MST) algorithm to generate a conceptual page hierarchy of the Web site. We demonstrate the effectiveness of our approach by conducting a preliminary experiment based on a real world Web data.

1 Introduction

The power of the Web lies in the myriad of links that are possible with hypertext linkage of documents. A Web site generally contains a wide range of topics which provide information for users who have different access interests and goals. This information is not randomly scattered, but well organized under a hierarchy encoded in the link structure of the site. This underlying page organization structure represents the Web designer's expectation on users traversal to this site. Visitors, therefore, can find what they want by following hyperlinks in each Web pages by navigation. However, designing a good Web site is not a simple

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 536–543, 2005.

task because hypertext structures can easily expand in a chaotic manner as the number of pages increased, the underlying organization structure of a Web site may be quite different from the organization expected by visitors to this site. Therefore, discovering the conceptual page hierarchy which is consistent with the user cognitive pattern can help web masters to determine the relationship among the web pages, refine the link structure of the site and facilitate effective user navigation so as to provide better service for Web users.

The Web is fundamentally hierarchical in its organization, although this may not always be apparent. These hierarchies are themselves encoded in the "URLs" and are visually represented by hyperlinks between pages. This hierarchical organization is a necessity on the Web. Most sites depend on hierarchies, moving from the most general overview of the site (the home page), down through increasingly specific submenus (the index page) and content pages. It simulates the user's mental models of how the information is organized. Fig.1 clearly illustrates this kind of underlying link hierarchy in a Web site.

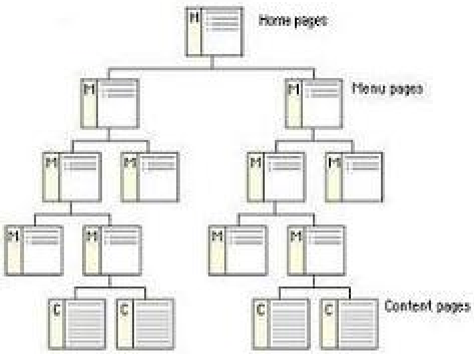


Fig. 1. An illustration of link hierarchy in a Web site

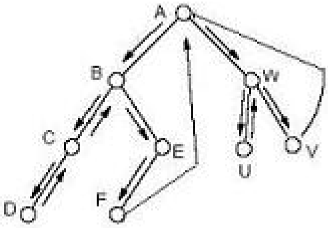


Fig. 2. Example for User Traversal Path

On the other hand, the Web user can also follow this link hierarchy to find out information. Generally speaking, the user traversing to a Web site usually follows a static pattern as follows. When they log onto a Web site and look for some interesting pages to read, the "Home Page" usually is their starting point (exclusive of some random visits). There normally are n links on each page which are pointing to another n pages respectively. From the context, especially anchor-text, users can find out which link most likely could guide them to their interest point and make a selection by clicking that link and enter a new page. Otherwise, they could click the "backtrack" button to revisit previous pages and re-determine their forward selection or simply give up this traversal and end up session. This pattern is repeated recursively during their access session. Fig.2 shows a common traversal path of a normal user on a Web site.

We observed that the user's traversal process to a Web site in fact a depth-first traversal over the implicit conceptual page hierarchy which starts from "homepage" (root), and following a hyperlink (edge) enter an "index page" (sub-menu), until finding the expected information (content page/leaf node), otherwise, they use the "backward" icon and enter another forward selection. Some pages might be revisited not because of their content, but for location. Fortunately, the Web server can automatically record every users' traversal history in its log files.

Chen et.al in their paper [1] first proposed the idea of using "[illegible]" to capture the above traversal pattern of Web users. A maximal forward reference is defined as the longest consecutive sequence of forward references before the first backward reference is made to visit some previously visited pages during a particular server session. Thus, the last reference in a maximal forward sequence indicates a content page that is desired by the user. Under such understanding , when a user searches for desired information, her information needs can be modelled by a set of maximal forward references occurred during her search process.

The basic assumption of our proposed approach is that we think a maximal forward reference can be taken as a depth-first traversal of the Web user over the underlying conceptual page hierarchy of the Web site. The pages in a MFR must be hierarchically conceptually related. Hence, we use a weight directed graph to represent the relationships between pages and the frequency of total visits between two pages are used as the weight to measure semantic relationship between them. Then a hierarchical clustering algorithm - Maximum Spanning Tree (MST) algorithm is applied to automatically group web pages into a hierarchical structure.

This paper is organized as follows: In the next section we will introduce some related works. In section 3, we explain our proposed approach in details. After that, we briefly describe our preliminary experiment. The last section outlines conclusions.

2 Related Works

2.1 Web Pages Clustering and Web Log Mining

Web pages may be considered similar based on their filenames, their locations in the site hierarchy, keywords frequency, or their correlation in visitor paths etc. Recently, Web log mining has been widely used in web pages clustering.

Web log mining is the process of applying data mining techniques to the discovery of patterns from the Web log files in the Web site. A Web log file is a collection of records of user requests for documents on a Web site. The basic assumption of relevant studies is the co-occurred pages in user visits on the web site which are conceptually-related.

Conventionally, Web documents are usually represented in an N-dimensional Web access user vectors [2] [3] and after applying a traditional clustering algorithm such as Self-Organization Map [4] or K-means [5], these documents could be partitioned into a set of mutual exclusive clusters. However, a Web server usually contains hundreds, thousands or even millions of access users at a certain period of time. It is obviously impractical to represent each web page as a high dimensional vector in which each dimension represents a user. In addition, each user most likely only accesses a very small part of pages on the site in one visit. Hence, the high dimensionality and sparseness of vector representation would cause clustering algorithms very inefficient and severely decreases the accuracy of clustering results.

To address this problem, Perkowitz et al. [6] proposed a PageGather algorithm, which creates a similarity matrix between pages according to their co-occurrence frequency in the visits and then use a graph to partition pages. Sue et al. [7] applied the same similarity function in his RDBC algorithm (a variant of DBSCAN algorithm). Mobasher et al. [8] combined hyper-graph partitioning and association rules to cluster the web documents.

On the other hand, People have also began to realize the gap between the Web site designer's expectations and visitor behavior and tried to find a solution to identify the gap.

Nakayama et al. [9] use the inter-page conceptual relevance to estimate the designer's expectation, and the inter-page access co-occurrence to estimate visitors. They focus on Web site design improvement by using multiple regression to predict hyperlink traversal frequency from page layout features. Srikant et al. [10] first take use of visitors' "backtrack" action to help find improper linked pages.

2.2 Hierarchical Clustering with Spanning Tree Algorithms

In this section, we give a brief introduction to hierarchical document clustering using spanning tree algorithm.

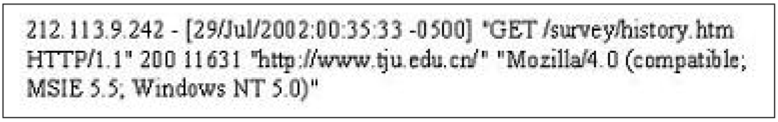
Hierarchical clustering constructs trees of clusters of objects, in which any two clusters are disjoint, or one includes the other. The cluster of all objects is the root of the tree. Minimum/Maximum spanning tree (MST) is one of the well-known techniques of hierarchical clustering [11]. MST uses the graph theory concept to connect a set of points and minimize/maximize the total length of connecting lines. A spanning tree of a graph is a set of $(n-1)$ edges that connect all the n objects of the graph without cycles. The MST is the set of edges with a minimum/maximum sum of the lengths over the $(n-1)$ edges. Most famous algorithms are the Kruskal algorithm (1956) and Prim's Algorithm (1957). The basic idea of these algorithms is to keep adding the smallest/largest cost edge to this tree until it is connected. Chu et. al [12] give a more efficient algorithm of finding a MST on a directed graph.

3 Conceptual Page Hierarchy Construction

In this section, we present a method for constructing the conceptual page hierarchy of a Web site from log files which are automatically generated by Web servers. It includes three main steps: data preprocessing, data modelling and Web page clustering.

3.1 Data Preprocessing

Most Web servers can provide log files in the [illegible] (CLF) or [illegible] (ECLF). Both formats include information such as the client IP (internet protocol) address, access time, request method, the URL of the page accessed and so on. Fig.3 shows an entry record in an ECLF.



```
212.113.9.242 - [29/Jul/2002:00:35:33 -0500] "GET /survey/history.htm
HTTP/1.1" 200 11631 "http://www.tju.edu.cn/" "Mozilla/4.0 (compatible;
MSIE 5.5; Windows NT 5.0)"
```

Fig. 3. An Entry Record in an ECLF

In our approach, only requested Web pages and the corresponding references within this site are used for construction of its page hierarchy. We, therefore, remove all irrelevant entries. We then scan the Web log sequentially, and it generates user sessions on the fly. A user session is organized as an ordered sequence of web pages the user visited on the Web site within a particular sever session. For example, the user session {A,B,C,D,E,F,W,U,V} stands for the user arriving to the Web site and travelling on this site from page 'A' then 'B', 'C', ..., until 'V' by order. As long as the user sessions are generating, maximal forward references within that session can be extracted. As the Web sever can't record users' backtracking action, sometimes MFR extraction algorithms need to consult the link structure of the Web site to distinguish the forward and back tracking point. Simply put, the data preprocessing results in a set of MFRs, such as {A,B,C,D}, {A,B,E,F}, {A,W,U} and {A,W,V}.

3.2 Graph-Based Representation of Web Pages

We will use a connected, weighted directed graph to represent all Web pages which only appeared in MFRs as well as their semantic inter-relationships between pages. Through this graph-based representation, we can convert a conventional multi-dimensional clustering problem to a spanning tree construction problem.

Let $D = \{p_i\}$ be a set of Web pages which occurred in MFRs. We define a connected directed graph $G(V, E)$, where the vertex set $V = \{p_i | p_i \in D\}$ and the edge set $E = \{< u, v > | u, v \in D \text{ and } u \neq v\}$, represents a set of possible inter-connections between pairs of two pages. We say two pages u and v are connected, denoted as $< u, v >$, if and only if $< u, v >$ is an ordered subsequence of a maximal forward reference. The edge also indicates there is a link that exists between two pages. Each $edge(u, v) \in E$ has a weight $weight(u, v)$ which is defined as the probability of user traversals from the page "u" to the page "v".

$$weight(u, v) = P(v|u) = \frac{N_{u,v}}{\sum_k N_{u,k}} \tag{1}$$

where $N_{u,v}$ is the no. of times "u" is followed by "v", $\sum_k N_{u,k}$ is the sum of times on all the out-links of page "u".

Fig.4 is a sample of our weighted page graph based on a set of maximal forward references. After the weighting process, we can clearly see that some traversal paths might be never used by the visitors, such as "$E \rightarrow G$" in fig.4 because of weight(E,G)=0. So it is one of main reasons why this kind of analysis could be helpful to improve the design of a Web site.

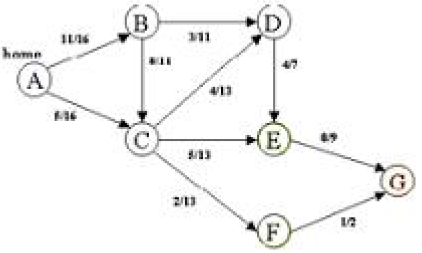


Fig. 4. A Sample of Weight Directed Page Graph

3.3 MST-Based Web Page Clustering Algorithm

We use a "Maximum Spanning Tree" algorithm to automatically generate a conceptual page hierarchy from a weighted directed page graph. Our goal is to find an acyclic sub-graph of "G" with specified root "r", a tree that connects all of the vertices and whose total weight is maximized. We slightly modified the MST algorithm proposed by Chu et.al in [12] to meet our goal. The pseudo code of this algorithm is described as follows in Fig.5.

The computational complex of this algorithm is $O(V^2)$.

3.4 Preliminary Experiment

We used a Web log file from a real commercial Web site (www.nbzc.net) to evaluate the effectiveness of our method. We collected all log entries on the 7th of May generated by their WWW Server. It contained 445727 requesters. In data pre-processing, we identified 3421 users, 14889 user sections, and 2536 Web pages. After applying the MFR extraction algorithm, we obtained 3961 MFRs and the average length was 4. At last, we used our proposed clustering algorithm to generate a page hierarchy. In order to find out the differences between the hyperlink hierarchy of the site which is designed by Web designer and the conceptual page hierarchy which is expected by users, we used a breadth-first search

```
Maximum Spanning Tree Algorithm
Input: G=(V,E) with arc costs $w_e$ for all arcs $e \in E$, and a root node $r \in V$
Output: T
Begin

 1. Discard the arcs entering the root "$r$" if any;
    Let $i = 0$, and let $G_0 = G$, $T_0 = \emptyset$.
 2. For each node $v \neq r$ that has no arc in $T_i$ directed into it in $G_i$:
      – Let $w_v$ be the maximum cost of an arc into $v$.
        For every arc $< u, v >$ directed into $v$, modify its cost to be
        $(w_{<u,v>} - w_v)$ (so subtract $w_v$ from the costs of all arcs into $v$).
        Now pick any zero-cost arc into $v$ to add to $T_i$.
 3. If no directed cycle formed by $T_i$ in the current digraph $G_i$, then
    $T_i$ forms a spanning arborescence of $G_i$, and we stop. Otherwise, continue.
 4. For each cycle formed by $T_i$, contract the nodes in the cycle into a pseudo-node.
    Let the newly formed digraph be called $G_{i+1}$, $T_{i+1}$ be the arcs in $T_i$ which weren't
    inside sets that were shrunk (i.e. arcs which didn't belong to cycles) and i=i+1.
    go back to Step 2.

End.
```

Fig. 5. Maximum Spanning Tree Algorithm

Table 1. The Comparing Results of Conceptual Page Hierarchy with Link Hierarchy

	Result
total number of Web pages	2536
The total levels of link hierarchy	10
The total levels of conceptual page hierarchy	6
The number of un-visited pages	56
The number of redundant links	176
The average reduction rate of links on each level	22%

strategy to generate the overall topological link structure of this site and then made a make a comparative analysis. The results are briefly showed in Table 1.

4 Conclusion

In this paper, we have presented a new method for discovering page hierarchy of web site from user traversal history. The first problem we encountered is noisy data. It may be resulted from random visit: Some visitors request pages in the site randomly without any purposes; or the highlight links directly pointing to the content Web pages on the homepage also may affect the accuracy of the result. Another problem is a multi-labelling problem. Semantically, a Web page can belong to more than one category. However, currently our proposed algorithm did not take this situation into account. In our future work, we are going to concentrate on finding a proper solution to address this problem.

Acknowledgement

We would like to express our thanks to Ningbo-Window Network Information Co., Ltd for providing us the log files.

References

1. Chen, M., Park, J., Yu, P.: Efficient data mining for path traversal patterns. IEEE Trans. on Knowledge and Data Engineering (TKDE) (1998)
2. Zeng, H.J., Chen, Z., Ma, W.Y.: A unified framework for clustering heterogeneous web objects. In: WISE. (2002)
3. Chen, M., LaPaugh, A., Singh, J.P.: Categorizing information objects from user access patterns. In: the Eleventh International Conference on Information and Knowledge Management. (2002)
4. Kath A.Smith, A.N.: Web page clustering using a self-organizing map of user navigation patterns. Decision Support Systems, Special issue: Web data mining **35** (2003)
5. Shahabi, C., Zarkesh, A.M., Adibi, J., Shah, V.: Knowledge discovery from users web-page navigation. (In: IEEE Workshop Research Issues in Data Engineering) 20–29
6. Perkowitz, M., Etzioni, O.: Adaptive web sites: Automatically synthesizing web pages. (In: the Fifteenth National Conf. on Artificial Intelligence (AAAI)) 727–732
7. Su, Z., Yang, Q., Zhang, H.J., Xu, X., Hu, Y.H.: Correlation-based document clustering using web logs. In: the 34th Hawaii International Conference On System Sciences(HICSS-34). (January 3-6,2001)
8. Mobasher, B., Cooley, R., Srivastava, J.: Automatic personalization based on web usage mining. In: Technical Report, TR99-010, Department of Computer Science, Depaul University. (1999)
9. Nakayama, T., Kato, H., Yamane, Y.: Discovering the gap between web site designers' expectations and users' behavior. In: the Ninth Int'l World Wide Web Conference, (Amsterdam)
10. Srikant, R., Yang, Y.: Mining web logs to improve website organization. In: WWW. (2001)
11. Rohlf, F.J.: Algorithm 76: Hierarchical clustering using the minimum spanning tree. Computing (1973)
12. Y.J.Chu, T.H.Liu: On the shortest arborescence of a directed graph. Science Sinica (1965)

Bayesian Neural Networks for Prediction of Protein Secondary Structure

Jianlin Shao[1], Dong Xu[2], Lanzhou Wang[1], and Yifei Wang[2]

[1] College of Life Sciences, China Jiliang University, Hangzhou,
Zhejiang Province, China, 310018
Colin_shao@cjlu.edu.cn, wlzcim@163.com
[2] College of Sciences, Shanghai University, Shanghai, China, 200444
xstone@citiz.net, yifei_wang@staff.shu.edu.cn

Abstract. A novel approach is developed for Protein Secondary Structure Prediction based on Bayesian Neural Networks (BNN). BNN usually outperforms the traditional Back-Propagation Neural Networks (BPNN) due to its excellent ability to control the complexity of the model. Results indicates that BNN has an average overall three-state accuracy Q_3 increase 3.65% and 4.01% on the 4-fold cross-validation data sets and TEST data set respectively, comparing with the traditional BPNN. Meanwhile, a so-called *cross-validation choice of starting values* is presented, which will shorten the burn-in phase during the MCMC (Markov Chain Monte Carlo) simulation substantially.

1 Introduction

With the completion of the Human Genome Project and other sequencing projects, a large number of genome and translated protein sequences have been accumulated. However, the available information about the protein structure is scare and this is widening the protein sequence-structure gap rapidly, which has restricted the discovery of protein structures and functions significantly. Therefore, three-dimension protein structure prediction from sequences is one of the most urgent and important tasks in molecular biology via bioinformatics techniques and Protein Secondary Structure Prediction from sequences continues to rise [1-3] as a result of the fact that protein secondary structure plays a significant role in the steady conformation of proteins.

In this study, we have systematically investigated the Bayesian approach for neural networks and employed Bayesian Neural Networks for Protein Secondary Structure Prediction.

2 Material and Method

2.1 Training and Testing Data Sets

For training and testing our model, we employ parts of the two classical data sets known as RS126 [1] and CB396 [2]. RS126 is screened to remove proteins that are

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 544 – 551, 2005.

shorter than 80 residues with the purpose of better non-redundant and no-homologous and 95 proteins are left to use for cross-validation training for our model. 16 proteins called TEST are drawn out from CB396 randomly for testing our method (Table 1). Training and testing data sets including Multiple-sequence alignment profiles are available at http://www.sander.ebi.ac.uk/hssp/.

Table 1. TEST data set from CB396 for testing our method

TEST data set	1cei 1cem 1cpn 1fua 3pgk 2sil 1udh 2bat 2cpo 2end 2gsq 2phy 2tgi 3chy 821p 1svb

2.2 Coding Scheme and Secondary Structure Definition

We adopt the local coding scheme of the protein sequence with a sliding window. Here, an input window of 7 amino acids is of utilization. For a single sequence, each amino acid is sparsely encoded and represented by binary strings of 21 bits, i.e. the first 20 bits for the amino acid type and the 21st bit to specify N- and C-terminal ends. As a consequence, the dimensionality of the input vector is 21×7. In addition, the assignment of secondary structure to experimentally determined 3D structures is performed by DSSP[4], which distinguishes the secondary structure into 8 categories: H (α- helix), G (3_{10} helix), I (π- helix), E (extended β- strand), B (isolated β- strand), T (turn), S (bend), and coil ("_"). In this study, the 8 structure classes are reduced into 3 classes: $\{\mathrm{H, G, I}\} \in \mathrm{H}$, $\{\mathrm{E, B}\} \in \mathrm{E}$, all other states to C. H, E, C are encoded by (1,0,0), (0,1,0), (0,0,1) respectively.

2.3 Bayesian Neural Networks

Bayesian approach for Neural Networks was pioneered by Buntine & Weigend [5] and reviewed by Mackay [6] and Neal [7]. Bayesian learning for Neural Networks can adjust regularization term coefficient online to control the complexity of the model. Under the Bayesian framework, the traditional BPNN presents better generalization ability, especially for small sample training data set.

Consider MLP (Multi-Layer Perceptron) for classification. For the $k-$ class problems, $C_1, \cdots, C_k$ correspond to k output units of the MLP Neural Networks. Let $x_n = \left(x_n^1, \cdots, x_n^m\right)$ denote input feature vector, and $y_n = \left(y_n^1, \cdots, y_n^k\right)$ denote target output vector and $\|y_n\| = 1$, where indices n and m/k correspond to sample sequence and the dimensionality of the input/output vector, respectively. Thus, when n belongs to C_k, $y_n^k = 1$; on the contrary, $y_n^k = 0$. For the sake of illustration of outputs under the probability framework, k softmax functions, denoting k output units, are adopted as follow

$$f_k\left(x_n, w\right) = \frac{\exp\left(g_k\left(x_n, w\right)\right)}{\sum_{i=1}^{k} \exp\left(g_i\left(x_n, w\right)\right)}, \tag{1}$$

where $g_k\left(x_n, w\right)$ denotes the actual value of the $k-$th output unit. From (1), probability density for the target vector y_n is then

$$p\left(y_n \left| x_n, w\right.\right)=\prod_{i=1}^{k} f_i\left(x_n, w\right)^{y_n^i} \ . \tag{2}$$

Thus, by Eq. (2), a Neural Networks model for classification can be regarded as a multinomial probability model, each sample with respect to a multinomial distribution given by

$$y_n=\left(y_n^1, \cdots, y_n^k\right) \sim M\left(1, f_1, \cdots, f_k\right) \ . \tag{3}$$

Let $D=\left\{\left(x_n, y_n\right)\right\}$, $\left(n=1, \cdots, N\right)$ denote the training data set, then D 's likelihood function $L\left(D \left| w\right.\right)$ is

$$L\left(D \left| w\right.\right)=p\left(D \left| w\right.\right)=\prod_{n=1}^{N} \prod_{i=1}^{k} f_i\left(x_n, w\right)^{y_n^i} \ . \tag{4}$$

The objective of traditional BPNN is maximization of Eq. (4) via some kind of iteration algorithms and only one classifier will be involved, which is on equality with minimizing the following term

$$E\left(w\right)=-\ln p\left(D \left| w\right.\right)=-\sum_{n=1}^{N} \sum_{i=1}^{k} y_n^i \ln f_i\left(x_n, y_n\right) \ . \tag{5}$$

Therefore, BPNN usually restricts the generalization ability of the model. Bayesian learning for Neural Networks can overcome above limitation and reduce the risk of overfitting. Under the Bayesian framework, all parameters in the Neural Networks are viewed as random variables with respect to some prior distribution before training, and the posterior distribution of the parameters is updated through data by Bayes' rule. After that, we can obtain the final predictive distribution

$$p\left(y_n \left| D\right.\right)=\int p\left(y_n \left| x_n, w\right.\right) p\left(w \left| D\right.\right) dw \ . \tag{6}$$

Let prior distribution for all parameters be $p\left(w\right)$ before obtaining any of training data. We can define a regularization term (several regularization terms) as

$$R\left(w\right)=-\ln p\left(w\right) \ , \tag{7}$$

which controls the complexity of the model. From (5) and (7), by Bayes' rule we can obtain

$$\begin{aligned}\ln p\left(w \left| D\right.\right) &= \ln p\left(D \left| w\right.\right)+\ln p\left(w\right)+C \\ &= -E\left(w\right)-R\left(w\right)+C\end{aligned} \ , \tag{8}$$

where C is a constant. Let the sum function of networks error be

$$U\left(w\right)=E\left(w\right)+R\left(w\right) \ . \tag{9}$$

by Eq. (8), the posterior distribution for w is

$$p\left(w|D\right)=\frac{1}{Z}\exp\left(-U\left(w\right)\right) \ , \tag{10}$$

where Z is a normalizing constant. The final Bayesian Neural Networks classifier is given by for a new input x_{N+1}

$$\begin{aligned}\hat{y}_{N+1}^{k} &= \int f_k\left(x_{N+1},w\right)p\left(w|D\right)dw \\ &= \int f_k\left(x_{N+1},w\right)\frac{1}{Z}\exp\left(-U\left(w\right)\right)dw\end{aligned} \ , \tag{11}$$

where $\hat{y}_{N+1}^{k}$ denotes the probability that x_{N+1} belongs to C_k . Evaluating the integral of Eq. (**11**) is a difficult task due to the complexity of the posterior distribution for the parameters and generally approximated with parametric approximation as Gauss approximation or with numerical approximation as MCMC.

Here, Protein Secondary Structure Prediction corresponds to 3-class problem. three-layer MLP is adopted with 21×7 input units, 28 hidden layer units and 3 output units, each layer having a bias unit and a Logistic activation function

$$f\left(x\right)=\frac{1}{1+e^{-x}} \ . \tag{12}$$

Four regularization terms are added to the sum function of error $U\left(w\right)$ for the controlling of complexity, i.e. w is grouped by w_{ij} , w_{jk} , $theta1_j$, $theta2_k$, denoting connected weights of input-hidden layer, hidden-output layer, bias-hidden layer, bias-output layer. Thus, function $U\left(w\right)$ is

$$\begin{aligned}U\left(w\right) &= E\left(w\right)+R_1\left(w_{ij}\right)+R_2\left(w_{jk}\right)+R_3\left(theta1_j\right)+R_4\left(theta2_k\right) \\ &= E\left(w\right)+\alpha_1\sum_{i,j}\frac{1}{2}w_{ij}^2+\alpha_2\sum_{j,k}\frac{1}{2}w_{jk}^2+\alpha_3\sum_{j}\frac{1}{2}theta1_j^2+\alpha_4\sum_{k}\frac{1}{2}theta2_k^2\end{aligned} \ , \tag{13}$$

where hyper-parameter α_1 , α_2 , α_3 , α_4 are weight decay term coefficients, specified by the precision of Gaussian prior distribution for each group weights.

During the implementation, we have used the single point Metropolis-Hasting algorithm with a normal proposal distribution for networks parameters and Gibbs sampling for hyper-parameters. The sampling algorithm runs as follows:

Step0, set starting value $w^{(0)}$ and $\alpha^{(0)}$;

Step1, fix $\alpha^{(0)}$, sampling $w^{(1)}$ from the proposal distribution and compute the $U\left(w^{(1)}\right)$ by (**9**), if $U\left(w^{(1)}\right)\le U\left(w^{(0)}\right)$ then {accept $w^{(1)}$ as current state ;}

else {accept $w^{(1)}$ by the probability $a\left(w^{(0)},w^{(1)}\right)=\exp\left(U\left(w^{(0)}\right)-U\left(w^{(1)}\right)\right)$ } ;

Step2, fix $w^{(1)}$, sampling $\alpha^{(1)}$ with Gibbs;

Step3, return *step1* and *step2* until Markov Chain converges.

3 Results

In this study, we employ 4-fold cross-validation data sets grouped by 95 training proteins (Table 2). The performances of the final classifier are given on them and `TEST` data set. Several prediction performance measures [1] are utilized including three-state overall per-residue accuracy Q_3 , Matthew's correlation coefficients (C_H , C_E , C_C) and the per-residue accuracy for each type of secondary structure (Q_H , Q_E , Q_C ; Q_H^{pre} , Q_E^{pre} , Q_C^{pre}).

Convergence diagnosis must be performed for Markov chain during the simulation. BGR [8], one of the most popular convergence assessment techniques for Markov Chain Monte Carlo is adopted in this study. Meanwhile, Deviance Information Criterion (DIC) [9] value is computed for determining the start point of the predictive model.

Table 2. The database of non-homologous proteins used for 4-fold cross-validation. All proteins have less than 25% pairwise similarity and lengths>80 residues

A	6cpa 5cpp 7cat_A 7rsa 1eca 256b_A 3sdh_A 4gcr 8abp 2sns 4cpv 5cyt_R 1paz 4bp2 6tmn_E 1l58 2wrp_R 9pap 2hmz_A 2ccy_A 1fkf 1fnc 1s01 2alp 2cyp 2gdm 2ltn_A
B	6acn 6cts 3cla 2fvx 2pab_A 2pcy 3rnt 5er2_E 9wga_A 1lmb_3 2gbp 2ak3_A 2tsc_A 1acx 1bbp_A 1gp1_A 1hip 1rbp 2cab 2i1b 2lhb 2rsp_A 2sod_B 3blm 5hvp_A
C	5lyz 1fxi_A 3cd4 3cln 4cms 2fxb 2gn5 2wsy_A 2wsy_B 4xia_A 4gr1 4pfk 6dfr 7icd 8adh 1cc5 1fdl_H 1rhd 2gd1_O 2stv 3gap_A 4ts1_A
D	1tnf_A 1azu 1lap 1mcp_L 2phh 5ldh 2aat 3pgm 3tim_A 3hmg_B 1r09_2 2tmv_P 1pyp 9api_A 4rhv_1 4rhv_3 1bmv_1 1bmv_2 2gls_A 3hmg_A 1etu

Let weight groups be $w1[i,j]$, $w2[j,k]$, $theta1[j]$, $theta2[k]$, hyper-parameters be $\alpha[l]$, where $i = 21\times 7$, $j = 28$, $k = 3$, $l = 4$. $w1[i,j]$, $w2[j,k]$, $theta1[j]$, $theta2[k]$ all have a Gauss prior distribution with mean 0 and corresponding precision $\alpha[l]$. Hyper-parameter samples are drawn from their full conditional distribution $Gamma(0.5, 0.05)$ and ordered "overrelaxation" method [10] is adopted to eliminate dependencies between samples. For the convenience of Convergence diagnosis, four independent chains are produced during each cross-validation training.

Markov chains haven't been diagnosed to converge until BGR ratio approximates 1. After convergence for all the chains, more iterations are usually performed for the estimation of all parameters. The ultimate predictive value is based on the expectation of the models that all the parameters are drawn from stationary Markov chains, but how to select the starting point for the predictive models is very significant. In this paper, we selected the starting point for the models via DIC value, which gives the Bayesian measures of model complexity and fit, and the model with the smallest DIC is estimated to be the model that would best predict a replicate dataset of the same structure as that currently observed (**Fig. 1**).

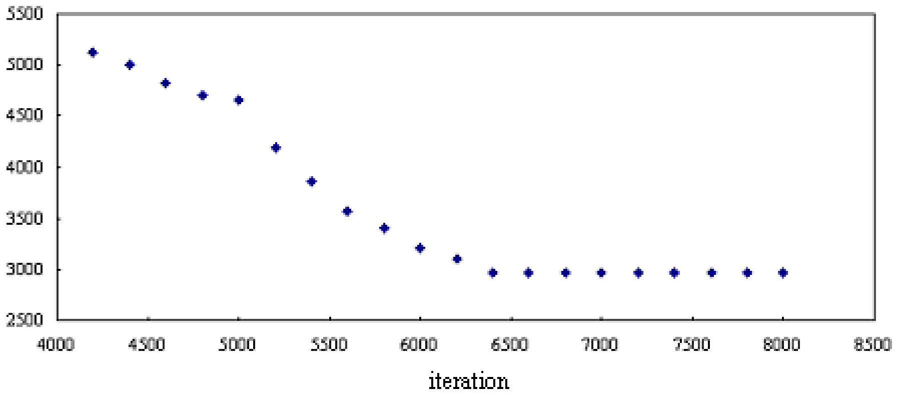
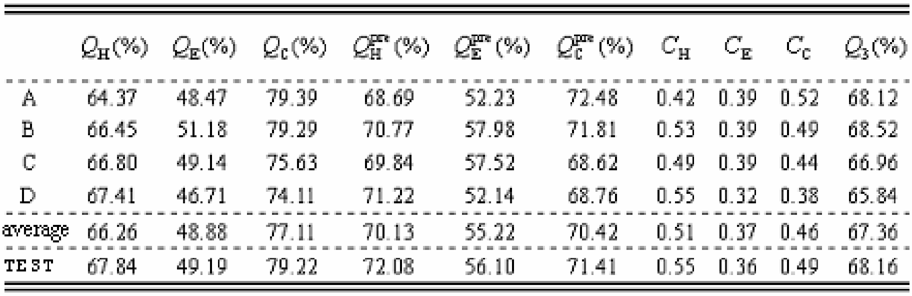


Fig. 1. Scatter plot of DIC value. The above plot denoted the DIC value on the cross-validation data set A with the B, C, and D training data sets

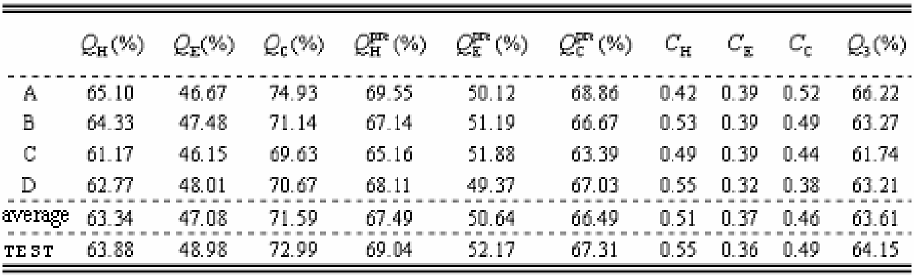
For the comparison of the performance between BNN and BPNN, we train them under the same architecture on the same data sets. Results indicate that the BNN outperforms the BPNN (Table 3 and Table 4).

Table 3. The testing accuracy of BNN on each cross-validation data set and `TEST` data set



	Q_H(%)	Q_E(%)	Q_C(%)	Q_H^{pre}(%)	Q_E^{pre}(%)	Q_C^{pre}(%)	C_H	C_E	C_C	Q_3(%)
A	64.37	48.47	79.39	68.69	52.23	72.48	0.42	0.39	0.52	68.12
B	66.45	51.18	79.29	70.77	57.98	71.81	0.53	0.39	0.49	68.52
C	66.80	49.14	75.63	69.84	57.52	68.62	0.49	0.39	0.44	66.96
D	67.41	46.71	74.11	71.22	52.14	68.76	0.55	0.32	0.38	65.84
average	66.26	48.88	77.11	70.13	55.22	70.42	0.51	0.37	0.46	67.36
TEST	67.84	49.19	79.22	72.08	56.10	71.41	0.55	0.36	0.49	68.16

Table 4. The testing accuracy of BPNN on each cross-validation data set and `TEST` data set



	Q_H(%)	Q_E(%)	Q_C(%)	Q_H^{pre}(%)	Q_E^{pre}(%)	Q_C^{pre}(%)	C_H	C_E	C_C	Q_3(%)
A	65.10	46.67	74.93	69.55	50.12	68.86	0.42	0.39	0.52	66.22
B	64.33	47.48	71.14	67.14	51.19	66.67	0.53	0.39	0.49	63.27
C	61.17	46.15	69.63	65.16	51.88	63.39	0.49	0.39	0.44	61.74
D	62.77	48.01	70.67	68.11	49.37	67.03	0.55	0.32	0.38	63.21
average	63.34	47.08	71.59	67.49	50.64	66.49	0.51	0.37	0.46	63.61
TEST	63.88	48.98	72.99	69.04	52.17	67.31	0.55	0.36	0.49	64.15

In theory, if the Markov chain is irreducible, the choice of starting values will not affect the stationary distribution. However, sampling often requires a very long burn-in phase because many parameters occur in the BNN model and high dependencies between parameters present.

A study of the relationship between burn-in phase and the starting values is made in this paper. We set the starting values of MCMC simulation with the parameter values of the final BPNN model via cross-validation. Results indicate that the burn-in phase is shortened substantially (**Fig. 2** and **Fig. 3**). The method of selecting starting values is called the *cross-validation choice of starting values* here.

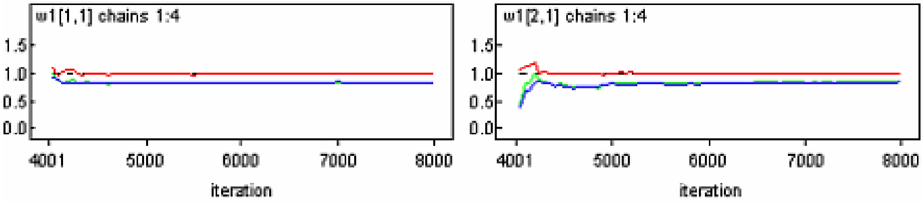


Fig. 2. BGR convergence plot for stochastic starting values. The above is the BGR convergence plot of part parameters before employing cross-validation starting values on B, C, and D training data sets (red line denotes BGR ratio and the chain convergences when BGR ratio approximates 1)

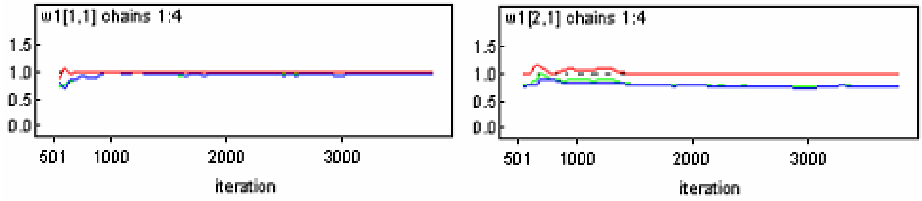


Fig. 3. BGR convergence plot for *cross-validation choice of starting values*. The above is the BGR convergence plot of part parameters after employing cross-validation starting values on B, C, and D training data sets (red line denotes BGR ratio and the chain convergences when BGR ratio approximates 1). From the plot, burn-in phase is shortened substantially

4 Conclusion

This paper addresses the utilization of BNN as a classifier in Protein Secondary Structure Prediction. Results show that BNN has better generalization ability than traditional BPNN. BNN possesses many unique modeling characteristics, including controlling of weights magnitude via hyper-parameters and etc. In addition, Bayesian learning for Neural Networks has the advantages of model selection and model comparison via DIC.

Acknowledgement

This work is supported by the National High Technology Research Development Program of China (863 Program) under grant No.2002AA23402 and the Natural Science Foundation of China Jiliang University under grant No.XZ0443.

References

1. Rost, B., Sander, C.: Combining Evolutionary Information and Neural Networks to Predict Protein Secondary Structure. PROTEINS: Structure, Function, and Genetics. 19 (1994)55-72
2. Cuff, J. A., Barton, G. J.: Evaluation and Improvement of Multiple Sequence Methods for Protein Secondary Structure Prediction. PROTEINS: Structure, Function and Genetics.34 (1999) 508-519
3. Guo, J.,Chen, H., Sun, Z.R., Lin, Y.D.: A Novel Method for Protein Secondary Structure Prediction Using Dual-Layer SVM and Profiles. PROTEINS: Structure, Function and Bioinformatics.54 (2004)738-743
4. Frishman, D., Argos, P.: Knowledge-based secondary structure assignment. Proteins: Struct. Funct. Genet.23 (1995)566-579
5. Buntine, W.L., Weigend, A.S.: Bayesian back propagation. Complex Systems.5 (1991) 603-643
6. Mackay, D.J.C.: A practical Bayesian framework for back propagation networks. Neural Computation. 4(1992)448-472
7. Neal, R.: Bayesian Learning for Neural Networks. Springer-Verlag, Berlin (1996)
8. Brooks, S.P., Roberts, G.O.: Convergence assessment techniques for Markov chain Monte Carlo. Statistics and Computing.8 (1998)319-335
9. Spiegelhalter, D.J., Best, N.G., Carlin, B.P., *et al.*: Bayesian measures of model complexity and fit (with discussion). Journal of Royal Statistic B. 64(2002)583-640
10. Neal, R.: Suppressing random walks in Markov chain Monte Carlo using ordered over-relaxation. In Learning in Graphical Model, Jordan M I, ed. Kluwer Academic Publishers, Dordrecht (1998)205-230

PromPredictor: A Hybrid Machine Learning System for Recognition and Location of Transcription Start Sites in Human Genome*

Tao Li and Chuanbo Chen

College of Computer Science & Technology,
Huazhong University of Science & Technology, Wuhan 430074, China
ljrlt@public.wh.hb.cn, chuanboc@163.com

Abstract. In this paper we present a novel hybrid machine learning system for recognition of gene starts in human genome. The system makes predictions of gene start by extracting compositional features and CpG islands information from promoter regions. It combines a new promoter recognition model, coding theory, feature selection and dimensionality reduction with machine learning algorithm. Evaluation on Human chromosome 4, 21, 22 was 64.47% in sensitivity and 82.20% in specificity. Comparison with the three other systems revealed that our system had superior sensitivity and specificity in predicting gene starts. PromPredictor is written in MATLAB and requires Matlab to run. PromPredictor is freely available at www.whtelecom.com/Prompredictor.htm.

1 Introduction

The publication and preliminary analysis of the human genome sequence [1,2] marks a significant milestone in the field of molecular biology. One of the main goals of the Human Genome Project is the characterization, annotation—recognition and categorization of genes from human genome to serve as a periodic table for biomedical research [3]. In the past few years, many efforts have been devoted to gene annotations. The National Center for Biotechnology Information (NCBI), Ensembl and Golden Path, for instance, provided the initial annotations, but the whole process of annotation is expected to go on for many years, and the current gene annotations only refer to protein-coding regions, relatively few tools have been developed to identify the regulatory regions required for the correct transcriptional activity of the genome. This task is particularly difficult in the case of eukaryotic organisms in which regulatory regions represent a small percentage, overwhelmed by presumably non-functional DNA. So prediction and characterization of regulatory regions is still a challenging problem. Here, we focus on detecting promoters, which are in the class of regulatory regions.

A promoter is the region of genomic sequence proximal to the transcription start site (TSS) responsible for the initiation of transcription. In spite of the fact that

* Supported by the National High Technology Research and Development Program(863 program)(No. 2001AA231071)

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 552 – 563, 2005.

characterizing regulation of gene expression is an important aspect of understanding gene function, for most human genes, promoters have not been defined or studied. Therefore, reliable recognition and characterization of promoters is a high priority goal in human genome study. Knowledge of promoters will be useful in elucidating regulation and expression mechanisms of genes and may shed light on the function of novel and uncharacterized genes.

A well-established measure for promoter prediction accuracy scores a TSS prediction as positive if it is within the range of 200 bp upstream to 100 bp downstream of the true TSS [4]. Several research groups have developed methods for in silico promoter prediction, including knowledge-based methods, comparative genome analysis as well as methods based on statistical-compositional properties of DNA sequences, for reviews see [4,5]. For most methods, the false-positive rate is roughly estimated at one per kilobase. In another aspect, the ratio of true prediction to false prediction is a small percent, with the exception of one method, PromoterInspestor, which shows predicted accuracy of 43% [6]. In recent years, many efforts have been devoted to improve promoter predicted accuracy by using CpG islands association [7,8,9,10], combination with exon/intron/3'-UTR predictions [11,12,13] and consensus promoter identification that combines several existing methods [14].

Motivated by these methods, we developed a new hybrid neural network system—the PromPredictor for human genome promoter recognition. It is a combination of a novel promoter recognition model, coding theory, feature selection and dimensionality reduction with machine learning algorithm. The method is based on the statistical concept of pentamer distributions in specific functional regions of DNA and selected the most significant pentamer vocabularies from training sequences by an unsupervised learning technique, in addition to CpG islands features.

2 Methods

This section describes the proposed prediction system—PromPredictor. We introduce compositional feature extraction and dimensionality reduction in Section 2.1. Section 2.2 describes two CpG islands features. Section 2.2 presents architecture of the prediction system. Finally we discuss system training and parameter optimization methods.

2.1 Feature Extraction

It is well known that genomes are characterized by species-specific compositional features, and that coding and non-coding DNA are distinguishable in terms of their pentamer and hexamer distributions [15]. In promoter regions except core promoter elements such as TATA boxes, CAAT boxes and transcription initiation sites (INR), there exists a couple of other individual elements or sequence properties that are associated with promoter sequences. Among these are higher CpG content—CpG islands [16], secondary structure elements like the HIV-1 TAR regions [17], cruciform DNA structures [18], or simple direct repeats [19]. In order to capture core elements as well as weak signals, we use pentamer frequency coding method.

The pentamer encoding method extracts various patterns of five consecutive nucleic acids in a DNA sequence and counts the number of occurrences of the extracted pentamer. For instance, given a DNA sequence CGAATCG, the pentamer encoding method gives the following results: 1 for CGAAT (indicating CGAAT occurs once), 1 for GAATC and 1 for AATCG. For each DNA sequence, there are 4^5=1024 possible pentamers in total.

If all the 1024 pentamers were chosen as input features of the neural network, it would require many weight parameters and training data, which makes it difficult to train the neural network. Different methods have been proposed to solve the problem by careful feature selection and by scaling of the input dimensionality [20]. What we are proposing here is to select relevant features by employing a distance measure to calculate the relevance of each feature [21].

Let X be a feature and x be its value. Let $p(x|Class=1)$ and $p(x|Class=0)$ denote the class conditional density functions for feature X, where *Class*_1 represents the positive class and *Class*_0 is the negative class. Let $D(X)$ denote the distance function between $p(x|Class=1)$ and $p(x|Class=0)$, defined as [22]:

$$D(X) = \int \left| p(x \mid Class = 1) - p(x \mid Class = 0) \right| \mathrm{dx} \tag{1}$$

The distance measure prefers feature X to feature Y if $D(X)> D(Y)$. Intuitively, this means that it is easier to distinguish between *Class*_1 and *Class*_0 by observing feature X than by observing feature Y. That is, X appears often in *Class*_1 but seldom in *Class*_0 or vice versa. In our work, each feature X is a pentamer. Let c denote the occurrence number of the feature X in a sequence S. Let l denote the total number of pentamers in S and $len(S)$ represents the length of S. We have $l = len(S)$-4. Define the feature value x for the pentamer X with respect to the sequence S as:

$$x = \frac{c}{len(S) - 4} \tag{2}$$

Since a promoter sequence may be short, random pairings can have a large effect on the result. $D(X)$ in Formula 1 can be approximated by the Mahalonobis distance [23] as:

$$D(X) = \frac{(m_1 - m_0)^2}{d_1^2 + d_0^2} \tag{3}$$

where m_1 and d_1 (m_0 and d_0, respectively) are the mean value and the standard deviation of the feature X in the positive (negative, respectively) training data set. Intuitively, in Formula 3, the larger the numerator is (or the smaller the denominator is), the larger the interclass distance is, and therefore the easier to separate *Class*_1 from *Class*_0 (and vice versa). The mean value m and the standard deviation d of the feature X in a set ψ of sequences are defined as:

$$m = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{4}$$

$$d=\sqrt{\frac{1}{N-1}\sum_{i=1}^{N}(x_i-m)^2} \tag{5}$$

where x_i is the value of the feature X with respect to sequence S_i□ ψ, and N is the total number of sequences in ψ.

Let X_1, X_2,…, $X_{N\mathrm{a}}$, N_a<1024, be the top N_a features (pentamers) with the largest $D(X)$ values. Intuitively, these N_a features occur more frequently in the positive training data set and less frequently in the negative training data set. For each DNA sequence S (whether it is a training or an unlabeled test sequence), we examine the N_a features in S, calculate their values as defined in Formula 2, and use the N_a feature values as input feature values to the HNN for the sequence S.

To compensate for the possible loss of information due to ignoring the other pentamers, a linear correlation coefficient (*LCC*) between the values of the 1024 pentamers with respect to the DNA sequence S and the mean value of the 1024 pentamers in the positive training data set is calculated and used as another input feature value for S. Specifically, the *LCC* of S is defined as:

$$LCC(S)=\frac{1024\sum_{i=1}^{1024}x_i\overline{x_i}-\sum_{i=1}^{1024}x_i\sum_{i=1}^{1024}\overline{x_i}}{\sqrt{1024\sum_{i=1}^{1024}x_i^2-(\sum_{i=1}^{1024}x_i)^2}\sqrt{1024\sum_{i=1}^{1024}\overline{x_i^2}-(\sum_{i=1}^{1024}\overline{x_i})^2}} \tag{6}$$

where $\overline{x_i}$ is the mean value of the i th pentamer, $1\le i\le 1024$, in the positive training data set, and x_i is the feature value of the i th pentamer with respect to S as defined in Formula 2.

2.2 CpG Islands Features

In the human genome, many genes were recognized and validated successfully [1,2] by using the so-called CpG islands as gene markers. CpG islands are unmethylated segments of DNA longer than 200 bp, with a G + C content of at least 50%, and the number of CpG dinucleotides being at least 60% of what could be expected from the G + C content of the segment [24,25,26,27]. CpG islands are found around a gene that starts in approximately half of mammalian promoters [26,27] and are estimated to be associated with ~60% of human promoters [28]. For this reason, Pedersen [29] suggested that CpG islands could represent a good global signal to locate promoters across genomes. At least in mammalian genomes, CpG islands are good indicators of gene presence. In our prediction system, we use two CpG island features—G+C content and ratio of expected to observed CG dinucleotides (Obs/Exp). Let *len* represent the length of one segment of a DNA sequence, the G+C content (*GC_con*) and Obs/Exp (*o*/*e*) [25] are defined as:

$$GC\_con=\frac{\text{number of C's }+\text{number of G's}}{len} \tag{7}$$

$$o/e = \frac{\text{number of CG's} * len}{(\text{number of C's } * \text{number of G's})} \tag{8}$$

2.3 Architecture of the Prediction System

The conceptual structure of our system is depicted in Figure 1–2. The overall system shown in Figure 1 comprises a collection of four basic classifiers: Promoter classifier, Exon classifier, Intron classifier and 3'-UTR classifier. Each of the classifiers is a modified BP neural network and has the same structure. The basic classifier of promoter is shown in Figure 2. Each classifier is trained by different training sets and the parameters for each classifier are optimized independently.

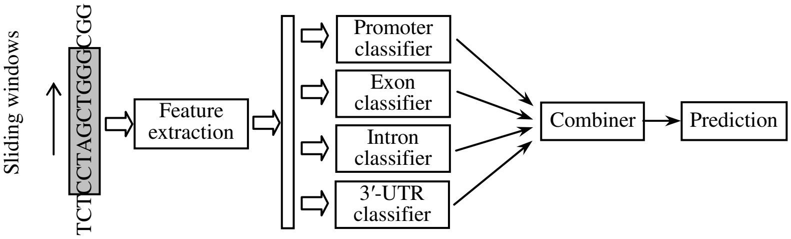


Fig. 1. Overall structure of the PromPredictor

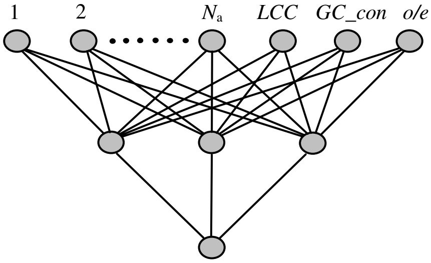


Fig. 2. Promoter_Classifier

An unknown sequence is partitioned into windows 250 bp long, shifted by 1 bp. For each sliding data window, we compute the feature values following procedures in the previous section and these feature values are used as input of the hybrid neural network.

The prediction system assigns the sequence to the class promoter if three classifiers—Exon, Intron, 3'-UTR decide that the sequence is not an exon, intron, 3'-UTR respectively, and only Promoter classifier decides that the sequence belongs to this class.

2.4 System Training and Parameter Optimization

From the vertebrate section of the Eukaryotic Promoter Database (EPD), V 79.0 [30], promoter sequences from 200 bp upstream to 50 bp downstream of the TSS were taken. Exon and intron sequences were taken from Exon–Intron Database [31], 3'-UTR sequences were extracted from the UTR database [32]. Sequence training set for the four basic classifiers (promoter, exon, intron and 3'-UTR) were created by randomly extracting non-overlapping sequences of 250 bp from the four database mentioned above. Redundant sequences were deleted by the program CLEANUP [33] which resulted in sets consisting of 1837 sequences from promoter regions, 5400 exon sequences, 6500 intron sequences and 6300 3'-UTR sequences. In these four sets, 2/3 of the sequences were used for training, and the rest were used for validation.

According to the definitions above, the number of N_a, the neuron number of hidden layers and the training algorithm must be determined for each classifier. Furthermore, an optimal assignment threshold must be calculated.

The training for each classifier is independent. For example, the positive training set for Exon classifier is exon, and the negative training set includes intron, promoter and 3'-UTR. We tested the Exon classifier's performance with different parameters and different training algorithms and recorded the optimized parameters based on accuracy and computation time. The training algorithms include Gradient descent algorithms, Conjugate gradient (CGB) algorithms [34], Quasi-Newton algorithm, One Step Sccant (OSS) algorithm [35], Resilient backpropagation (RPROP) algorithm [36] and Levenberg-Marquardt (LM) algorithm [37]. Table 1 summarizes the default threshold, corresponding training algorithm and the optimized parameters for four basic classifiers. To test the effect of the four classifiers, we used the validation set to evaluate the performance of four classifiers. Table 2 shows the results. In this test,

Table 1. Parameters and default threshold for four basic classifiers

Basic classifiers	N_a	Neuron numbers in hidden layer	Training algorithm	default threshold
Promoter_classifier	900	3	LM algorithm	0.9368
Exon_classifier	800	2	RPROP algorithm	0.9470
Intron_ classifier	800	2	RPROP algorithm	0.9353
3'-UTR_classifier	800	2	RPROP algorithm	0.9170

Table 2. Results of the four classifiers and PromPredictor on validation sequences

Sets	Number of sequences	Promoter classifier	Exon classifier	Intron classifier	3'-UTR classifier	PromPredictor
Promoter	612	484	20	74	81	423
Exon	1800	32	1782	247	254	12
Intron	2166	85	678	2144	912	19
3'-UTR	2100	344	455	528	1887	159
Sensitivity: $S_e(\%)$		51.22				69.00
Specificity: $S_p(\%)$		79.08				69.12
Correlation Coefficient: $CC(\%)$		63.64				69.06

Promoter classifier detected 484 promoters in 612 promoter sequences, but in other three datasets, Promoter classifier made 32, 85, 344 false predictions. The other three classifiers were designed to detect these false predictions. PromPredictor is the combination of four classifiers. After combination, we can see that PromPredictor can improve the sensitivity in predicting promoter regions obviously.

One class of classifier assigns an unknown sequence to this class when the predicted value is above threshold. In Table 2 four basic classifiers' threshold are all 0.8. The result of Promoter classifier on the validation set is 51.22% in sensitivity and 79.08% in specificity. While PromPredictor (each classifier's threshold is 0.8) is 69.00% in sensitivity and 69.12% in specificity. The combination of four classifiers achieves a correlation coefficient (69.06%) greater than the Promoter classifier. The sensitivity, specificity and correlation coefficient are defined as:

$$S_e = \mathrm{TP}/(\mathrm{TP}+\mathrm{FN}) \tag{9}$$

$$S_p = \mathrm{TP}/(\mathrm{TP}+\mathrm{FP}) \tag{10}$$

$$cc = \frac{\mathrm{TP}\times\mathrm{TN} - \mathrm{FP}\times\mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})\times(\mathrm{TP}+\mathrm{FN})\times(\mathrm{TN}+\mathrm{FP})\times(\mathrm{TN}+\mathrm{FN})}} \tag{11}$$

TP is true positive, FP is false positive, TN is true negative and FN is false negative.ï

The result of PromPredictor is the boundaries for the predicted promoter region. We define the position of predicted TSS as:

$$TSSpos = \frac{upstreamBound + downstreamBound}{2} + 75 \tag{12}$$

3 Results

To evaluate the performance of PromPredictor, we compared PromPredictor to in silico TSSs prediction tools using the three human chromosomes from the review by V.B. Bajic [13]. Chromosome 4, 21, and 22 were selected for the analysis because of their different G + C contents in order to better understand the behavior of our system and the other three system, Dragon GSF [13], FirstEF [8] and Eponine [38].

The sequences and annotation data of chromosomes 4 and 21 were downloaded from NCBI GenBank (http://www.ncbi.nlm.nih.gov/). The updates for the two chromosomes were Aug 24, 2004. The reference TSS locations of the two chromosomes was obtained from DBTSS (http://dbtss.hgc.jp/) mapped on genomic contig. These mappings are based on promoter sequences (-1000, +200) of human known genes identified by full-length cDNAs. The chromosome 22 sequence and the latest annotation data are based on Collins et al. [39], and the sequence was obtained from the World Wide Web at http://www.sanger.ac.uk/HGP/Chr22/. The total length of chromosomes 4, 21, and 22 used in the analysis is 187,161,218 bp, 34,171,998 bp, and 34,748,585 bp, respectively.

The rule of the counted hits (predictions) is bases on the review by V.B. Bajic [40]. Hits are counted as strand specific. For PromPredictor we used the default threshold and the predicted TSS as defined in Formula 12. For Dragon GSF we used the predictions of the gene start (the position denoted by identifier “GS” in the report file) as the TSS prediction. For FirstEF we used the position of the TSS as determined by the point 500 nt downstream of the first nucleotide of the promoter region predicted by FirstEF, which is in accordance with the explanation provided in the original publication on FirstEF. When FirstEF makes cluster predictions, we used only the highest ranked prediction as correct. For Eponine, we used all predictions as reported because we did not know how big the gap should be, and what should be considered as the predicted TSS after the clustering.

All hits that fell in the region (-2000, +2000) around the mapped TSS/Gene start were counted as correct, and all respective genes represented TP. All known genes missed in this way were counted as false negative. All hits that fell on the annotated part of the gene on the region (+2001, EndOfTheGene) were counted as FP hits. Other hits were not considered in counting TP and FP.

The main results are summarized in Tables 3–6. In these experiments, Dragon GSF, FirstEF, and Eponine have been used with their default parameter settings. Figure 3. shows the distribution of predictions from all four programs in the interval (-2000, +2000) relative to the start of gene transcripts determined based on DBTSS data and the annotation by Collins. The calculated values are taken in bins of 50 nt in length.

Table 3. Results on chromosome 4 based on annotation and sequence from NCBI GenBank

Program	TP	FP	Total # of TSSs	Total # of prediction	$S_e(\%)$	$S_p(\%)$	$CC(\%)$
PromPredictor	177	26	306	1407	57.84	87.19	71.02
DGSF	179	55	306	1349	58.50	76.50	66.89
FirstEF	221	170	306	3620	72.22	56.52	63.89
Eponine	120	38	306	2296	39.22	76.92	54.92

Table 4. Results on chromosome 21 based on annotation and sequence from NCBI GenBank

Program	TP	FP	Total # of TSSs	Total # of prediction	$S_e(\%)$	$S_p(\%)$	$CC(\%)$
PromPredictor	64	12	89	353	71.91	84.21	77.82
DGSF	62	17	89	383	69.66	78.48	73.94
FirstEF	74	108	89	1236	83.15	40.66	58.14
Eponine	46	16	89	816	51.69	74.19	61.93

Table 5. Results on Chromosome 22 Based on the annotation and sequence Used by Collins et al

Program	TP	FP	Total # of TSSs	Total # of prediction	$S_e(\%)$	$S_p(\%)$	$CC(\%)$
PromPredictor	64	12	89	353	71.91	84.21	77.82
DGSF	62	17	89	383	69.66	78.48	73.94
FirstEF	74	108	89	1236	83.15	40.66	58.14
Eponine	46	16	89	816	51.69	74.19	61.93

Table 6. Overall performance on Chromosomes 4, 21, and 22 with TSSs determined based on DBTSS data and the annotation by Collins et al

Program	TP	FP	Total # of TSSs	Total # of prediction	$S_e(\%)$	$S_p(\%)$	$CC(\%)$
PromPredictor	64	12	89	353	71.91	84.21	77.82
DGSF	62	17	89	383	69.66	78.48	73.94
FirstEF	74	108	89	1236	83.15	40.66	58.14
Eponine	46	16	89	816	51.69	74.19	61.93

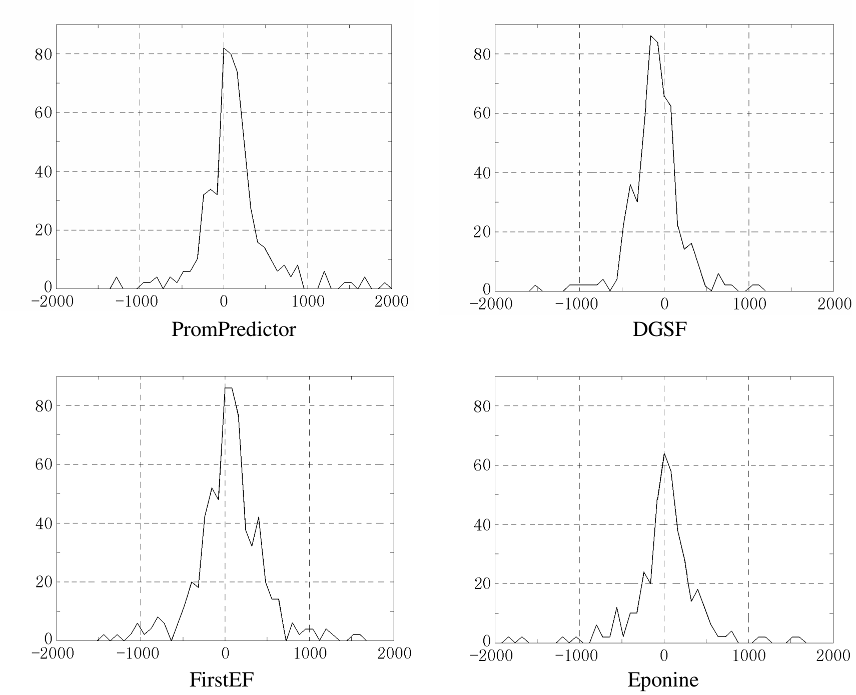


Fig. 3. Distributions of predictions from all four programs in the interval (-2000, +2000) relative to the start of gene transcripts determined based on DBTSS data and the annotation by Collins et al. The calculated values presented on these graphs are taken in bins of 50 nt in length

4 Conclusion

Promoter recognition is crucial for location transcription start sites in human genome. The secret of promoter function lies in the combination of several promoter elements that need to cooperate in transcriptional activation, while none of them can achieve alone. It is necessary to compile several individual weak signals into a composite signal which then indicates a potential promoter. PromPredictor is designed to capture these core elements as well as weak signals. It is based on the statistical concept of pentamer distributions in specific functional regions of DNA and selected the most significant pentamer vocabularies from training sequences by an unsupervised learning technique, in addition to CpG islands features.

PromPredictor yielded good results with Bajic's evaluation scheme. The result of PromPredictor predictions on chromosomes 4 and 21 was better than other three systems, while on chromosomes 22 Dragon GSF was better than PromPredictor. We think the reason is the difference of G + C content of the three chromosomes. Chromosome 22 is the second most G + C rich human chromosome (G + C content of ~48%). Chromosome 21 has a G + C content of ~41%, which is approximates the average for the human genome, and chromosome 4 is the most GC-poor human chromosome (G + C content of ~38%). Dragon GSF uses prediction of CpG islands as one of the global signals, so it is suitable for the analysis and discovery of promoters of those genes that are associated with CpG islands.

All systems use the concept of CpG islands, but the method is different. PromPredictor uses only two parameters (G + C content and ratio of expected to observed CG dinucleotides) in a sliding window, together with hundreds of pentamer distribution features, so the difference of G + C content has less influence on PromPredictor than on other three systems. This can be seen from Table 3. PromPredictor achieved a sensitivity = 57.84% and specificity = 87.19% on chromosome 4.

The overall performance of PromPredictor on the three human chromosomes shows our novel method is promising for modeling biological systems in general, which does not require any specific knowledge about a particular promoter to make a prediction and thus has a big advantage especially when nothing is known about the promoter to be predicted.

References

1. Lander, E.S., et al. Initial sequencing and analysis of the human genome. Nature, 409:860-921, 2001.
2. Venter, J.C., et al. The sequence of the human genome. Science, 291:1304-1351, 2001.
3. Lander, E.S. The new genomics: global views of biology. Science, 274:536-539, 1996.
4. Fickett, J.W., Hatzigeorgiou, A.G., 1997. Eukaryotic promoter recognition. Genome Res., 7:861-878.
5. Ohler, U., Niemann, H. Identification and analysis of eukaryotic promoters: recent computational approaches. TRENDS Genet., 17:56-60, 2001.
6. Scherf, M., Klingenhoff, A., Werner, T. Highly specific localization of promoter regions in large genomic sequences by PromoterInspector: a novel context analysis approach. J. Mol. Biol., 297:599-606, 2000.
7. Ioshikhes, I.P., Zhang, M.Q. Large-scale human promoter mapping using CpG islands. Nature Genetics, 26:61-63, 2000.
8. Davuluri, R.V., Grosse, I., Zhang, M.Q. Computational identification of promoters and first exons in the human genome. Nature Genetics, 29:412-417, 2001.
9. Hannenhalli, S., Levy, S. Promoter prediction in the human genome. Bioinformatics, 17:90-96, 2001.
10. Ponger, L., Mouchiroud, D. CpGProD: identifying CpG islands associated with transcription start sites in large genomic mammalian sequences. Bioinformatics, 18:631-633, 2001.

11. Bajic, V.B., et al. Dragon Promoter Finder: recognition of vertebrate RNA Polymerase II promoters. Bioinformatics, 18:198-199, 2002.
12. Bajic, V.B., et al. Computer model for recognition of functional transcription start sites in RNA polymerase II promoter of vertebrates. Journal of Molecular Graphic and Modeling, 21:323-332, 2003.
13. Bajic, V.B., Seah, S.H. Dragon Gene Start Finder: an advanced system for finding approximate locations of the start of gene transcriptional units. Genome Res., 13:1923-1929, 2003.
14. R.X., Liu., David, J. Consensus promoter identification in the human genome utilizing expressed gene markers and gene modeling. Genome Res., 3:462-469, 2002.
15. Claverie, J.M., Sauvaget, I., Bougueleret, L. K-tuple frequency analysis from intron/exon discrimination to Tcell epitope mapping. Methods Enzimol., 183:237-252, 1990.
16. Shago, M., Giguere, V. Isolation of a novel retinoic acid-responsive gene by selection of genomic fragments derived from CpG-island enriched DNA. Mol. Cell Biol., 16:4337-4348, 1996.
17. Bohjanen, P.R., Liu, Y., GarciaBlanco, M.A. TAR RNA decoys inhibit Tat-activated HIV-1 transcription after preinitiation complex formation. Nucleic Acids Res., 25:4481-4486, 1997.
18. Wang, W.D., Chi ,T.H., Xue, Y.T., Zhou, S,., Kuo, A. Architectural DNA binding by a high-mobility-group/kinesin-like subunit in mammalian SWI/SNF-related complexes. Proc Natl Acad Sci USA, 95:492-498. 1998
19. Bell, P.J.L., Higgins, V.J., Dawes, I.W., Bissinger P.H. Tandemly repeated 147 bp elements cause structural and functional variation in divergent MAL promoters of Saccharomyces cerevisiae. Yeast, 13:1135-1144, 1997.
20. Chuzhanova, N.A., Jones, A. J., Margetts, S. Feature selection for genetic sequence Classification. Bioinformatics, 14:139-143, 1998.
21. Dash M., Liu H. Feature selection for classification. Intelligent Data Analysis, 3:1-6, 1997.
22. Bassat M.B. Use of distance measures, Information measures and error bounds in feature evaluation. Classification,Pattern Recognition and Reduction of Dimensionality:Handbook of Statistics, Volume 2, Krishnaiah P.R., Kanal L.N. Editors, North-Holland Publishing Company,Amsterdam, 773-791, 1982.
23. Solovyev V.V., Makarova K.S. A novel method of protein sequence classification based on oligopeptide frequency analysis and its application to search for functional sites and to domain localization. Computer Applications in the Biosciences, 9(1):17-24, 1993.
24. Bird, A.P., et al. Non-methylated CpG-rich islands at the human α-globin locus: Implications for evolution of the α-globin pseudogene. EMBO J., 6:999-1004, 1986.
25. Gardiner-Garden, M., Frommer, M. CpG islands in vertebrate genomes. J. Mol. Biol., 196:261-282, 1987.
26. Larsen, F., Gundersen, G., Lopez, R., Prydz, H. CpG islands as gene markers in the human genome. Genomics, 13:1095-1107, 1992.
27. Cross, S.H., Bird, A.P. CpG islands and genes. Curr. Opin.Genet., Dev. 5:309-314, 1995.
28. Cross, S.H., Clark, V.H., Bird, A.P. Isolation of CpG islands from large genomic clones. Nucleic Acids Res., 27:2099-2107, 1999.
29. Pedersen, A.G., Baldi, P., Chauvin, Y., Brunak, S. The biology of eukaryotic promoter prediction—A review. Comput. Chem., 23:191-207, 1999.
30. Cavin, PeÂrier, R., Junier, T., Bucher, P. The Eukaryotic Promoter Database EPD. Nucl. Acids Res., 26:353-357, 1998.

31. Saxonov, S., Daizadeh, I., Fedorov, A., and Gilbert, W. EID: The Exon-Intron Database—An exhaustive database of protein-coding intron-containing genes. Nucleic Acids Res. 28: 185–190, 2000.
32. Pesole, G., et al. UTRdb and UTRsite: specialized database of sequences and functional elements of 5' and 3' untranslated regions of eukaryotic mRNAs.Update 2002. Nucl. Acids Res., 30:335-340, 2002.
33. Grillo, G., Attimonelli, M., Liuni, S., Pesole G. CLEANUP: a fast computer program for removing redundancies from nucleotide sequence databases. Comput. Applic. Biosci., 12:1-8, 1996.
34. Powell, M.J.D. Restart procedures for the conjugate gradient method. Mathematical Programming, 12:241-254, 1977.
35. Battiti, R. First and second order methods for learning: Between steepest descent and Newton's method. Neural Computation, 4(2):141-166, 1992.
36. Riedmiller, M., Braun H. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. Proceedings of the IEEE International Conference on Neural Networks, San Francisco, 1993.
37. Hagan, M.T., Menhaj M. Training feedforward networks with the Marquardt algorithm. IEEE Transactions on Neural Networks, 5(6):989-993, 1994.
38. Down, T.A., Hubbard, T.J. Computational detection and location of transcription start sites in mammalian genomic DNA. Genome Res. 12: 458–461, 2002.
39. Collins, J.E., Goward, M.E., Cole, C.G., et al. Reevaluating human gene annotation: A second-generation analysis of chromosome 22. Genome Res. 13: 27-36, 2003.
40. Bajic, V.B., Tan, S.L., Suzuki, Y., Sugano, S. Promoter prediction analysis on the whole human genome. Nature Biotechnology, 22(11):1467-73, 2004.

Robust Ensemble Learning for Cancer Diagnosis Based on Microarray Data Classification

Yonghong Peng

Department of Computing, University of Bradford,
West Yorkshire, BD7 1DP, U.K
y.h.peng@bradford.ac.uk

Abstract. DNA microarray technology has demonstrated to be an effective methodology for the diagnosis of cancers by means of microarray data classification. Although much research has been conducted during the recent years to apply machine learning techniques for microarray data classification, there are two important issues that prevent the use of conventional machine learning techniques, namely the limited availability of training samples and the existence of various uncertainties (e.g. biological variability and experiment variability). This paper presents a new ensemble machine learning approach to address these issues in order to achieve a robust microarray data classification. Ensemble learning combines a set of base classifiers as a committee to make appropriate decisions when classifying new data instances. In order to enhance the performance of the ensemble learning process, the approach presented includes a procedure to select optimal ensemble members that maximize the behavioural diversity. The proposed approach has been verified by three microarray datasets for cancer diagnosis. Experimental results have demonstrated that the classifier constructed by the proposed method outperforms not only the classifiers generated by the conventional machine learning techniques, but also the classifiers generated by two widely-used conventional Bagging and Boosting ensemble learning methods.

1 Introduction

Advances of DNA microarray technology allow recording the expression levels of thousands of genes simultaneously. Recently research has demonstrated convincingly that accurate cancer diagnosis can be achieved by performing microarray data classification, i.e. by constructing classifiers to compare the gene expression profile of a tissue of unknown cancer status to a database containing expression profiles from tissues of known cancer status.

A microarray dataset is represented by a $m \times n$ matrix $M=[e_{ij}]_{m \times n}$, where e_{ij} denotes the expression level of the i-th gene in the jth sample, and the m and n are respectively the numbers of genes and the number of samples. One row, $g_i=(e_{i1},e_{i2},\ldots,e_{in})$, represents the expression profile for the associated gene over various biological conditions (such as healthy and cancerous), while one column, $s_j=(e_{1j},e_{2j},\ldots,e_{mj})$, represents the gene expression profile over the associated samples/tissues.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 564–574, 2005.

Many machine learning techniques have been applied to microarray data classification and cancer diagnosis, including the decision tree [1,2], K-NN [3,4], Neural Networks [5, 6], and Support Vector Machines (SVMs) [7,8,9], and feature selection techniques [10,11]. Given a training microarray dataset, $T = \{(s_1, c_1),\dots, (s_n, c_n)\}$, in which $s_j \in R^m$ represents a gene expression vector of the j-th sample, and each sample is labelled with a class $c_j = \{-1, +1\}$ (j=1~n) representing normal and cancer conditions respectively, the task of microarray classification is to find a optimal classifier c: $R^m \rightarrow \{-1,+1\}$ that maximises the probability that $c(s_j) = c_j$ for j=1~n.

There are several intrinsic difficulties when apply the conventional machine learning techniques for microarray data classification: (1) the microarray data inherently contains a huge number of genes but a small number of samples (e.g. the leukaemia dataset [12] contains 6817 genes and 72 instances), which is known as the curse of dimensionality in machine learning; (2) the microarray data is presented with a variety of uncertainties. The process of microarray data gathering, which includes fabrication, hybridisation, image processing etc, always adds various sources of noise [13, 14]. As a result, there is a great need to develop general approaches and robust methods that are able to overcome the limitation of the small number of training samples and reduce the influence of uncertainties. The idea behind the proposed approach is to reduce the impact of the uncertainties and enhance the robustness of classification model by making as much use as possible of the information redundancy and functional overlapping of different partitions of genes (gene subsets).

The method presented in this paper is based on the idea of ensemble learning which seeks an optimal classification based on the combination of multiple classifiers. The existing ensemble learning approaches can be summarised as: 1) to use different learning algorithms, 2) to re-sample the original training dataset. The re-sampling of instances and sub-sampling of features are two widely-used re-sampling methods in the literature. In the first approach, a set of training datasets are generated by means of re-sampling from the original dataset with instance replacement, i.e. the re-sampled dataset contains the same number of instances as the original set and the same number of features. The output of the second re-sampling method is a set of new training sets associated with different feature subsets. The widely-used Bagging [15,16] and Boosting [17, 18] ensemble learning methods are based on the re-sampling of training instances. A few researcher [19, 20, 21] have investigated the ensemble learning method based on the feature sub-sampling. Facing the fact of having limited number of training instances but large number of genes, the approach presented in this paper is based on the sub-sampling of the features (genes). The idea behind this approach is to use the functional diversity of gene subsets to generate diverse base classifiers, and to apply the overlapping of gene functions to reduce the influence of the experimental noise and biological uncertainties.

This paper is organised as follows. Section 2 presents the proposed approach and Section 3 discusses the method of characterizing and sampling the genes. In the Section 4, the proposed method for the characterization of classifier behavior and the selection of appropriate base classifiers. Section 5 presents the experimental results to demonstrate the effectiveness of the proposed method, in which a discussion of the results is also presented. The conclusions are given in Section 6.

2 The Proposed Ensemble Learning Approach

Recent research has shown that a necessary and sufficient condition for an ensemble classifier to outperform its individual members is that the base classifiers should be accurate and diverse [22, 23]. Furthermore, several researchers have recently demonstrated that a compact ensemble committee constructed by optimally selecting base classifiers outperforms that without the selection of base classifiers [24, 25].

Classifier 5	+	+	+	+	+	○	○	○
Classifier 4	○	○	○	+	+	+	+	+
Classifier 3	+	+	+	○	○	○	+	+
Classifier 2	+	○	○	○	+	+	+	+
Classifier 1	○	○	○	+	+	+	+	+
Samples	1	2	3	4	5	6	7	8

+ Correctly classified ○ Incorrectly classified

Fig. 1. Classification behaviour of base classifiers

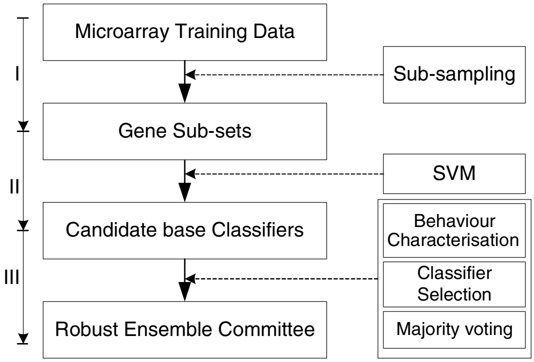


Fig. 2. The proposed ensemble learning approach

The example shown in Fig.1 illustrates that for a group of classifiers with same classification accuracy (62.5%); different ensemble committees would ultimately produce different accuracies. For example, the ensemble of classifiers 1, 2 and 3 produce accuracy of 62.5% (no accuracy improvement), and the ensemble of 1, 2 and 4 or all classifiers achieves an accuracy of 75%, while the ensemble of 1, 3 and 5 produces the best accuracy (87.5%). This example demonstrates that selecting appropriate base classifiers is essential in order for achieving an improved robust classification. This paper therefore proposes a three-step based ensemble learning approach, as shown in Fig.2.

Step 1. Sub-sampling of genes. The gene sub-sampling is performed in terms of each gene's significance in distinguishing the samples, as discussed in Section 3.

Step 2. Construction of candidate base classifiers. This step applies machine learning methods to generate a pool of candidate classifiers. The inputs for each classifier correspond to the associated gene subset. The particular machine learning algorithm used in this study is the SVM, which is briefly discussed in Section 4.

Step 3. Construction of a robust classification committee, which consists of three sub-steps: characterisation of the behaviour of candidate classifiers, selection of a subset of candidate classifiers, and the ensemble of the selected classifiers. The classifiers having high behavioural diversity (they mostly disagree with each other) and being accurate will be selected as the committee members. The majority voting method is then employed to make the final decision for the ensemble classification.

3 Characterisation and Sampling of Genes

In the approach presented in Fig.2, the base classifiers are constructed based on the sub-sampled genes from the original gene set. In this study, the genes with high significance levels would have more chances to be selected in order to produce accurate and diverse candidate base classifiers.

Gene Characterisation. A gene is characterised by its discriminatory power to distinguish the samples under different biological conditions. For a microarray dataset represented by a $m \times n$ matrix $M=[e_{ij}]_{m \times n}$, one effective method [12] to characterize the significance of a gene (g_i) is measured by

$$p_i = \left| \frac{\mu_{i+} - \mu_{i-}}{\sigma_{i+} + \sigma_{i-}} \right| \tag{1}$$

where $[\mu_{i+}, \sigma_{i+}]$ and $[\mu_{i-}, \sigma_{i-}]$ are the means and standard deviation of the expression levels of samples in class '+1' and '-1'. A large value of p_i indicates that the associated gene is more differentially expressed across two classes.

Gene Sub-sampling. The Monte Carlo method is employed in sampling the gene, as detailed in Fig.3. The sampling probabilities of genes are determined by their significance indicated by p_i. By repeating K times the sub-sampling procedures, totally K gene subsets are then produced, $G^S = \{G_1, G_2, \ldots G_K\}$.

```
Inputs: Gene set G={$g_1$,... $g_n$}, the associated significance levels S={$s_1$,... $s_n$}, and the
  number of genes to be selected $n_G$. Outputs: Selected gene set $G_k$
   1). Normalise the significance of genes in G: $\hat{p}_i = p_i \Big/ \sum_{i=1}^{n} p_i$
   2). Calculate the cumulative significance distribution function: $cd_i = \sum_{j=1}^{i} \hat{p}_j$
   3). Let $|G_k| = 0$ ($G_k$ is initially empty) and repeat the following steps
       a) Generate a pseudo-random number $\xi \in [0,1]$.
       b) Retrieve the gene number $i$ such that $cd_{i-1} < \xi \le cd_i$.
       c) Check if $|G_k| > n_G$, then terminate and return $G_k$, otherwise go to d).
       d) Check if $i \notin G_k$, then let $G_k = G_k \cup \{i\}$ and go to a), otherwise go to a).
```

Fig. 3. Algorithm for Gene Sub-sampling

4 Ensemble Learning Based on SVM

4.1 SVM Classifier

The Support Vector Machine (SVM) is employed in this research. Unlike most of the modelling methods attempting to minimise an objective function (such as the mean square error) for the whole training instances, SVM finds the hyperplanes that produce the largest separation between values for the decision function for the instances located at the borderline between two classes [26].

Given a labelled microarray data $M=\{s_i, c_i\}$ where $s_i = (e_{1i}, e_{2i}, \ldots, e_{mi}),$, $i = 1, 2, \ldots, n$, the target of SVM learning is to construct a decision function $f(s)$: R^m → $\{+1,-1\}$, such that for each s_i, the function yields $f(s_i)>0$ for $c_i=+1$, and $f(s_i)<0$ for $c_i=-1$. A SVM employs a linear $f(s) = W^T s + b$ or nonlinear decision function $f(s) = W^T \phi(s) + b$. The function $f(x)$ is determined by minimizing $J(w,\xi) = \frac{1}{2}\|W\|^2 + C\sum_{i=1}^{l} \xi_i$ subject to $c_i(W^T c_i + b) \ge 1 - \xi_i$ (linear) or $c_i(W^T \phi(c_i) + b) \ge 1 - \xi_i$ (nonlinear), where $C > 0$ is a regularisation parameter and $\xi_i \ge 0$ (i=1,2,...,l) are slack parameters. The function $K(u,v) = \phi^T(u)\phi(v)$ is usually called the kernel. Three typical kernel functions used in SVM classification are the Linear $K(u,v) = u \cdot v$, Polynomial $K(u,v) = (u \cdot v + 1)^\gamma$ and Gaussian function $K(u,v) = \exp\left(\frac{-\|u-v\|^2}{\sigma^2}\right)$.

4.2 Construction of Robust Ensemble Classifiers

As discussed above, when seeking a robust ensemble classifier, a set of base classifiers should be ideally selected such that both diversity and accuracy are maximised. Several researchers have applied the feature selection methods for the selection of compact ensemble committee and have demonstrated their effectiveness [24,25]. The method proposed in this paper first characterises the behaviour of the candidate base classifiers, and then selects the appropriate base classifiers in terms of their classification behaviour, as shown in Fig.4.

Behaviour Characterisation of Base Learning. The leave-one-out performance is applied to characterise the behaviour of base classifiers (and the associated gene subsets). Given a training dataset with n_0 samples $V = (s_1, s_2, \ldots s_{n_0})$, by applying the leave-one-out evaluation method, one classifier is characterised by a vector

$$v_i = (\hat{y}_{i1}, \hat{y}_{i2}, \ldots \hat{y}_{in_0}) \tag{2}$$

where $\hat{y}_{ij} = c_i(s_j)$ is the prediction of classifier $c_i(\cdot)$ for sample s_j, and the classifier $c_i(\cdot)$ is constructed by the remaining (n_0-1) instances. This vector v_i is known as the characteristic vector of classifier $c_i(\cdot)$ and $E=\{v_i, i=1{\sim}K\}$ forms the behaviour matrix characterising the candidate base classifiers.

Inputs: A pool of gene subsets generated by the hybrid sub-sampling algorithm.
K, the number of subsets of genes.
K_c, the number of ensemble members expected.
Outputs: A classification committee.

1). By using the *K* gene subsets and a training dataset having n_0 instances, a behaviour matrix $E = [\hat{y}_{ij}]_{n_0 \times k}$ is generated using the leave-one-out method, where $\hat{y}_{ij} = c_i(s_j)$ is the predicated class for instance s_j and $v_i = (\hat{y}_{i1}, \hat{y}_{i2}, \ldots \hat{y}_{in_0})$ is the characteristic vector for classifier $c_i(\cdot)$.

2. By applying the k-means algorithm on the behaviour matrix, the candidate classifiers are distributed into K_c clusters.

3. Rank the classifiers in each cluster according to their misclassification rates

$$e(v_i, Y) = \frac{1}{n_0} \sum_{j=1}^{n_0} \delta(\hat{y}_{ij}, y_j)$$

where $Y = (y_1, y_2, \ldots, y_i)$ and y_i are the true classes for the *j*-th sample, $\delta(x,y)$ is calculated by $\delta(x, y) = \begin{cases} 0 & x = y \\ 1 & x \neq y \end{cases}$

4. Select classifiers with minimal error rate in different clusters to construct the classification committee for final classification.

Fig. 4. The Proposed Algorithm for Robust Ensemble Machine Learning

Selection of Base Classifiers for Ensemble. Given the behaviour matrix *E*, one can apply a feature selection method to select the base classifiers. In this paper, the selection of base classifiers is performed based on clustering candidate classifiers. The k-means algorithm is used to group the base classifiers that have similar performances, and distribute the candidate base classifiers, which disagree with each other, into different clusters. By selecting the most representative classifiers from different clusters, it is possible to increase the level of disagreement of the ensemble classifier, and improve indirectly the robustness of the ensemble classifier. The most representative classifier is selected in terms of the distances between the associated classifiers with all other classifiers in the same cluster, and the associated error rate (d/err), where *d* is the average distance between one classifier and all others, the *err* is the error rate of the associated classifier.

Ensemble Method. In this paper, the majority vote method is employed to perform the final decision based on the outputs of base classifiers. Given k_c classifiers $c_i(\cdot)$: $R^n \rightarrow \{-1,+1\}$, for a new instance *s*, each classifier predicated class $c_i(\cdot) \in \{-1,+1\}$ $i=1 \sim k_c$ and the majority vote method generates a final classification by

$$c_{en}(s) = sign\left(\sum_{i=1}^{K_c} \omega_i c_i(s)\right) \tag{3}$$

Where $\omega_i \in [0,1]$ is the weight for classifier $c_i(\cdot)$, which reflects its significance in classification, and is defined in terms of the accuracy of classifier $c_i(\cdot)$:

$$\omega_i = \frac{1-e(v_i,Y)}{\sum_{i=1}^{K_c}\left(1-e(v_i,Y)\right)} \tag{4}$$

5 Experiments and Results

5.1 The Datasets

The proposed method has been evaluated by three publicly available microarray datasets for cancer diagnosis, which are the breast cancer microarray dataset [27], the colon tumour dataset [28], and the prostate cancer dataset [29]. The characteristics of each data is summarized in Table 1. In this study, the transformed microarray data in c4.5 format were obtained from the Kent Ridge Bio-medical Data Set Repository[1].

Table 1. Microarray Datasets

Dataset	Number of Genes	Number of Samples
Breast	24481	97 (46+, 51-)*
Colon	2000	62 (40+, 22-)*
Prostate	12600	102 (52+, 50-)*

* n_+ n_- represent the number of samples of class '+1' and '-1'.

5.2 Performance Evaluation Method

The leave-one-out validation method is employed in this study to evaluate the performance of classification, which, compared to the k-fold cross-validation method, is more appropriate for cases having a limited number of data samples. Given a dataset with *N* data instances, (*N*-1) data instances are used to construct a classifier and then apply the reminding one to test the classifier. By repeating this process of successively using each data instances (x_i) as the testing data instance, a total of *N* prediction e_i=$c(x_i)$ (i=1~N) are obtained, and the performance of classifier is assessed by the average classification rate.

$$Acc_r = \frac{1}{N}\sum_{i=1}^{N}\mu(e_i,y_i) \tag{5}$$

where y_i is the true class label for instance s_i, and $\mu(x,y)$ is calculated by

$$\mu(x,y) = \begin{cases} 0 & x \neq y \\ 1 & x = y \end{cases} \tag{6}$$

5.3 Results

Four sets of experiments have been performed to evaluate the effectiveness of the proposed method. The OSU SVM Classifier Matlab Toolbox+ is employed to generate the candidate base classifiers.

[1] http://sdmc.lit.org.sg/GEDatasets/Datasets.html

+ which is available from http://www.ece.osu.edu/~maj/osu_svm/

The first experiment is intended to compare the performances of the proposed ensemble learning method (denoted by enSVM) to that of the single SVM classifier (denoted by SVM), the Bagging ensemble learning (denoted as Bagging) and Boosting ensemble learning (denoted as Boosting). In this experiment, the top 50 genes (the 50 most significant genes) are used in single SVM, Bagging and Boosting, and the sub-sampled 50 genes are used in enSVM. The number of candidate base classifiers is set to be K = 200, from which K_c = 25 base classifiers are selected to become the classification committee.

Table 2 and Fig.5 report the accuracy of these methods for the associated datasets. These results clearly show that the enSVM method not only performs better than signal SVM classifier but also outperforms these two widely-used ensemble methods (Bagging and Boosting). Furthermore, it has been shown that the Bagging learning method performs better than Boosting method, which has also been recognised by other researches (e.g. [2]). One explanation, based on the experiments performed in this study, is that given the fact of limited training instances in the dataset, the boosting method always terminates before the given number of classifier has been obtained and the base classifiers generated in the latter stages of boosting focuses on learning from only a few instances.

Table 2. Classification Accuracies (%)

	Breast	Colon	Prostate
SVM	75.3	80.7	92.2
Bagging	78.4	82.3	90.2
Boosting	78.4	80.7	89.2
enSVM	81.4	88.7	95.1

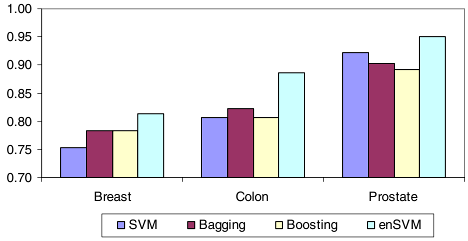
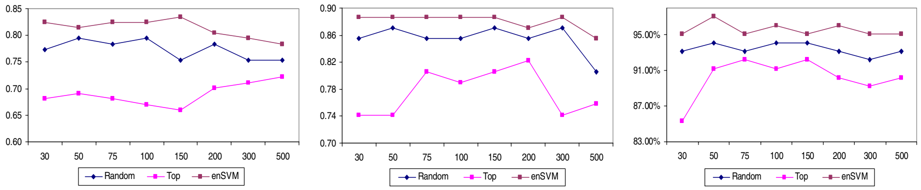


Fig. 5. Classification Accuracies

The second experiment is performed to evaluate the effectiveness of the proposed method for base classifier selection. In this experiment, the number of base classifiers is 25 which are selected from 200 candidate classifiers, and the number of genes is varied. The performance of the proposed enSVM method (based on clustering analysis of candidate classifiers) is compared with the results of 1) randomly selecting base classifiers (denoted by Random), and 2) selecting the best accurate classifiers (denoted by Top). The results for these three methods are shown in Tables 3 and Fig. 6. From these experimental results, it is clearly shown that the proposed method has successfully improved the accuracy for all these three datasets under varying numbers of genes. One interesting observation is that the selection of top base classifiers produces inferior classification accuracy than the randomly selected base classifiers, which implies that the selection of base classifier is important in robust ensemble learning techniques.

Table 3. Accuracies of using different base classifier selection methods (%)

		Number of Genes							
Dataset	Method	30	50	75	100	150	200	300	500
Breast	Random	77.3	79.4	78.3	79.4	75.3	78.3	75.3	75.3
	Top	68.0	69.1	68.0	67.0	66.0	70.1	71.1	72.2
	enSVM	82.5	81.4	82.5	82.5	83.5	80.4	79.4	78.3
Colon	Random	85.5	87.1	85.5	85.5	87.1	85.5	87.1	80.6
	Top	74.2	74.2	80.6	79.0	80.6	82.3	74.2	75.8
	enSVM	88.7	88.7	88.7	88.7	88.7	87.1	88.7	85.5
Prostate	Random	93.1	94.1	93.1	94.1	94.1	93.1	92.2	93.1
	Top	85.3	91.2	92.2	91.2	92.2	90.2	89.2	90.2
	enSVM	95.1	97.1	95.1	96.1	95.1	96.1	95.1	95.1



(a) Breast Cancer Data (b) Colon tumour Data (c) Prostate Cancer Data

Fig. 6. Accuracies of using different base classifier selection methods

5.4 Discussion

By comparing with the performance of single classifier and two widely-used instance re-sampling based ensemble learning methods (Bagging and Boosting), the experimental results illustrate that consistent and significant improvements of accuracy have been produced by the method proposed.

For comparison with existing microarray data classification methods, the leukaemia dataset has been used to verify the performance of the proposed method. This dataset was originally provided in [12], and contains the expression levels of 6817 genes of 72 patients, among which, 47 patients suffer from the Acute Lymphoblastic Leukaemia (ALL) and 25 patients suffer from the Acute Myeloid Leukaemia (AML). On the original test dataset comprising 34 instances, the enSVM method assigns the correct label for 33 instances. This can be directly compared to the results obtained original in [12], in which 29 instances were correctly classified, and the results obtained in [9] applying SVM classifier, in which improved results are reported from 30 to 32 correct classifications.

The colon dataset classification has been investigated in [30], in which five different ensemble machine learning methods had been compared. Compared with the best accuracy obtained in [30] for the colon tumour dataset (which is a 14.52% classification error rate) the enSVM improves the accuracy by reducing the misclassification rates to between 11.3% and 12.9% when using less than 300 genes.

6 Conclusions

Microarray data classification has been shown to be an effective methodology for cancer diagnosis. The challenges to traditional machine learning techniques are from the limited availability of training instances, and the existence of various sources of uncertainties.

This study aims at developing a generic machine learning approach that can address the issues that exist in microarray data analysis and produce a robust method of classification for microarray data. The principal idea behind this study is to involve multiple classifiers constructed by using different subsets of genes so as to fuse the information from diverse gene subsets. The approach presented in this paper consists of three basic steps, namely gene sub-sampling, the generation of candidate base classifiers, and the construction of a robust ensemble classifier. From the experimental results we can draw the following conclusions:

1) The gene sub-sampling based ensemble learning methods provide more effective approach to implement robust classification for microarray data. The experimental results show that the gene sub-sampling based ensemble learning always outperforms the instance re-sampling ensemble learning method such as Bagging and Boosting methods.
2) For the development of robust ensemble classification, it is essential to select appropriate base classifiers from the available candidate classifiers. Experimental results also illustrate the effectiveness of the method presented in this paper in the selection of the base classifiers with due regard to the behavioural characterisation of base classifiers.
3) The experimental results demonstrated that the proposed method outperforms the signal classifiers and the conventional ensemble learning methods (Bagging and Boosting).

The ensemble learning approach proposed in this paper is generic, which is less sensitive to the selection of genes and can be applied on different base classifiers. These characteristics of the proposed method suggest that the proposed method has enormous potential for the developments of generic platform for microarray data classification.

References

1. Berrar, D., Sturgeon, B., I. Bradbury, Dubitzky, W.: Microarray data integration and machine learning techniques for lung cancer survival prediction, *CAMDA-2003* (2003).
2. Tan, A. C., Gilbert, D.: Ensemble machine learning on gene expression data for cancer classification, *Appl Bioinformatics*. 2(3 Suppl), 75-83 (2003).
3. Li, L., Weinberg, C.R. et al.: Gene selection for sample classification based on gene expression data: study of sensitivity to choice of parameters of the GA/KNN method, *Bioinformatics*, 17, (2001) 1131-1142.
4. Cho S. B., Won, H. H.: Machine learning in DNA microarray analysis for cancer classification, *Proceedings of the First Asia-Pacific Bioinformatics Conference* (2003).
5. Cho S. B., Won, H. H.: Neural network classifiers and gene selection methods for Microarray Data on Human Lung Adenocarcinoma, *CAMDA 2003 Conference* (2003).
6. Friedman, N., Linial, M., et al.: Using Bayesian networks to analyze expression data. *Journal of Computational Biology* 7 (2000) 601-620.

7. Brown, M.P., Grundy, W.N., et al.: Knowledge-based analysis of microarray gene expression data by using support vector machines, *Proc Natl Acad Sci USA* 97 (2000) 262-7.
8. Mukherjee, S. Tamayo, P., Mesirov, J.P., et al.: Support vector machine classification of microarray data, *Technical Report 182*, AI Memo 1676, CBCL (1999).
9. Furey, T.S., Cristianini. N., et al: Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics*, 16 (2000) 906-914.
10. Guyon, I., Weston, J., Barnhill S. Vapnik, V.: Gene selection for cancer classification using support vector machines, *Machine Learning*, 46 (2002) 389-422.
11. R. Blanco, P. Larrañaga, I. Inza and B. Sierra, Gene selection for cancer classification using wrapper approaches, *International Journal of Pattern Recognition and Artificial Intelligence*, to appear (2004).
12. Golub, T. R., Slonim, D.K., Tamayo, P., et al.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, *Science* 286 (1999) 531-537.
13. Coombes, K. R., Highsmith, W.E. et al.: Identifying and quantifying sources of variation in microarray data using high-density cDNA membrane arrays, *J Comput Biol*. 9(4) (2002) 655-669.
14. Wang, X., Hessner, M.J., Wu, Y. et al.: Quantitative quality control in microarray experiments and the application in data filtering, normalization and false positive rate prediction, *Bioinformatics*. 19(11) (2003) 1341-7.
15. Bauer, E., Kohavi, R., An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36(1-2) (1999) 105-139.
16. Breiman, L.: Bagging Predictors, *Machine Learning*, 24(2) (1996)123-140.
17. 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) (1997) 119-139.
18. Schapire, R. E.: A brief introduction to boosting, *The 16th International Joint Conference on Artificial Intelligence*, 1401–1406 (1999).
19. Ho, T.K.: The Random Subspace Method for Constructing Decision Forests, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 20(8) (1998) 832-844.
20. Ho, T.K.: C4.5 Decision Forests, *Proceedings of the 14th International Conference on Pattern Recognition,* Brisbane, Australia, 545-549 (1998).
21. Cao, J., Ahmadi, M. , Shridhar, M.: Recognition of handwritten numerals with multiple feature and multistage classifier, *Pattern Recognition*, 28(2) (1995) 153-160.
22. Hansen L. K., Salamon, P. : Neural network ensembles, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 12(10) (1990) 993-1001.
23. Dietterich, T. G.: Ensemble methods in machine learning, *First International Workshop on Multiple Classifier Systems*, 1-15 (2000).
24. Xing, E., Jordan, M. Karp, R.: Feature selection for high-dimensional genomic microarray data, *Proceedings of the Eighteenth International Conference on Machine Learning*, 601–608 (2001).
25. Yu, L., Liu, H.: Redundancy based feature selection for microarray data, *Proceedings of the Tenth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 737-742 (2004).
26. Cristianini, N., Shawe-Taylor, J.: An introduction to support vector machines, *Cambridge University Press* (2000).
27. van't Veer L. J., Dai H. et al., Gene expression profiling predicts clinical outcome of breast cancer, *Nature* 415 (2002) 530-536.
28. Alon, U., Barkai, N., Notterman, D.A., Gish, K., et al.: Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Proc Natl Acad Sci USA*, 96(12) (1999) 6745-6750.
29. Singh, D., Febbo, P.G., et al.: Gene expression correlates of clinical prostate cancer behaviour, *Cancer Cell* 1 (2002) 203-209.
30. Dettling M., Bühlmann, P.: Boosting for tumor classification with gene expression data, *Bioinformatics*, 19(9) (2003) 1061-1069.

A Comprehensive Benchmark of the Artificial Immune Recognition System (AIRS)

Lingjun Meng[a], Peter van der Putten[b,1], and Haiyang Wang[a]

[a] Network Center, Shandong University, P.R. China
{mlj, why}@sdu.edu.cn
[b] Leiden Institute of Advanced Computer Science, Leiden University, The Netherlands
putten@liacs.nl

Abstract. Artificial Immune Systems are a new class of algorithms inspired by how the immune system recognizes, attacks and remembers intruders. This is a fascinating idea, but to be accepted for mainstream data mining applications, extensive benchmarking is needed to demonstrate the reliability and accuracy of these algorithms. In our research we focus on the AIRS classification algorithm. It has been claimed previously that AIRS consistently outperforms other algorithms. However, in these papers AIRS was compared to benchmark results from literature. To ensure consistent conditions we carried out benchmark tests on all algorithms using exactly the same set up. Our findings show that AIRS is a stable and robust classifier that produces around average results. This contrasts with earlier claims but shows AIRS is mature enough to be used for mainstream data mining.

1 Introduction

In recent years there has been a rapid growth in the interest in Artificial Immune Systems for applications in data mining and computational intelligence [4]. The immune system is sometimes called the 'second brain' for its abilities to recognize new intruders and remember past occurrences. Simulating the immune system or translating immune system mechanisms into machine learning is an interesting topic on its own. However, to be accepted as a candidate algorithm for data mining applications, the source of inspiration for these algorithms is not really an issue of interest. Rather empirical evidence is needed that these algorithms produce high quality, reliable results over a wide variety of problems compared to a range of other approaches, without the need of expert fine-tuning.

In this paper we present such benchmarking results. Given that we are interested in applicability of artificial immune systems for real world data mining, and that classification is one of the most important mining tasks, we focus on the Artificial Immune Recognition System (AIRS) algorithm. AIRS was introduced in 2001 as one of the first immune systems approaches to classification [14] and seemed to perform reasonably well on various classification problems. Until then, several papers have been published dealing with AIRS benchmarking [5,6,10,15]. However, in our

[1] Corresponding author.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 575 – 582, 2005.

opinion these approaches were relatively limited, given that comparisons were made on a small number of data sets and algorithms, and that the benchmark results were sourced from literature rather than produced under exactly the same conditions as for the AIRS algorithm.

In contrast to the previous work mentioned, all our experiments have been run from scratch, to guarantee consistent experimental conditions. This includes applying AIRS on a wide range of representative real-world datasets with large differences in number of instances, attributes and classes.

The remainder of this paper is organized as follows. Section 2 provides some background on the AIRS algorithm. Section 3 reports on the set up of the benchmarking experiments. Section 4 discusses the results and identifies directions for future benchmarking research and section 5 concludes the paper.

2 The Artificial Immune Recognition System (AIRS)

The recognition and learning capabilities of the natural immune system have been an inspiration for researchers developing algorithms for a wide range of applications. This section introduces some basic immune system concepts and provides the history and background behind the AIRS algorithm for classification.

2.1 Natural Immune Systems

The natural immune system offers two lines of defense, the innate and adaptive immune system. The innate immune system consists of cells that can neutralize a predefined set of attackers, or ‘antigens’, without requiring previous exposure to them. The antigen can be an intruder or part of cells or molecules of the organism itself. In addition, higher animals like vertebrates possess an adaptive immune system that can learn to recognize, eliminate and remember specific new antigens. This is accomplished by a form of natural selection. The bone marrow and thymus continuously produce lymphocytes and each of these cells can counteract a specific type of antigen. Now if for example a B-cell lymphocyte encounters an antigen it codes for, it will produce antibody molecules that neutralize the antigen and in addition a large number of cloned B-cells are produced that code for the same antigen (‘clonal expansion’ or ‘clonal selection’). The immediate reaction of the innate and adaptive immune system cells is called the primary immune response. A selection of the activated lymphocytes is turned into sleeper memory cells that can be activated again if a new intrusion occurs of the same antigen, resulting in a quicker response. This is called the secondary immune response [4].

2.2 Artifical Immune Systems

Natural immune systems have inspired researchers to develop algorithms that exhibit adaptivity, associative memory, self – non self discrimination and other aspects of immune systems. These artificial immune system algorithms (also known as immunocomputing algorithms) have been applied to a wide range of problems such as biological modeling, computer network security & virus detection, robot navigation, job shop scheduling, clustering and classification [4].

The Artificial Immune System algorithm (AIRS) can be applied to classification problems, which is a very common real world data mining task. Most other artificial immune system research concerns unsupervised learning and clustering. The only other attempt to use immune systems for supervised learning is the work of Carter [2]. The AIRS design refers to many immune system metaphors including resource competition, clonal selection, affinity maturation, memory cell retention, and so on. AIRS builds on the concept of resource limited clustering [3,13].

According to the introductory paper, AIRS seems to perform well on various classification and machine learning problems [14]. Watkins claimed "the performance of AIRS is comparable, and in some cases superior, to the performance of other highly-regarded supervised learning techniques for these benchmarks". Later on, Goodman, Boggess, and Watkins investigated the "source of power for AIRS" and its performance on multiple-class problems. They claim "AIRS is competitive with the top five to eight classifiers out of 10-30 best classifiers on those problems", "it was surprisingly successful as a general purpose classifier" and it "performed consistently strong across large scope of classification problems" [5,6].

2.3 AIRS: The Algorithm

From a data mining point of view, AIRS is a cluster-based approach to classification. It first learns the structure of the input space by mapping a codebook of cluster centers to it and then uses k-nearest neighbor on the cluster centers for classification. The attractive point of AIRS is its supervised procedure for discovering both the optimal number and position of the cluster centers.

In AIRS, there are two different populations, the Artificial Recognition Balls (ARBs) and the memory cells. If a training antigen is presented, ARBs (lymphocytes) matching the antigen are activated and awarded more resources. Through this process of stimulation, mutation and selection a candidate memory cell is selected which is inserted to the memory cell pool if it contributes enough information. This process is repeated for all training instances and finally classification takes place by performing a nearest neighbor search on the memory cell population.

To describe the AIRS algorithm in detail, let us assume we have a training data set X containing n labelled instances $ag_i = \{x_i, t_i\} \in \mathbb{R}^d \bullet \mathbb{Z}$ with x_i an input with d attributes and t_i a one dimensional target class ($i=1,2,\ldots,n$). The algorithm goes through the following steps [15]:

1. Initialization

First all the data items will be normalized so that the affinity of every two training instances ag_i and ag_j is in the range [0,1]. In AIRS, the affinity is usually represented by Euclidean distance over the attributes. We assume the set MC as the memory cell pool containing m memory cells: $MC=\{mc_1,mc_2,\ldots,mc_m\}$, and set AB as the ARB-population containing r ARBs: $AB=\{ab_1,ab_2,\ldots,ab_r\}$, with $mc_j = \{x_j^{mc}, t_j^{mc}\}$, ($j=1,2,\ldots,m$); $ab_k = \{x_k^{ab}, t_k^{ab}\}$, ($k=1,2,\ldots,r$). Then the memory cells pool MC and the ARB population AB are seeded by randomly adding training instances.

2. Memory cell identification and ARB generation

From now on, antigens (training instances) will be presented to the algorithm one by one. If an antigen $ag_i = \{x_i, t_i\}$ is presented to the system, the algorithm will identify a memory cell $mc_{match} = \{x^{mc}_{match}, t^{mc}_{match}\}$ which has the same class label ($t^{mc}_{match} = t_i$) and lowest distance to ag_i. If there is no mc_{match} available at this moment, just let ag_i act as the mc_{match}. This mc_{match} will then be cloned to produce new *mc* clones. First the attributes of mc_{match} will be mutated with a certain probability. If any mutations occurred for this particular clone, the class label will be mutated as well with the same probability

3. Competition for Resources and Development of a Candidate Memory Cell

At this moment, there are a set of ARBs including mc_{match}, mutations from mc_{match}, and others from previous training. AIRS mutates these memory cell clones to generate new ARBs. The number of ARBs allowed to produce is calculated by the product of the hyper clonal rate, clonal rate (both default 10), and the stimulation level (1-distance to ag_i). The newly generated ARBs will be combined with the existing ARBs.

AIRS then employs a mechanism of survival of the fittest individuals within the ARB population. First, each ARB will be examined with respect to its stimulation level when presented to the antigen. In AIRS, cells with high stimulation responses that are of the same class as the antigen and cells with low stimulation response that are not of the same class as the antigen are rewarded most and allocated with more resources. The losers in competing for resources will be removed from the system. Then the ARB population consists of only those ARBs that are most stimulated and are capable in competing for resources.

Then the stop criterion is evaluated. The stop criterion is reached if the average stimulation value of every class subset of AB is not less than the stimulation threshold (default 0.8). Then the candidate memory cell $mc_{candidate}$ is chosen which is the most stimulated ARB of the same class as the training antigen ag_i. Regardless whether the stop criterion was met the algorithm proceeds by allowing the ARBs the opportunity to proliferate with more mutated offspring. This mutation process is similar to the mutation of phase 2, with a small exception: the amount of offspring than can be to produced is calculated by the product of stimulation level and the clonal rate only. If the evaluation criterion was not met in the last test, the process will start again with the stimulation activation and resource allocation step. Otherwise the algorithm will stop.

4. Memory Cell Introduction

Now if $mc_{candidate}$ is more stimulated by the antigen than mc_{match}, it will be added into the memory cell pool. In addition, if the affinity value between and $mc_{candidate}$ and mc_{match} is also less than the product of the affinity threshold (average affinity between all training items) and the affinity threshold scalar (a parameter used to provide a cut-off value, default 0.8), which means $mc_{candidate}$ is very similar to mc_{match}, $mc_{candidate}$ will replace mc_{match} in the set of memory cells. By this mechanism, better classifying memory cells can replace existing memory cells so that the data reduction capabilities of the algorithm are improved. Training is completed now for this training instance ag_i. and the process is repeated from step 2 for the next instance.

5. Classification

With the training completed, the evolved memory cell population $MC=\{mc_1, mc_2...,mc_m\}$ $(m<n)$ will be used for classification using *k*-nearest neighbor. The classification for a test instance will be determined by the majority vote of the *k* most stimulated memory cells.

3 Benchmark Experiments

The goal of the benchmark experiments is to evaluate the predictive performance of AIRS in a real world application setting. We assume that our users are non data mining experts, e.g., business users, who may lack knowledge or time to fine-tune models. To ensure consistency, the experiments for all classifiers were carried under exactly the same conditions, in contrast to some earlier published work on AIRS.

We selected data sets with varying number of attributes, instances and classes, from simple toy data sets to difficult real world learning problems, from the UCI Machine Learning and KDD repositories [1]. The TIC data sets are derived from the standard TIC training set by downsampling the negative outcomes to get an even distribution of the target. In addition, TIC5050S only contains the most relevant according attributes according to a subset feature selection method [11,12].

In the experiments, we selected some representative, well known classifiers as challengers. These classifiers include naive Bayes, logistic regression, decision tables, decision trees (C45/J48), conjunctive rules, bagged decision trees, multi layer perceptrons (MLP), 1-nearest neighbor (1-NN) and 7-nearest neighbor (7-NN). This set of algorithms was chosen because they cover most of the algorithms used in business data mining and correspond to a variety of classifier types and representations - instance based learning, clustering, regression type learning, trees and rules, and so on. Furthermore we added classifiers that provide lower bound benchmark figures: majority class simply predicts the majority class and decision stumps are decision trees with one split only. For AIRS we chose the 1 and 7 nearest neighbor versions of the algorithm. We used the Java version of AIRS by Janna Hamaker [7] and the WEKA toolbox for the benchmark algorithms [9].

All experiments are carried out using 10-fold stratified cross validation. The data is divided randomly into ten parts, in each of which the target class is represented in approximately the same proportions as in the full dataset. Each part is held out in turn and the classifier is trained on the remaining nine-tenths; then the classification accuracy rate is calculated on the holdout validation set. Finally, the ten classificaton accuracy rates on the validation sets are averaged to yield an overall accuracy with standard deviation. To test the robustness of classifiers under real world conditions, all classifiers were run with default settings, without any manual fine-tuning.

4 Results and Discussion

The results of the experiments can be found in Table 1. With respect to the worst case classifiers we highlight some interesting patterns. Almost all classifiers outperform majority vote. The comparison with decision stumps is more striking. For example, for all data sets with the exception of the waveform data set the conjunctive rules classifier does not perform better than decision stumps. Other examples are the TIC data sets: none of the classifiers other than C45 on TICTRAIN5050s perform better than decision stumps. This demonstrates the power of a very simple decision rule in a real world black box modeling environment (see also [8]).

Table 1. Benchmark comparison of average and standard deviation of validation set accuracy for 10 folds

	Sonar	**W. Breast Cancer**	**Waveform**	**Iris**	**Ionosphere**	**Pima diabetes**	**German credit**	**TIC5050**	**TIC5050s**
Majority Class	53.4 ± 1.7	65.5 ± 0.5	33.8 ± 0.1	33.3 ± 0.0	64.1 ± 1.4	65.1 ± 0.4	70.0 ± 0.0	49.7 ± 0.4	49.7 ± 0.4
1-NN	86.6 ±7.0	95.3 ±3.4	73.6 ± 1.3	95.3 ± 5.5	86.3 ± 4.6	70.2 ± 4.7	72.0 ± 3.1	55.9 ± 7.8	59.9 ± 5.1
7-NN	80.8 ± 7.8	96.6 ± 2.2	80.1 ± 1.1	96.7 ± 3.5	85.2 ± 4.3	74.7 ± 5.0	74 ± 4.1	61.1 ± 3.2	65.4 ± 8.4
Decision Stump	73.1 ± 8.3	92.4 ± 4.4	56.8 ± 1.5	66.7 ± 0.0	82.6 ± 4.8	71.9 ± 5.1	70 ± 0.0	68.5 ± 4.7	68.5 ± 4.7
C45/J48	71.2 ± 7.1	94.6 ± 3.6	75.1 ± 1.3	96 ± 5.6	91.5 ± 3.3	73.8 ± 5.7	70.5 ± 3.6	68.1 ± 5.5	69.1 ± 4.4
Naive Bayes	67.9 ± 9.3	96.0 ± 1.6	80.0 ± 2.0	96.0 ± 4.7	82.6 ± 5.5	76.3 ± 5.5	75.4 ± 4.3	62.8 ± 6.4	68 ± 3.3
Conj. Rules	65.9 ± 8.7	91.7 ± 4.5	57.3 ± 1.3	66.7 ± 0	81.5 ± 5.4	68.8 ± 8.67	70.0 ± 0	67.4 ± 3.7	68.3 ± 4.5
Bagging	77.4 ± 0.1	95.6 ± 3.1	81.8 ± 1.4	94 ± 5.8	90.9 ± 4.4	74.6 ± 3.6	74.4 ± 4.9	59.9 ± 5.8	68.4 ± 4.1
Logistic	73.1 ± 13.4	96.6 ± 2.2	86.6 ± 2.3	96 ± 5.6	88.9 ± 4.9	77.2 ± 4.6	75.2 ± 3.4	62.7 ± 4.6	66.5 ± 3.4
MLP	82.3 ± 10.7	95.3 ± 2.6	83.6 ± 1.7	97.3 ± 3.4	91.2 ± 2.8	75.4 ± 4.7	71.6 ± 3.0	60.7 ± 4.3	65.4 ± 4.7
Decision Table	74.5 ± 8.2	95.4 ± 2.7	73.8 ± 1.6	92.7 ± 5.8	89.5 ± 4.5	73.3 ± 3.6	72.2 ± 4.1	61.9 ± 4.5	69.1 ± 5.7
AIRS-1	84.1 ± 7.4	96.1 ± 1.8	75.2 ± 1.7	96 ± 5.6	86.9 ± 3.1	67.4 ± 4.6	68 ± 5.1	56.8 ± 4.4	55 ± 6.5
AIRS-7	76.5 ± 8.4	96.2 ± 1.9	79.6 ± 2 .2	95.3 ± 5.5	88.6 ± 5.0	73.6 ± 3.5	71.4 ± 3.1	57.8 ± 5.5	59.1 ± 6.1

To get a better picture on the relative performance of AIRS we compare it to the average classifier performance (excluding decision stump and majority vote). AIRS-1 performs better than average on 3 of these 9 datasets. AIRS-7 performs better than average on 6 of these 9 datasets. This conflicts with the claims made in earlier studies that were cited in section 2.2. We also made some comparisons to the IB-k algorithms, because these may be closest to a trained AIRS classifier. AIRS-1 improves on IB-1 more often than the other way around; this is probably due to the fact that AIRS-1 provides some useful generalization. However IB-7 performs better than AIRS-7 on all of the data sets. AIRS-7 performs better than AIRS-1on 7 out of 9 data sets. Using more clusters may give better results but not to the extent that IB-7 can be beaten (basically as many cluster centers as data points).

That said, with the exception of AIRS-1 on German credit data, the AIRS algorithms produce at least around average results. This suggests that AIRS is a mature classifier that delivers reasonable results and that it can safely be used for real world classifications tasks.

In our future work we want to make a more extensive investigation into what claims can be made at the various significance levels. In addition, assuming that there will be no classifier that outperforms all other classifiers across all problem domains, we want to investigate on what kind of data sets AIRS performs well, for example by relating properties such as data set size to performance of AIRS relative to other algorithms. Furthermore, we intend to study the relation between AIRS and other algorithms by looking at patterns of performance across algorithms.

6 Conclusions

In this paper we have presented benchmark results for the AIRS immuno-computing algorithm and provided directions for interpretation of these results. We are interested in immuno-computing because it is one of the newest directions in biologically inspired machine learning and focused on AIRS because it can be used for classification, which is one of the most common data mining tasks.

To our knowledge this the first benchmark of AIRS that compares AIRS across a wide variety of data sets and algorithms, using a completely consistent experimental set up rather than referring to benchmark results from literature. In contrast to earlier claims, we find no evidence that AIRS consistently outperforms other algorithms. However, AIRS provides stable, near average results so it can safely be added to the data miner's toolbox. Whether the relative complexity of the algorithm is justified by demonstrating outstanding performance in specific, identifiable problem domains remains a question for further research.

Acknowledgements

We would like thank Janna Hamaker for providing the AIRS source code and Andrew Watkins for his assistance with understanding the details of the AIRS algorithm.

References

[1] Blake, C. and C. Merz. ‘UCI Repository of machine learning databases’, http://www.ics.uci.edu/~mlearn/ MLRepository.html . 1998

[2] Carter, J. H. The immune systems as a model for pattern recognition and classification. Journal of the American Medical Informatics Association 7(1), 28-41, 2000

[3] De Castro, L. N. and F. von Zuben. The clonal selection algorithm with engineering applications. In: D. Whitley, D. Goldberg, E. CantuPaz, L. Spector, I. Parmee, and H. Beyer (eds.): Proceedings of Genetic and Evolutionary Computation. San Francisco, CA, Morgan Kaufman. Pp. 36-37, 2000

[4] De Castro, L.N. and J. Timmis. Artificial Immune Systems: a New Computational Intelligence Approach. Springer Verlag, 2002

[5] Goodman, D. E. , L. Boggess, and A. Watkins. Artificial Immune System Classification of Multiple-Class Problems. In Artificial Neural Networks in Engineering (ANNIE), 2002

[6] Goodman, D. E. , L. Boggess, and A. Watkin. An Investigation into the Source of Power for AIRS, an Artificial Immune Classification System. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), 2003

[7] Hamaker, J. and A. Watkins. Artificial Immune Recognition System (AIRS), Java source code, 2003

[8] Holte, R.: 1993, ‘Very Simple Classification Rules Perform Well on Most Commonly Used Datasets’. Machine Learning 11, 63–91.

[9] Ian H. Witten and Eibe Frank Data Mining. Practical machine learning tools with Java implementation. Morgan Kaufmann, San Francisco, 2000

[10] Marwah G. and L. Boggess. Artidicial immune systems for classification: Some issues. In 1st International Conference on Artificial Immune Systems, pp. 149-153, 2002

[11] van der Putten, P. and M. van Someren (eds) . CoIL Challenge 2000: The Insurance Company Case. Published by Sentient Machine Research, Amsterdam. Also a Leiden Institute of Advanced Computer Science Technical Report 2000-09. June 22, 2000.

[12] van der Putten, P. and M. van Someren. A Bias-Variance Analysis of a Real World Learning Problem: The CoIL Challenge 2000. Machine Learning, vol. 57, iss. 1-2, pp. 177-195, Kluwer Academic Publishers, October 2004,

[13] Timmis, J. and M. Neal. A Resource Limited Artificial Immune System. Knowledge Based Systems 14(3/4), 121-130, 2001

[14] Watkins, A., AIRS: A Resource Limited Artificial Immune Classifier. M.S. thesis, Department of Computer Science. Mississippi State University, 2001

[15] Watkins, A., J. Timmis, and L. Boggess Artificial Immune Recognition System (AIRS): An Immune-Inspired Supervised Learning Algorithm. Genetic Programming and Evolvable Machines, 5 (3): 291-317, 2004

An Analysis of Missing Data Treatment Methods and Their Application to Health Care Dataset

Peng Liu[1], Elia El-Darzi[2], Lei Lei[1], Christos Vasilakis[2], Panagiotis Chountas[2], and Wei Huang[2]

[1] School of Information Management and Engineering, Shanghai University of Finance and Economics, Shanghai, 200433, P.R. China
liupeng@mail.shufe.edu.cn
[2] Health Care Computing Group, School of Computer Science, University of Westminster, London, Northwick Park, HA1 3TP, UK
{eldarze, chounp, vasilc, huangw}@wmin.ac.uk

Abstract. It is well accepted that many real-life datasets are full of missing data. In this paper we introduce, analyze and compare several well known treatment methods for missing data handling and propose new methods based on Naive Bayesian classifier to estimate and replace missing data. We conduct extensive experiments on datasets from UCI to compare these methods. Finally we apply these models to a geriatric hospital dataset in order to assess their effectiveness on a real-life dataset.

1 Introduction

Data Mining (DM) is the process of discovering interesting knowledge from a large amounts of data stored either in databases, data warehouse, or other information repositories [1]. According to [2], about 20% of the effort is spent on the problem and data understanding, about 60% on data preparation and about 10% on data mining and analysis of knowledge, respectively. Why is more than half of the project effort spent on data preparation? Actually, there are a lot of serious data quality problems in real-world datasets. Problems often encountered include incomplete, redundant, inconsistent or noisy data [2]. These serious quality problems if not addressed reduce the performance of data mining algorithms. Hence, in many cases a lot of effort is spent on the data preparation phase in order to achieve a good result. Missing data is a common issue in many real-life datasets. Rates of less than 1% missing data are generally considered trivial, 1-5% manageable. However, 5-15% requires sophisticated methods to handle, and more than 15% may severely impact any kind of interpretation [3].

This paper discusses and evaluates some treatment methods for missing data. Missing mechanism and the guidelines for treatment are presented in Section 2. Section 3 introduces some popular treatment methods of missing data and proposes a new model based on Naive Bayesian Classifier and information gain. Experimental analysis and model comparison are described in Section 4. The proposed models are applied to a hospital dataset and the results are reported in Section 5. Conclusions and further work are discussed in Section 6.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 583 – 590, 2005.

2 Missing Mechanism and Guidelines for Treatment

The effect of the missing data treatment methods mainly depend on the missing mechanism. According to [4], missing data can be classified into three: missing completely at random (MCAR), missing at random (MAR) and not missing at random (NMAR).

Some data mining approaches treat missing data with internal algorithms, say, C4.5. However it is still significantly important to construct complete datasets with treatment methods for missing data for two reasons [5]. First, any data mining approaches can be used with a complete dataset and second, it can prove a basic point for the comparison of the data mining approaches.

In general, treatment methods for missing data can be divided into three common approaches [3]:

Ignore or Discard the Instances Which Contain Missing Data.

Parameter Estimation. In this approach, variants of Expectation-Maximization algorithm are used in the maximum likelihood procedures to estimate the parameters for missing data. These methods are normally superior to ignore or discard methods. However, there are not widely applied mainly because the assumption of variable distributions can not be easily derived and the high degree of complexity for calculation [6].

Imputation Techniques. It uses the present information of the dataset to estimate and replace missing data correspondingly. It aims to recognize the relationships among the values in dataset and to estimate missing data based on these relationships.

In general, missing data treatment methods should satisfy three rules. First, any missing data treatment method should not change the distribution of the dataset. Second, the relationship among the attributes should be retained. Third, it must not be too complex or computationally costly for a method to be applied in real life.

3 Missing Data Treatment Methods

In this section we introduce some popular missing data treatment methods and our proposed models which are based on the concept of information gain and Naive Bayesian Classifier.

3.1 Popular Treatment Methods

Case Deletion. This method discards the cases with missing data for at least one attribute. A variation of this method is to delete the instances or attributes which have high missing rate. However before deleting any attribute, it is necessary to run relevance analysis.

Constant Replacement. The mean (for numeric data) or mode (for nominal data) is used to replace missing data. To reduce the influence of exceptional data, the median can also be used.

Internal Treatment Method for C4.5. This method uses a probabilistic approach to handle missing data [7].

K-Nearest Neighbor Imputation. This method uses k-nearest neighbor (KNN) algorithms to replace missing data. Computational efficiency is thought to be a big problem for this method. While the KNN look for the most similar instances, the whole dataset should be searched. On the other hand, the selection of the value "k" and the measure of similarity will greatly influence the results.

3.2 Naive Bayesian Imputation (NBI)

Naive Bayesian Classifier is a popular classifier, not only for its good performance, but also for its simple form [6]. It is not sensitive to missing data. In this case, the prediction accuracy of Naive Bayesian Classifier is always higher than that of C4.5 [1]. Missing data treatment methods based on Naive Bayesian Classifier and concept of information gain, named Naïve Bayesian Imputation (NBI), consist of two phases.

In phase1, the order of the attributes to be treated is defined. This process treats attributes with missing data one by one. It is an iterative process. The order of the attribute to be treated affects the overall results. Each attribute's importance for classification is different. The performance of random order of attributes is therefore weak and unreliable. To overcome this inherent deficiency in random order we propose several methods: (1) Using the original data to estimate all the missing data. It is independent of the estimate order of attributes. (2) On descending order of missing rate. Initially, it replaces missing data in the attribute with the highest missing rate, then uses the modified dataset to estimate and replace missing in the next attribute. (3) On descending order of information gain. For the classification task of data mining, information gain reflects the importance of the attributes for the classification task. (4) On descending order of the weighted index by missing rate and information gain. This method combines missing rate and information gain concepts. It is proved to be effective. (5) In experiments, while using methods 3 or 4, the performance of the algorithm is best when the first n (usually 3~4) attributes have been treated. Therefore, treatment of the first three or four attributes will be enough. Method 1 can be applied to the remaining attributes. In this way, the quality and efficiency can be balanced.

In phase2, the Naive Bayesian classifier is built to estimate and replace the missing data, using the attribute defined in the first phase as class attribute and the whole dataset as the training subset. According to the different estimate order of attributes, NBI has five different combinations, namely Model 1, Model 2, Model 3, Model 4, and Model 5.

4 Experimental Analysis

To compare the methods introduced in Section 3, we use three datasets from UCI [8], namely, Nursery, Crx and German (Table 1).

Table 1. Datasets summary

dataset	Inst.	Attar.	class
Nursery	12960	8	5
Crx	690	15	2
German	1000	20	2

In order to evaluate the performance of missing data treatment methods, Decision tree C4.5 classifier is built on the modified dataset. If the performance of the classifier is turned out to be satisfactory, the performance of missing data treatment model is considered to be satisfactory. Predictive accuracy of model, predictive accuracy of class and predictive profit (against C4.5 internal method) which are used in the paper for measuring performance of data mining algorithm are defined as follows:

$$\text{Prediction accuracy of model} = \frac{\text{Number of correct categorize d instances}}{\text{Total number of instances}} \times 100\% \tag{1}$$

$$\text{Prediction accuracy of class} = \frac{\text{Number of correct categorize d instances in the class}}{\text{Total number of instances in the class}} \times 100\% \tag{2}$$

$$\text{Prediction profit} = \frac{\text{Prediction accuracy - prediction accuracy of C4.5intern al method}}{\text{prediction accuracy of C4.5intern al method}} \times 100\% \tag{3}$$

The "number of correct categorized instances" and the "number of correct categorized instances in the class" are calculated by C4.5 using the modified dataset and the "prediction accuracy of C4.5 internal method" comes from the classifier which uses the modified dataset built by C4.5 internal missing data treatment method.

The experiments are as follows: Firstly, the datasets are randomly divided into two subsets: 66% of records as a training subset and the remaining 33% as a testing subset. Missing data are artificially implanted in different rates, from 10% to 60% of records into the training subsets in order to maintain the integrity of testing subsets. Decision tree C4.5 is used as a classifier in this paper and the analysis for most representative node attributes is desirable. Hence attributes with high information gain are selected to be inserted with missing data. Finally, three methods are applied into the training subsets these are: Mean replacing method, C4.5 internal method and Model 4 on Naive Bayesian Classifier. The experiments are repeated three times for each method and the average error rate is calculated. Due to the lack of space, only part of the results is presented in Tables 2 and 3, and the comparative performances are graphed in Figures 1 and 2.

From Figures 1 and 2 we can see that, in most cases Model 4 of NBI is superior to mean replacing method and C4.5 internal method. For dataset Nursery, while the

Table 2. Error rates for Dataset Nursery

Missing	Att.: heal, par			Att.: heal, par, fina		
proportion	C4.5 (%)	Mean (%)	Bayes (%)	C4.5 (%)	Mean (%)	Bayes (%)
0%	3.8±0.3	-	-	3.8±0.3	-	-
10%	4.2±0.2	4.2±0.3	3.9±0.1	4.5±0.1	4.5±0.3	4.2±0.6
20%	4.7±0.3	5.3±0.6	5.0±0.6	5.1±0.4	5.5±0.4	5.3±0.2
30%	5.7±1.1	7.0±1.9	6.4±0.5	6.9±0.8	6.8±0.3	6.6±0.1
40%	7.4±1.0	7.0±0.5	7.1±0.4	8.1±0.6	7.4±0.2	7.5±0.8
50%	10.8±0.1	10.5±1.0	8.8±1.0	11.0±0.1	10.1±0.3	10.1±0.4
60%	11.0±0.3	13.6±6.3	9.8±1.2	11.8±0.4	15.4±6.9	11.3±0.9

Table 3. Error rates for Dataset Crx

Missing	Att.: A5, A6			Att.: A5, A6, A2		
proportion	C4.5 (%)	Mean (%)	Bayes (%)	C4.5 (%)	Mean (%)	Bayes (%)
0%	13.5±0.3	-	-	13.5±0.3	-	-
11%	13.5±0.3	13.2±0.4	13.5±0.3	13.5±0.3	13.5±0.8	13.5±0.3
21%	13.4±2.3	12.3±0.9	13.5±0.3	13.7±2.2	12.3±0.9	12.1±1.1
35%	14.6±1.3	14.3±2.3	12.9±0.7	14.6±1.3	13.7±2.9	12.3±0.9
50%	13.8±2.2	13.4±0.7	13.2±1.2	14.3±1.6	13.0±0.3	13.5±1.1
60%	17.7±4.6	15.3±2.7	12.6±0.5	16.0±0.5	12.7±0.2	12.6±1.2

proportion of missing data is small, the performances of the three methods are similar. As the missing rate goes beyond 40%, the difference among the three methods becomes more obvious. The performance of the mean replacing method worsens as the missing rate increases. However, in the tree structure, the node attributes and their levels do not change markedly. Increases in the proportion of the missing data proportion do not influence the structure of the decision tree, because the information gain ratio of the attribute heal, par and fina are much bigger than that of other attributes. The performance of C4.5 internal methods is very similar to the performance of model 4.

Results for dataset Crx are illustrated in Figure 2. As the proportion of missing data increases, the classification error rates of both mean replacing methods and C4.5 internal methods increase. However results for Model 4 remain stable and very close to the rate of the original dataset which does not have missing data. In this experiment, attribute A5, A6 and A2 were found to be strongly dependent on other attributes. In order to find the dependency relationship among attributes, each attribute, one by one, is predicted by the other attributes in dataset. If the error rate for a classifier is low then the attribute has a strong relationship with other attributes. For Crx, error rates of these three attributes are very low, about 25%. It makes the new dataset completed by Model 4 very close to the original dataset which does not have missing values. Therefore, the predictive error rates of Model 4 always fluctuate around that of the original dataset. We also find that an increase in the missing data does not affect the predictive error rates of missing data, but an increase in the number of attributes with missing data will influence the performance of the missing treatment methods.

For dataset German, the C4.5 performed well when missing data are inserted into one attribute. When missing data are inserted into three attributes, the performances of the three methods are similar to each other. For both Crx and German datasets, as the proportion of missing data increases, the nodes of the decision tree changed. The high levels attributes became lower levels or are not even selected as nodes. Using the attributes with weak classifying power will reduce the performance of the decision tree [3]. After changing the structure of the decision tree, the performance of Model 4 is better than that of C4.5 internal methods.

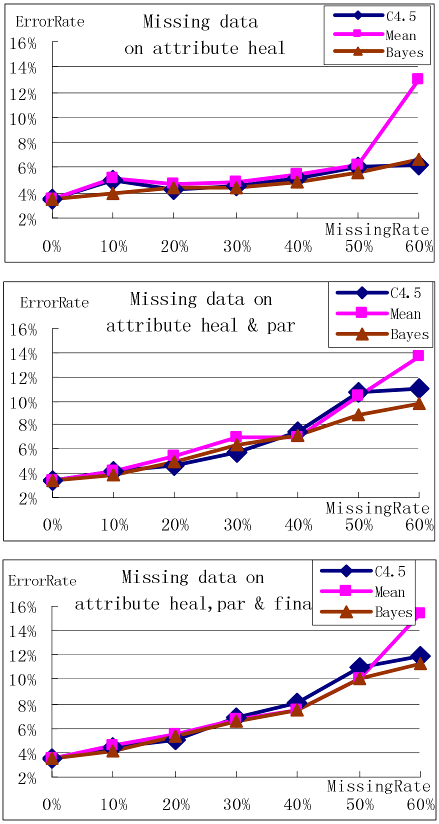


Fig. 1. Comparative results for Nursery

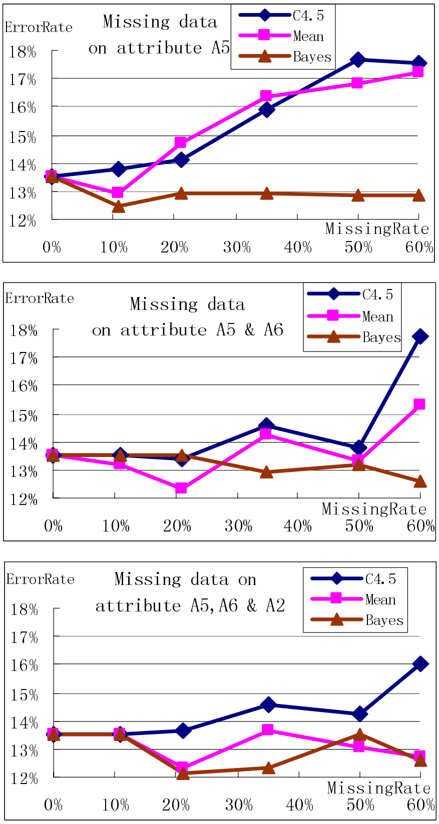


Fig. 2. Comparative results for Crx

5 Application in Healthcare

The approaches of data mining have a wide use in the healthcare domain. If the inpatient length of stay (LOS) can be predicted efficiently, the planning of hospital resources can be greatly enhanced [9]. However, most healthcare datasets contain a lot of missing data. Treatment methods for missing data discussed earlier in this paper are applied to a real life dataset to improve the accuracy of predictive models of LOS.

5.1 Clinics Dataset

The Clinics dataset contains data from a clinical computer system that was in use between 1994 and 1997, for the management of patients in a Geriatric Medicine department of a metropolitan teaching hospital in the UK [10]. It contains 4722 patient records including patient demographic details, admission reasons and LOS. For ease of analysis, LOS was categorized into three groups: short-stay group (0-14 days), medium-stay group (15-60 days) and long-term group (61+ days) (variable LOS GROUP). These boundaries are chosen in agreement with clinical judgment to help describe the stages of care in such a hospital department. The missing data account for a lot in the Clinics dataset. There are 3017 instances (63.89%) that contain missing data. The proportion of missing data per LOS GROUP is 63.29%, 61.81% and 74.86%, respectively. There are 8 attributes with missing data.

5.2 Practice and Analysis

After Applying these five models based on Naive Bayesian Classifier proposed in this paper to Clinics dataset we obtained all the prediction accuracy and the prediction profits as in Table 4.

Table 4. The prediction profits against C4.5 and prediction accuracy of all models and classes

Model	accuracy of model	accuracy of class			prediction profits of model	prediction profits of class		
		Short stay	Medium stay	Long stay		Short stay	Medium stay	Long stay
C4.5	52.44%	44%	69%	10%	-	-	-	-
Model 1	55.62%	46%	69%	29%	6.10%	4.5%	0.0%	190.0%
Model 2	55.23%	46%	68%	30%	5.30%	4.5%	-1.4%	200.0%
Model 3	55.87%	48%	68%	31%	6.50%	9.1%	-1.4%	210.0%
Model 4	55.82%	47%	69%	31%	6.50%	6.8%	0.0%	210.0%
Model 5	55.53%	45%	70%	30%	5.90%	2.3%	1.5%	200.0%

From Table 4, we can see that the average prediction accuracy of NBI is higher than the C4.5 internal model, especially for the long stay category. In Clinics dataset, 6 attributes with missing data were treated. The missing proportion for two of them is about 20% and for one above 40%. In this case, NBI outperform C4.5 internal method. Among these five models, Model 4 and Model 5 performed better than the others. Furthermore, for Model 3, 4 and 5, three or four attributes were enough to obtain a good result. NBI can improve the prediction accuracy of the whole model, especially for the long stay category. The highest prediction profit for the long stay has reaches 210%. In the cases where the missing data proportion is large, there are many attributes with missing data and a strong relationship among attributes is exhibited, treatment methods for missing data based on Naive Bayesian Classifier perform well.

6 Conclusions

This paper presents a comparative analysis of several well known missing data treatments and proposes an efficient and effective missing data predictive model, NBI. These methods were tested on the Nursery, Crx, German datasets from UCI and Clinics dataset from a geriatric department. NBI performs better than C4.5 internal model. The type of the attributes with missing data affects the results of the treatment methods. While the important attribute for classifying contains fewer missing data or none, C4.5 internal model perform very well. However, treatment methods based on Naive Bayesian Classifier are more commonly used. Comparatively, in the case of high missing data proportion and many attributes with missing data, NBI will perform more satisfactorily.

References

1. Han J. and Kamber M., *Data Mining Concepts and Techniques*, Morgan Kaufmann Publishers, 2000
2. Cios K.J. and Kurgan L., *Trends in Data Mining and Knowledge Discovery*. In N.R. Pal, L.C. Jain, and Teoderesku N., editors, *Knowledge Discovery in Advanced Information Systems*. Springer, 2002.
3. Acuna E. and Rodriguez C., *The treatment of missing values and its effect in the classifier accuracy*. In D. Banks, L. House, F.R. McMorris,P. Arabie, W. Gaul (Eds).Classification, Clustering and Data Mining Applications. Springer-Verlag Berlin-Heidelberg, 639-648. (2004).
4. Little R. J. and Rubin D.B., *Statistical Analysis with Missing Data*. Second Edition. John Wiley and Sons, New York. (2002).
5. Magnani M., *Techniques for Dealing with Missing Data in Knowledge Discovery Tasks*, accessed at http://magnanim.web.cs.unibo.it/index.html on Aug. 28, 2004
6. Hand D., Mannila H., and Smyth P., *Principles of data mining*. MIT Press, 2001
7. Quinlan J. R., *C4.5 Programs for Machine Learning*. Morgan Kaufmann, CA, 1988.
8. Merz C. J. and Murphy P. M., UCI Repository of Machine Learning Datasets, 1998. http://www.ics.uci.edu/ mlearn/MLRepository.html.
9. Marshall, A. Vasilakis, C. and El-Darzi, E. *Modelling Hospital Patient Flow: Recent Developments and Future Directions*. (accepted) Health Care Management Science.
10. Marshall, A. McClean, S. Shapcott, C. Hastie, I. and Millard, P. *Developing a Bayesian Belief Network for the Management of Geriatric Hospital Care*. Health Care Management Science, 4(1), pp 25-30, 2001.

Parallel Genetic Algorithm and Parallel Simulated Annealing Algorithm for the Closest String Problem

Xuan Liu, Hongmei He, and Ondrej Sýkora

Department of Computer Science, Loughborough University,
Loughborough, Leicestershire, LE11 3TU, The United Kingdom
{x.liu, h.he, o.sykora}@lboro.ac.uk

Abstract. In this paper, we design genetic algorithm and simulated annealing algorithm and their parallel versions to solve the Closest String Problem. Our implementation and experiments show usefulness of the parallel GA and SA algorithms.

1 Introduction

With the development of biology and pharmacy industry, some problems of bioinformatics become of great importance. One of such problems is finding the closest string representing a set of genes. Also creation a drug, which would kill several closely related pathogenic bacteria, while it might be relatively harmless to humans, requires solution of the problem. Another example can be finding a consensus sequence, which is a single sequence that best represents a collection of related sequences.

Let us define the [illegible] $d(x,y)$ between two strings x and y of the same size as the number of positions in which x and y differ. We define the Closest String Problem as follows.

Closest String Problem (CSP): Given a set $= \{s_1, s_2, \ldots, s_n\}$ of n strings each of length m, over an alphabet A, find a string x of length m over A minimising d such that for each string s_i in , $d(s_i, x) \leq d$.

As the CSP is NP-HARD [1], several different heuristic and approximation algorithms have been implemented. Lanctot et al. [2] designed $\frac{4}{3}$ approximation algorithms, Li et al. [3] presented a PTAS. They used the standard linear programming and random rounding technique in their approximation algorithms.

In this paper we use machine learning approaches genetic and simulated annealing algorithms to solve the CSP.

The idea of genetic algorithm (GA) was originated by Holland in the 1960s. GAs are based on the principles of natural selection and adaptation and are claimed to be able to explore good solutions quickly in a large and complicated search space. The power of the algorithms comes from the mechanism of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 591–597, 2005.

evolution, which allows searching through a huge number of possibilities for solutions.It is that chromosomes are the information carriers and that the evolution process works at the chromosome level through reproduction. The reproduction can be made by either combining chromosomes from the parents to produce offspring, a process called crossover, or by a random change occurring in the chromosome pattern, termed mutation.

A GA creates an initial population of solutions at beginning. Then, the GA evaluates fitness function to all individuals of population, to characterise them from the most fit to the least fit. Afterwards, genetic operators transform the parent chromosomes to their offspring according to the criteria of fitness. The GA repeats the processes of selection, crossover and mutation to artificially simulate genetic operations. If the GA reaches the termination of the algorithm, the GA will output the best population.

Simulated Annealing (SA) is an advanced Local Search method, which finds its inspiration from the physical annealing process studied in statistical mechanics [4]. The SA algorithm repeats an iterative procedure that looks for the better configurations while offering the possibility of accepting worse configurations. The SA algorithm provides opportunities to jump out from local optima.

Parallel computing has been a useful tool for improving running time and enlarge feasible size of problems with low cost. In this paper, we focus on the implementation of sequential and parallel GA and SA. We compared their performance by using different algorithm parameters and realised series of experiments focusing on the parallel processing issue.

2 Description of Algorithms

2.1 The Sequential GA for CSP

The GA generates an initial population $P(t)$ of random candidate solutions $ind_0, \ldots, ind_{popsize-1}$ for a set of input strings to the CSP problem. Each individual of the initial population contains a string, (of length m) over the alphabet $A = \{a, c, g, t\}$. Let d_{max} be the largest value of Hamming distance between an individual of the population and any string in .

We define the Hamming distance between two strings as the number of positions on which they differ. For example, given $string_A$ = "*actgatttggcc*", $string_B$ = "*gctaggttccgg*", the Hamming distance is 8, since there are 8 positions, on which the characters of the strings $string_A$ and $string_B$ are different. The $fitness$ function is defined as the difference $m - d_{max}$. A larger $fitness$ value means a closer string, so we try to maximise the fitness value.

We use multi-point-crossover (MPX) in our GA as follows: two parental individuals, ind_x and ind_y, are chosen randomly depending on the probability p_{ind}. The probability of an individual is in proportion to the fitness of the individual for selection.

Two chosen parents exchange parts between two randomly picked points in the strings to form the offspring. The offspring have a new order of the strings, one part from $father$ and the other part from *mother*.

Afterwards, a *mutation* on any individual is executed with some probability p_m. Two positions are randomly chosen and exchanged in the individual. Repeat this procedure until the termination criterion is met.(see Algorithm 1)

Algorithm 1. [Sequential Genetic Algorithm Structure]

```
$t \leftarrow 0$
initialize $P(t) = \{ind_i \in P(t), i = 0, 1, ...popSize - 1\}$
evaluate $P(t)$ to get the fitness of each individual
calculate the probability of each individual, $p_i \propto ind_i.fitness$
$currBest$=best_ind($P(t)$);
$bestInd = ind_{currBest}$;
while ( $t <$ TERMINATION_CRITERION) do
    $i$=0;
    while ($i < popSize/2$) do
        select ( $ind_x$ $ind_y$ ) from $P(t)$ according their $p_{ind}$ - if two random values are
        located in the probability, $p_x$ and $p_y$ respectively
        $\{chd_{(2i)}, chd_{(2i+1)}\}$ = crossover( $ind_x$ , $ind_y$ )
    end while
    $i$=0;
    while ($i < popSize$) do
        $r = rand()mod100$
        if $r < p_m$ then
            mutate( $chd_i$)
        end if
        $P(t+1) \leftarrow P(t+1) \bigcup chd_i$
    end while
    evaluate $P(t+1)$ to get the fitness of each individual
    calculate the probability of each individual, $p_i \propto ind_i.fitness$
    $worst$=worst_ind($P(t+1)$);
    $ind_{worst} \leftarrow bestInd$
    $currBest$=best_ind($P(t+1)$);
    if ($ind_{currBest}.fitness > bestInd.fitness$) then
        $bestInd = ind_{currBest}$;
    end if
    $t \leftarrow t + 1$
end while
```

The algorithm terminates when the number of generations reaches a preset value.

2.2 The Parallel Island GA Approach

In this paper, we use [illegible] to implement our parallel GA-CSP algorithm. The idea of [illegible] is to distribute the total population to the available processors. A sequencial GA runs with a sub-population on each processor. The sub-populations are independent from each other and therefore each processor starts with a randomly generated subpopulation. Each processor will send its current best individual to another processor (randomly chosen with a probability, here, we set 1%) and receive the best individual from it in turn, and both replace the current worst individual by the received one.

2.3 The Sequential SA for CSP

Simulated Annealing is a generalisation of the Monte Carlo method for examining the equations of a state and the frozen state of $n-$body systems [5]. The idea comes from freezing of liquids or crystallisation of metals by the process of annealing. In this process, the system initially starts with a high temperature, then it is slowly cooled down until it approaches a "frozen" ground state. During the cooling process, the system is approximately in thermodynamic equilibrium. In the original Metropolis scheme [5] an initial state of a thermodynamic system was chosen at an energy E and a temperature T, holding T constant, while the initial configuration is perturbed and the change in energy ΔE is computed. If the change in energy is negative, the new configuration is accepted. If the change in energy is positive, it is accepted with a probability given by the Boltzmann factor $e^{-\frac{\Delta E}{T}}$. The system will iterate this procedure several times to get good sampling statistics for the current temperature, and then the temperature is decremented and the entire process is repeated until a frozen state is achieved at $T = 0$.

Simulated annealing has been used in various combinatorial optimization problems.[4].

CSP belongs to a class of discrete minimisation problems, so we need to map the state of the thermodynamic system, which is analogous, to the current solution of the discrete problem as below.

- Configuration: A string over the alphabet, $\{A\}$, which has the same length as the strings in S.
- Rearrangement: A point in a string is picked randomly, then the two parts divided by the point are exchanged.
- Energy Function: E is defined as follows:$E = max_{s_i \in S} d(s_i, x)$, where x is the current solution string.
- Annealing schedule: Avoidance of entrainment in local minima is dependent on the "annealing schedule", the choice of initial temperature, how many iterations are performed at each temperature, and how much the temperature is decremented at each step as cooling proceeds. This requires a lot of experimentation. We first do some random rearrangements, and use the results to determine the range of values of ΔE that will be encountered from one state to another state. Choosing a starting value for the parameter T which is considerably larger than the largest ΔE normally encountered. In our experiments, T is half of the possible maximal Hamming Distance, which is the length of string, so $T = m/2$. At each temperature 100

iterations are performed. The temperature reduction factor, γ, is set to 0.9. When the termination criterion, $T = 0.001$, is met, algorithm stops. (Algorithm 2)

Algorithm 2. [Sequential SA algorithm for CSP]

```
randomly generate an initial string $s_c$
set an initial $T = T_{max}$
set an initial repeat times $L$
set $\gamma$
while termination = false do
    for $0 \leq$ I $< L$ do
        string $s_n$ is obtained by rearrangement of $s_c$
        $\Delta = f(s_n) - f(s_c)$
        if $\Delta \leq 0$ or $((\Delta > 0)$ and $( e^{-\frac{\Delta}{T}}$ $is\ verified)$ then
            $s_c \leftarrow s_n$
        end if
        if $T < T_{min}$ then
            termination = true
        end if
    end for
    $T \leftarrow \gamma T$
end while
output
```

2.4 The Parallel SA Approach

We used PVM to implement our parallel SA algorithm for CSP. The main idea of parallelism comes from the genetic algorithm [illegible]. Let each processor run the sequencial SA algorithm independently. At a randomly chosen time, the master in PVM randomly chooses two processors, and the current temperatures of these processors are exchanged. Each of the two processors will set the received temperature to be its initial temperature for the next annealing process. Each processor will return its best result to the master after having reached the termination-condition. Otherwise, the processors will carry on in the annealing process and exchange their temperature with others on the master request.

3 Test Results

We have two kinds of test platform, one is that we fix the string numbers and change the string length (see Table 1), another is that we fix the string length and change the string numbers (see Table 2). For each of the randomly generated problem instances every algorithm was run 20 times. We got the average results. We used the following configurations of sequential GA: the total population size

Table 1. The average test results for 10 strings and different string length. The results in the figure are the total length of string minus the maximal values of the Hamming distance between the return string and each of strings in S

String Length	GA	SA	Parallel-GA	Parallel-SA
10	4	4	4	4
20	7	7	8	8
30	11	9	13	11
40	13	11	16	14

Table 2. The average test results for the strings, which are 20 bits long, but with different strings numbers

Strings	GA	SA	Parallel-GA	Parallel-SA
10	8	8	9	8
20	7	7	9	7
30	7	6	8	7
40	6	5	7	6

was 50, the probability of mutation was 0.5 and the number of generation was 2000. For our sequential SA algorithm we used the following configurations: the initial temperature was $m/2$, the temperature reduction factor was 0.9, and the minimum temperature was 0.001.

We tested our PVM implementation of the algorithms on a cluster of 20 Sun ULTRAsparc workstations running Debian GNU/Linux. They are connected with 100Mbit Ethernet using Cisco 2950 switches. We compared the sequencial and parallel GA and SA algorithms. Both parallel algorithms used 4 processors. In our parallel [illegible] GA, we set only 1% opportunity that any two processors will exchange their individuals.

From our results, one can see that the parallel GA produces better results than others. The parallel algorithms produce better results than their sequential versions. One possible reason is that parallel algorithms efficiently use the processors to search in a larger solution space. Our tests show usefulness of parallelism: increasing the number of processors in the parallel GA and SA algorithms, produces better solutions.

4 Conclusions

In this paper, we designed sequential GA and SA algorithms for CSP. Comparison of all four parallel and sequential algorithms shows the superiority of the parallel island GA algorithm and usefulness of the parallel versions of our machine learning algorithms.

References

1. Frances, M., Litman, A., On covering problems of codes,*Theory of Computing System* ,**30**,(1997) 113-119.
2. Lanctot, K., Li, M., Ma, B., Wang, S., Zhang, L., Distinguishing string selection problems, *Information and Computation*, **185**,(2003) 41-55.
3. Li, M., Ma B., Wang, L., Distinguish string search problems, *Processings of the Thirty-first Annual ACM Symposium on Theory of Computing*,(1999) 473-482.
4. Kirkpatrick, S., Gelatt, C., Vecchi, M., Optimization by simulated annealing, *Science*,**220**,(1983) 671-680.
5. Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller A., Teller, E., Equation o State Calculations by Fast Computing Machines, *J. Chem Phys.*, **21**, (1953) 1087 - 1092.

Mining Interesting Association Rules in Medical Images

Haiwei Pan, Jianzhong Li, and Zhang Wei

Dept. of Computer Science, Harbin Institute of Technology, Harbin, P.R. China
heaven_007cn@yahoo.com.cn
{lijzh, wzhang74}@hit.edu.cn

Abstract. Image mining is more than just an extension of data mining to image domain but an interdisciplinary endeavor. Very few people have systematically investigated this field. Mining association rules in medical images is an important part in domain-specific application image mining because there are several technical aspects which make this problem challenging. In this paper, we extend the concept of association rule based on object and image in medical images, and propose two algorithms to discover frequent item-sets and mine interesting association rules from medical images. We describe how to incorporate the domain knowledge into the algorithms to enhance the interestingness. Some interesting results are obtained by our program and we believe many of the problems we come across are likely to appear in other domains.

1 Introduction

Mining knowledge from large databases has been the focus of many recent studies and applications. However, most of these methods have emphasized corporate data typically in alphanumeric databases. Little research has been conducted on mining image. Advances in image acquisition and storage technology have led to tremendous growth in very large and detailed image databases [4]. A vast amount of image data is generated in our daily life and each field, such as medical image (CT images, ECT images and MR images etc), satellite images and all kinds of digital photographs. These images involve a great number of useful and implicit information that is difficult for users to discover.

Image mining can automatically discover these implicit information and patterns from the high volume of images and is rapidly gaining attention in the field of data mining. Image mining is more than just an extension of data mining to image domain. It is an interdisciplinary endeavor that draws upon computer vision, image processing, image retrieval, machine learning, artificial intelligence, database and data mining, etc. While some of individual fields in themselves may be quite matured, image mining, to date, is just a growing research focus and is still at an experimental stage. Research in image mining can be broadly classified to two main directions: (1) domain-specific applications; (2) general applications [7]. Data mining in medical images belongs to the first direction.

A few interesting studies and successful applications involving image mining have been reported. For example, [2] describes the CONQUEST system that combines satellite data with geophysical data to discover patterns in global climate change. The

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 598 – 609, 2005.

SKICAT system [1] integrates techniques for image processing and data classification in order to identify “sky objects” captured in a very large satellite picture set. A multimedia data mining system prototype MultiMediaMiner [3, 4] uses a data cube structure for mining characteristic, association, and classification rules. However, the system does not use image content to the extent we wanted. In [9], localization of the visual features, their spatial relationships and their motion in time (for video) are presented. A discovering association rules algorithm based on image content from a simple image dataset is presented in [10]. Some algorithms used in the medical images [8, 11, 12] are generally for classification. [13, 14] propose the methods for mining association rules in relational databases. They are not suitable to find rules in the medical images because there are important differences between relational databases versus image databases: (1) Absolute versus relative values. In relational databases, the data values are semantically meaningful. For example, one item is milk is well understood. However, in medical image databases, the data values themselves may not be significant unless the domain of medicine supports them. For example, a grey scale value of 46 could appear darker than a grey scale value of 87 if the surrounding context pixels values are all very bright. (2) Spatial information (Independent versus dependent position). Another important difference between relational databases and medical image databases is that the implicit spatial information is critical for interpretation of image contents but there is no such requirement in relational databases.

In this paper, we extend the concept of association rule using object, object-based attribute and relationship predicate, and image-based attribute in medical images. To detect objects in medical images, a progressive water immersion algorithm is presented. Then we propose a new method with guidance of domain knowledge to discover frequent item-sets and mine interesting association rules from medical images. Throughout the paper we try to provide a general framework to understand our approach. We believe many of the problems we are facing are likely to appear in other domains. As such this work tries to isolate those problems which we consider will be of most interest to the database community doing research on association rules.

The rest of the paper is organized as follows: section 2 is pre-processing describing an algorithm to detect objects in medical images. Association rule mining in medical images is presented in section 3. Section 4 briefly reports our performance study. Conclusions are presented in section 5.

2 Pre-processing

Since the images we got were raw Computerized Tomography (CT) scans that were scanned at different illumination conditions, some of them appeared too bright and some were too dark. We should digitize them to no loss, no compression and 256 gray scale images through special medical scanner.

To discover association rules, our medical images have to be transformed into a suitable format. In this paper, we firstly use progressive water immersion method with guidance of domain knowledge to detect region of interest (ROI) in medical images, then we combine these ROIs with their location, size and other descriptors to form a table for mining.

Water immersion algorithm is considered to be a powerful technique for ROI detection. It works by grouping pixels with similar gradient information. Direct application of water immersion method to the digitized medical images typically produces over-segmentation of the trivial regions. Instead, we propose a progressive water immersion algorithm with guidance of domain knowledge to cope with this situation. Details of the algorithm follow.

First, an N×N window is used to locate the local optimal points in the image. For each segmented patches, we place the center of the window over each pixel in the patches. If the grey level of the central pixel is optimal with respect to all the other pixels in the window, we say that the central pixel is a local optimum; otherwise, the window will move to be centered at another pixel to continue the search for all local optimal points. At the end of this phase, all the optimum is marked and they will be treated as the starting seeds for water immersion method. One advantage of using the sliding window approach is that with the appropriate window size, it is possible to eliminate a large amount of optimal points that correspond to the light and dark reflection regions thus removing false detection. This is because the grey level of the optimal points corresponding to the light and dark reflection patches are generally lower and higher than that of potential ROI. Given that the distances between the optimal points of the light and dark reflection patches and the nearest optimal points of the neighboring ROI are generally less than that between two touching ROI, it is possible to set the window size in such a way that these false optimal points are 'absorbed' by the neighboring ROI optimal points while the true optimal points are not affected.

Having identified the true optimal points, water immersion process starts from these detected points and progressively floods its neighboring pixels. The neighboring pixels are defined to be the 8-direction neighbors. These neighbors are placed in a growing queue structure sorted in descending order of the grey level of the pixels. The lowest and highest grey pixel in the growing queue will be 'immersed' first but respectively and it is marked as belonging to the same region label as the current seed. The marked pixel is then removed from the growing queue. All neighboring pixels whose grey level is lower or higher than the marked pixel are added to the growing queue. This immersion process continues until the growing queue is empty.

Unfortunately, simple application of the water immersion technique has the tendency of over-flooding. To overcome this problem, our progressive water immersion ignores all those pixels whose grey level doesn't belong to the bound GS[5]. Bound GS is defined with guidance of domain knowledge and includes low bound GS and high bound GS that describes the degree of dark and light. So the optimality of point in a certain region is defined as follows:

$$\text{optimality}= \begin{cases} \text{maximal grey level, if point belongs to high bound pixel set ;} \\ \\ \text{minimal grey level, if point belongs to low bound pixel set ;} \end{cases}$$

After the above process, all the ROIs are detected and we will call them objects later, see figure 1. Next, images with many different objects are represented by transactions, but there exist relationship among these objects and identical objects (either bright or dark) can repeat in an image. Therefore, we use a table to describe these transactions, see table 1. In table 1, Part I and Part II is image id and object id respectively. Part III is the attribute set of object, for example, grey level, location, etc., and Part IV is the relationship predicate set, for instance, next-to, inside and predicate related to domain. Part V is the attribute set of image.

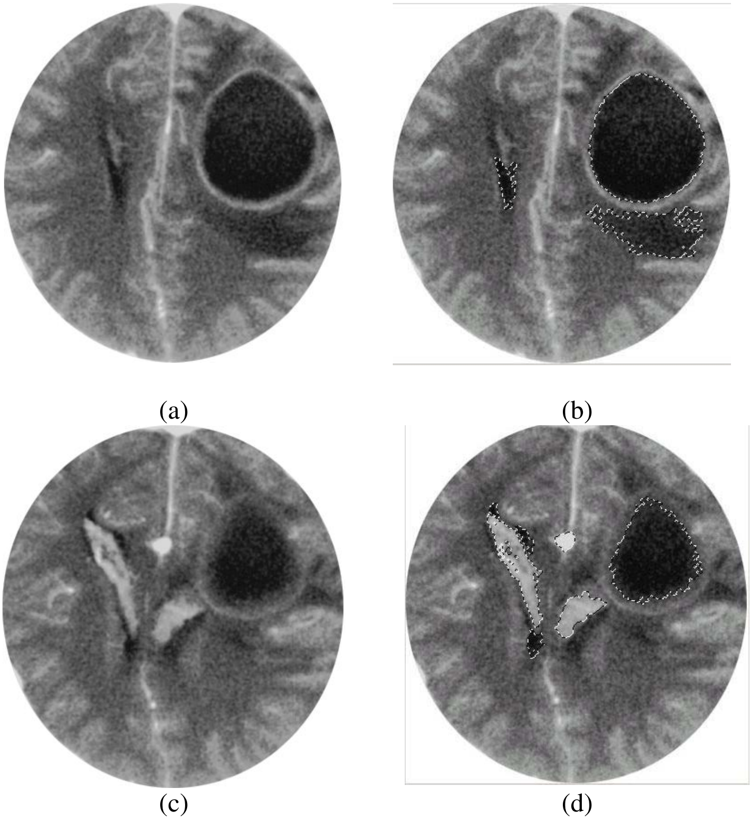

(a)
(b)
(c)
(d)

Fig. 1. Figure (a) and (c) are two original brain images. Progressive water immersion algorithm is used to mark the objects with dotted line in figure (b) and (d)

Table 1. Images are modeled by transactions

Part I	*Part II*	*Part III*			*Part IV*			*Part V*		
Image ID	Object ID	$oattr_1$	…	$oattr_n$	RP_1	…	RP_m	$iattr_1$	…	$iattr_k$
IM_1	O_1	$oattr_1$_v	…	$oattr_n$_v	O_2	…	O_1,O_2	$iattr_1$_v		$iattr_1$_v
IM_1	O_1	…	…	…	…	…	…	…	…	…
IM_1	O_2	…					…			
IM_2	O_1	…					…			
IM_2	O_2	…					…			
…	…	…					…			
IM_n	O_n	…	…	…	…	…	…	…	…	…

3 Association Rules in Medical Images

3.1 Definition

We extend the traditional concept of association rule to the medical image set.

Definition 2.1. An association rule in medical images is a rule that associates visual object features and the relationship among objects in images, and is of the form:

$\alpha_1 P_1 \wedge \alpha_2 P_2 \wedge \ldots \wedge \alpha_n P_n \wedge \lambda_1(P_1, P_2, \ldots, P_n) \wedge \lambda_2(P_1, P_2, \ldots, P_n) \wedge \ldots \wedge \lambda_g(P_1, P_2, \ldots, P_n) \rightarrow \beta_1 Q_1 \wedge \beta_2 Q_2 \wedge \ldots \wedge \beta_m Q_m \wedge \gamma_1(Q_1, Q_2, \ldots, Q_m) \wedge \gamma_2(Q_1, Q_2, \ldots, Q_m) \wedge \ldots \wedge \gamma_h(Q_1, Q_2, \ldots, Q_m)$ (c%)

where c% is the confidence of the rule, one or more P_i, $i \in [1..n]$ and Q_j, $j \in [1..m]$ are not just features or objects, but can also be related descriptors of image and object, the mean over their histogram etc., and α_i, $i \in [1..n]$, and β_j, $j \in [1..m]$, are integers quantifying the occurrence of the object or feature or item. $\alpha_n P_n$ is true if and only if P_n has α_n occurrences. λ_i, $i \in [1..g]$, and γ_j, $j \in [1..h]$, are relationship predicates indicating the relationship of the object or feature or item. $\lambda_i(P_1, P_2, \ldots, P_n)$ and $\gamma_j(Q_1, Q_2, \ldots, Q_m)$ don't always exist and include all items in the bracket.

Definition 2.1 leads to the introduction of two notions of support.

Definition 2.2. The object-based support of an itemset X in a set of images S denoted by ob_support(X) is the percentage of objects in all images S. The image-based support of an itemset X in a set of images S denoted by ib_support(X) is the percentage of images with the same attributes in all images S. The confidence of an association rule X→Y is the ratio ib_support(X∧Y)/ib_support(X) if X or Y includes the attribute of image. Otherwise, the confidence is the ratio ob_support(X∧Y)/ob_support(X).

Definition 2.3. An itemset x is sufficiently frequent in a set S if the support of x is no less than its corresponding minimum support threshold σ', and no more than its corresponding maximum support threshold Σ'.

Definition 2.4. An association rule X→Y in a set of images S is sufficiently strong if X and Y are sufficiently frequent (X and Y ∈ [σ'...Σ']) and the confidence of X →Y is greater than φ'.

We propose two supports based on object and image because an object maybe appears repeatedly in an image but image is unique. The different supports mean different counting and different interestingness. We will give an example to describe this difference in next subsection.

3.2 Discovering Frequent Item-Sets

The reader is asked to read this subsection carefully as it provides the motivation for the algorithm we will describe later. Several medically important association rules are discussed as well as rules which are not interesting. To simplify our discussion, we will use an abstract example where images are transactions of objects and the same objects can repeat. While objects are multidimensional, in this discussion we will treat them as items with only one dimension. This example also includes one descriptor of the image that is either N(normal) or A(abnormal).

Example: Let us consider the images represented in Table 2 by a set of transactions S_1. Each image with a related description is a set of objects that can repeat. At this point, we ignore the descriptors of the objects for simplicity. To determine ob_support of each object and ib_support of the description, a first scan of the database is done and each time a distinct object and description appear, their counters are incremented. Table 3 shows the result of the counting. C_1 contains all unique objects with ob_support and descriptions with ib_support. Notice that ob_support is far more than ib_support because of the repetition of the same object.

Table 2. Image transaction table S_1

Image ID	Objects	Description
IM_1	$\{O_1, O_1, O_2\}$	A
IM_2	$\{O_1, O_2, O_2\}$	A
IM_3	$\{O_1, O_1\}$	N
IM_4	$\{O_1, O_1, O_1,O_2\}$	A
IM_5	$\{O_1, O_1, O_2,O_2\}$	N

Table 3. Candidate item table C_1

Object	Ob_support	Description	Ib_support
$\{O_1\}$	10	{A}	3
$\{O_2\}$	6	{N}	2

For simplicity, the support is expressed in an absolute value. Let the minimum support of object min_ob be 3 and the maximum support max_ob be 6. Let the minimum support of description min_ib be 2 and the maximum support max_ib be 4. Frequent k item-sets can be found using min_ob and min_ib by filtering the non-frequent k-1 item-sets. However, pruning with max_ob and max_ib to find sufficiently frequent item-sets should be left to the end of the process since too frequent k-1 item-sets may end up frequent enough at k level. Table 3 also shows F_1, the list of frequent 1 item-sets. Notice that O_1 was not eliminated even if it appears too often in the data set (ob_support(O_1) > max_ob). Given F_1, we can filter out from S_1 all irrelevant objects, and all transactions that do not contain frequent objects present in F_1. Table 4 shows S_2, the image transactions with only the interesting objects and descriptions. Here, all the transactions are reserved and even though either of objects and descriptions is not frequent, we will still keep the priority of frequent itemset.

Table 4. Filtered image transaction table S_2

Image ID	Objects	Description
IM_1	$\{O_1, O_1, O_2\}$	A
IM_2	$\{O_1, O_2, O_2\}$	A
IM_3	$\{O_1, O_1\}$	N
IM_4	$\{O_1, O_1, O_1,O_2\}$	A
IM_5	$\{O_1, O_1, O_2,O_2\}$	N

Table 5. Candidate 2 item-sets table C_2

2 itemsets	Ob_support
$\{O_1, O_1\}$	4
$\{O_1, O_2\}$	4
$\{O_2,O_2\}$	2
2 itemsets	Ib_support
$\{O_1, A\}$	3
$\{O_1, N\}$	2
$\{O_2, A\}$	3
$\{O_2, N\}$	1

Table 6. Sufficiently frequent 2 item-sets F_2

Frequent 2 item-sets	Ob_support
$\{O_1, O_1\}$	4
$\{O_1, O_2\}$	4
Frequent 2 item-sets	Ib_support
$\{O_1, A\}$	3
$\{O_1, N\}$	2
$\{O_2, A\}$	3

Table 7. Sufficiently frequent 3 item-sets F_3

3 itemsets	Ob_support
$\{O_1, O_1, O_2\}$	3

3 itemsets	Ib_support
$\{O_1, O_1, A\}$	2
$\{O_1, O_1, N\}$	2
$\{O_1, O_2, A\}$	3

The generation of the candidate 2 item-sets is done by joining F_1 with itself to create all possible pairs with frequent objects and descriptions. It is similar to the apriori algorithm in [6] except replication of objects in images and pruning according two different type of support. Table 5 shows all the 2 item-sets. After filtering the infrequent 2 item-sets, F_2 is produced, see table 6. The candidate 3 item-set list C_3 is produced by joining F_2 elements. In table 7, we can notice that there are not infrequent 3 item-sets and most items include repetition objects. At the same time, since ib_support is not equal to ob_support, the number of the item-sets containing description is more than that of not containing it. After filtering the infrequent 3 item-sets, F3 is produced. The candidate 4 item-sets is produced the same way by joining the frequent 3-item sets and pruning the unnecessary ones. For instance, since $\{O_1,O_2,O_2\}$ are not in F_3, $\{O_1,O_2,O_2, A\}$ are eliminated. Finally, as no 5 item-set can be induced,

Table 8. Candidate 4 item-sets F_4

4 itemsets	Ob_support
$\{O_1, O_1, O_2, A\}$	2

Table 9. Sufficiently frequent 4 item-sets F_4

4 itemsets	Ob_support
$\{O_1, O_1, O_2, A\}$	2

the result is all F_i without their item-sets that have a support higher than the maximum support max_ob or max_ib.

Here we summarize the main difficulties we have isolated so far trying to discover frequent item-sets in the medical images.

Association size. Associations and rules that involve many items are hard to interpret and can potentially generate a very high number of rules. And further, they slow down the interactive process by the user. Therefore, there should be a default threshold for association size. The biggest size of found associations is a practical bottleneck for algorithm performance. If for a given support the k-itemset X is frequent then all $Y \neq \Phi$ and $Y \subset X$ are frequent and then there are $O(2^k)$ frequent item-sets included. It is easy to see that no matter how efficient the algorithm is, the approach above will be slow for a large k.

Associations having uninteresting combinations of Items. This is the case where certain combinations are known to be trivial or have such a high support that do not really tell something new about the data set. Consider items x_i and x_j, if the association $X_1=\{x_i, x_j\}$ is not interesting then any other association $X_2 \supset X_1$ will not be interesting. Therefore, many of the items can be grouped by the domain expert to discard uninteresting associations. We assume small groups can be identified by domain knowledge.

With all the above problems, we propose the following algorithm. We introduce a function group(), which comes from domain knowledge, to constraint the number of frequent item-sets to be discovered. Let $C_k = \{O_1, O_2, \ldots, O_k\}$ be a set of items to be mined. We define group(O_i) to be an integer t if the item O_i belongs to a group. The constraint for discovering frequent item-sets was set as follows. Since the brain is composed of the left and right hemisphere, we set group(O_i) = 0 if the object O_i is only in the left hemisphere, 1 if the object O_i is only in the right, and 2 if the object O_i bestrides both brain hemispheres. Let *imax* be the maximum number of items appearing in one rule.

Algorithm 1 (SFIMI). Find sufficient frequent item-sets based on two supports in medical images.

Input: (i) S_1 a set of transactions without relationship predicate; (ii) the maximum and minimum object-based support and image-based support thresholds max_ob, min_ob, max_ib and min_ib;

Output: Sufficiently frequent item-sets.

```
(1)  $C_1 \leftarrow$ {candidate 1 item-set};
(2)  $F_1 \leftarrow$ {frequent 1 item-set from $C_1$};
(3)  k = 2;
(4)  while ( $k \leq imax$ ) {
```

```
(5) Extend $F_{k-1}$ by one item belonging to any $F_{k-1;}$
    let $C_k = \{O_1, O_2, \ldots, O_k\}$;
    if (group($O_i$) $\neq$ group($O_j$), for $i \neq j$ and $1 \leq i, j \leq k$
    then $C_k$ is a candidate;
(6) Check support of all candidate $C_k$ making one pass over the transactions;
    let X be item-sets only containing items about objects;
    let Y be item-sets containing items about images;
    $F_k \leftarrow \{c \in C_k \mid (c \in X \wedge (ob\_support(c) \geq min\_ob))$
        $\vee (c \in Y \wedge (ib\_support(Y) \geq min\_ib))\}$;
(7) k=k+1;
(8) }
(9) Result$\leftarrow \cup_k\{c \in F_k \mid (k>1) \wedge ((c \in X \wedge (ob\_support(c) \leq max\_ob))$
        $\vee (c \in Y \wedge (ib\_support(Y) \leq max\_ib)))\}$;
```

Line 1-3 are done in the same initial scan. In line 5, the candidate item-sets are discovered by joining (k-1) frequent item-sets. We use function group() to prune away the item-sets that probably are frequent item-sets. In line 6, only the frequent item-sets that are higher than the minimum support are kept. It is only at the end of the loop (line 9) that maximum support is used to eliminate item-sets that appear too frequently.

Lemma 1. Itemset interestingness is anti-monotonic in group(O_i) constraints.

Lemma 2. The group(O_i) constraints can be used to prune away association.

Lemma 3. Let F be a frequent k itemset. Assume *imax* < k then there are $2^k - \binom{k}{imax} 2^{imax}$ pruned associations.

3.3 Generating Association Rules

Most papers published in the database literature concentrate on optimizing the above phase, but few to improve rule generation. For mining association rule in medical images, there exist the following problem.

Items can appear only in the antecedent, only in the consequent or in either. Note that given the interesting rule $X \rightarrow Y$ no matter where an item appears the association $X \cup Y$ must be a frequent itemset, but where the item appears prunes out many uninteresting rules. In other words, support is still needed to prune uninteresting associations but confidence is not enough to prune out uninteresting rules because there may be many rules having high confidence containing forbidden items in the antecedent or in the consequent. Therefore items need to be constrained to appear in a specific part of the rule.

With this requirement, we propose the following algorithm. Let $F_1, F_2, \ldots, F_M$ be all frequent item-sets obtained in the above algorithm (SFIMI). We propose a constraint function cons() to eliminate the rules to be mined. We define cons(f_i) to be 1 if the item f_i can appear either in the antecedent or in the consequent, and 2 if it can appear only in the consequent of the rule.

Algorithm 2 (GAR). Generate association rules from the sufficient frequent item-sets.
Input: (i) Sufficient frequent item-sets F_1, F_2, …, F_M; (ii) minimum confidence;
Output: Association rules.

```
(1) for m=1 to M {
(2) for all non-empty subsets of $F_m$ {
(3)    Let $f_j$, $f_k \in F_m$;
       if ($f_j \cap f_k = \Phi$) and (cons(i) $\neq 2\ \forall$ i $\in f_j$) and (cons(i) $\neq 1\ \forall$ i $\in f_k$)
           and (support($f_j \cup f_k$)/support($f_j$) $\geq$ minconfidence)
       then $f_j \rightarrow f_k$ is valid;
(4)   }
(5) }
```

Lemma 4. Let F be a frequent k itemset. Assume m items are conf(f_i) constrained . Then the upper limit of discarded rules are ($C_k^1 + C_k^2 + ... + C_k^{k-1}$)-($C_{k-m}^1 + C_{k-m}^2 + ... + C_{k-m}^{k-m}$).

4 Experiments

The goal of the following experiment was to relate CT to brain tumor to validate actual diagnosis rules used by medical experts. In this case the purpose was mainly to confirm the validity of medical knowledge and tried to find new rules.

The dataset utilized in our experiments was real data from hospital. The main reason why we study on real brain CT images instead of any simulative data is to avoid insignificance and uninterestingness and the reliability of the discovered knowledge. To have access to real medical images is a very difficult undertaking due to legally privacy issues and management of hospital. But with some specialists' help and support, we got 618 precious images and their corresponding diagnosis records which, for simplicity, we generalized to normal(N) and abnormal(A).

The following is the constraint for generating association rules. For the descriptions of the images, like diagnosis record, they were constrained to appear in the consequent of the rule, that is, cons(i)=2, because it is more significant not only to medical doctors but patients. Objects with their attributes were constrained to appear concurrently either in the antecedent or in the consequent, i.e. cons(i)=1. It is evidently unmeaningful that an object is in the consequent and its attribute is in the antecedent.

We ran experiments with both of two max support = 100% and association rule size *imax* = 6. We varied minimum confidence and minimum support of object-based and image-based to get rules. The program can't be run without constraining and maybe the excessive rules will be generated. Table 10 shows our experiment results.

Table 10. Experiment results with medical images

Min_ob	Min_conf_o (object-based)	Min_ib	Min_conf_i (image-based)	the number of the rules
0.3	0.6	0.2	0.6	103
0.4	0.5	0.3	0.5	72
0.5	0.4	0.4	0.4	13

When the minimum support was lower, many of these rules were not interesting. But when it was higher, most of them were interesting and even important. We will show two rules that validate the actual diagnosis and domain knowledge.

Rule 1: $2O_1 \wedge 2O_2 \wedge \text{symmetry}(O_1,O_1) \wedge \text{symmetry}(O_2,O_2) \rightarrow N$ [90%]
Rule 2: $1O_1 \wedge 1O_2 \wedge \text{homolateral}(O_1,O_2) \rightarrow 2O_1 \wedge \text{homolateral}(O_1,O_1) \wedge A$ [60%]

O_1 and O_2 in rule 1 and rule 2 are light region and dark region respectively. The integer is the number of the objects repetition. Relationship predicate symmetry() represents that two objects are symmetrical and homolateral() represents that two objects are concurrently in the left or right hemisphere. Rule 1 describes that if there are two light regions and two dark regions in a brain image and they are symmetrical for a pair of regions with similar intensity, then this image belongs to a normal person. Rule 2 describes that if there are one light region and one dark region in one side of the brain image, then there are two dark regions in another side of the brain image and this image belongs to an abnormal person. With constraint function group() and cons(), we eliminate many rules like $2O_1 \rightarrow 1O_2$, where O_1 and O_2 are in the same side. Many previous works will generate this kind of rules that is not practically significant.

In any case fairly large medical data sets exist but they are not available to us. Also, it would be interesting to apply these ideas in other domains where large complex data sets are available.

5 Conclusion

In this paper, we have extended the concept of association rule using object, object-based attribute and relationship predicate, and image-based attribute in medical images. Two different supports based on object and image mean different counting and different interestingness. To detect objects in medical images, we have presented a progressive water immersion algorithm with guidance of domain knowledge to preprocessing the medical image set. Then we proposed two new algorithms with guidance of domain knowledge to discover frequent item-sets and mine interesting association rules from medical images. We have described the problem with a general form to provide a common framework for other problems appeared in other domains. Some interesting results were obtained by our program and solved the excessive uninteresting rules to some extent.

References

[1] U. M. Fayyad, S. G. Djorgovski, and N. Weir. Automating the analysis and cataloging of sky surveys. In U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, editors, Advances in Knowledge Discovery and Data Mining, pages 471–493. AAAI/MIT Press, 1996.

[2] P. Stolorz, H. Nakamura, E. Mesrobian, R. Muntz, E. Shek, J. Santos, J. Yi, K. Ng, S. Chien, C. Mechoso, and J. Farrara. Fast spatio-temporal data mining of large geophysical datasets. In Proc. Int. Conf. on KDD, pages 300–305, 1995.

[3] O. R. Zaiane, J. Han, Z.-N. Li, J. Y. Chiang, and S. Chee. MultiMediaMiner: A system prototype for multimedia data mining. In Proc. ACM-SIGMOD, Seattle, 1998.

[4] O. R. Zaiane, J. Han, Z.-N. Li, and J. Hou. Mining multimedia data. In CASCON'98: Meeting of Minds, Toronto, 1998.

[5] Pan Haiwei, Jianzhong Li, Zhang Wei. Classification of Medical Brain Images based on pixel's clustering. NDBC, 2002.

[6] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. VLDB, pages 487–499, 1994.

[7] WYNNE HSU, MONG LI LEE, JI ZHANG. Image Mining: Trends and Developments. Journal of Intelligent Information Systems, 19:1, 7–23, 2002.

[8] Wynne Hsu, Mong Li Lee, Kheng Guan Goh. Image Mining in IRIS: Integrated Retinal Information System. Proceedings of the ACM SIGMOD, May 2000, Dellas, Texas, U.S.A., pp. 593.

[9] Osmar R. Zaiane, Jiawei Han, Hua Zhu. Mining Recurrent Items in Multimedia with Progressive Resolution Refinement. ICDE 2001.

[10] Ordonez, C. and Omiecinski, E. (1999). Discovering Association Rules Based on Image Content. In IEEE Advances in Digital Libraries Conference.

[11] Maria-Luiza Antonie, Osmar R. Zaiane, Alexandru Coman. Application of Data Mining Techniques for Medical Image Classification. Proceedings of the Second International Workshop on Multimedia Data Mining (MDM/KDD'2001).

[12] Osmar R. Zaiane, Maria-Luiza Antonie, Alexandru Coman. Mammography Classification by an Association Rule-based Classifier. Proceedings of the Third International Workshop on Multimedia Data Mining (MDM/KDD'2002).

[13] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In Proc. 1994 Int. Conf. Very Large Data Bases (VLDB'94), pages 487-499, Sept. 1994.

[14] Gao Cong, Anthony K. H. Tung,Xin Xu, Feng Pan,Jiong Yang.FARMER: Finding Interesting Rule Groups in Microarray Datasets. In Proc. 2004 ACM-SIGMOD Int. Conf. Management of Data (SIGMOD'04).

Hybrid Feature Ranking for Proteins Classification

Ricco Rakotomalala[1], Faouzi Mhamdi[2], and Mourad Elloumi[2]

[1] ERIC Laboratory - University of Lyon 2, France
Ricco.Rakotomalala@univ-lyon2.fr
[2] URPAH – University of Tunis, Tunisie
Faouzi.Mhamdi@ensi.rnu.tn
Mourad.Elloumi@fsegt.rnu.tn

Abstract. Hybrid feature ranking is a feature selection method which combines the quickness of the filter approach and the accuracy of the wrapper approach. The main idea consists in a two steps procedure: building a sequence of feature subsets using an informational criterion, independently of the learning method; selecting the best one with a cross-validation error rate evaluation, using explicitly the learning method. In this paper, we show that in the protein discrimination domain, few examples but numerous descriptors, compared to a traditional approach where each descriptor is evaluated separately in the first step, to take account of their redundancy in the construction of candidate subsets of features reduces the size of the optimal subset and improves, in certain cases, the accuracy.

1 Introduction

The nature of the feature selection process has changed considerably. Previously, works in machine learning concentrated on the research of the best subset of features for a learning classifier, the number of features was rather reduced and the computing time was not a major constraint. Today, it is common to treat dataset comprising thousands of descriptors. This is particularly true in unstructured data processing where features are generated automatically from the original format of the data. In our case, a protein discrimination process from their primary structures, a protein is in the beginning described by a succession of characters representing amino acids, it is not possible to run directly a learning algorithm. We then generated a Boolean attribute-value table by checking the presence or absence of 3-grams (a sequence of consecutive 3 characters) for each protein. Because there are 20 kinds of characters (amino acids), we can produce 8000 descriptors. Consequently, the problem of feature selection always consists in finding the most accuracy subset of descriptors but by introducing a new strong constraint: the computing time must remain reasonable.

If we set aside the methods which embed automatically a feature selection during the learning process such as decision trees, there are two main categories

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 610–617, 2005.

of feature selection methods. "Wrapper" explicitly uses the characteristics of the learning algorithm to select the best subset of features, it uses cross-validation to compare the error rate of the candidate subsets [1]. Even if it can be very powerful in many cases, it presents two major disadvantages: it requires orders of magnitude more computation time because the learning process is repeatedly called; the repeated use of cross-validation on a single dataset can lead to growth of the probability of finding solutions that performs well on the validation data by chance alone. This approach is not feasible in the protein classification domain where there is a few examples - cross-validation can overfit - and numerous features - computation is not tractable.

"Filter" methods are at the opposite. They consist in selecting in an independent way, with ad hoc criterion, the best subset of features. Very efficient approaches based on correlation measurements were developed [2], they make it possible to treat quickly a dataset containing a considerable number of features. In spite of its recognized qualities, theses approaches are based nevertheless on a strong conjecture which is not always well controlled: the subset of features selected by the filtering method would be most powerful whatever the characteristics of the learning algorithm implemented. Although the empirical studies showed that this assertion seems judicious, it would be interesting to connect the solutions proposed by filter approach with the characteristics of the learning algorithm in order to determine the "best" subset.

In this paper, we experiment a technique in which we try to combine the advantages of the two approaches described above. The selection process is carried out in two steps: initially, we use a feature ranking framework i.e. we rank the descriptors according to correlation measurements; in the second time, we are based on this ordering to evaluate subsets of features in increasing size using a cross-validation error rate evaluation. This strategy named FS-ORDERED is not new [3] but in this approach, feature ranking step evaluates the relevance of each feature independently, thus leaving potentially redundant features. The main characteristic of the work which we present here is that we take account the redundancy of the features when we build successive subsets of features. Thus, if the features most correlated with the class attribute (protein family) were duplicated 9 times, instead of to hold the 10 first places, only the first one would be well classified, owing to the fact that they are strongly correlated with those that were already introduced, the others will occupy less advantageous positions and they will be preceded by the features which bring additional information to the already selected features. Our hope is to produce successive subsets of features as orthogonal as possible and thus obtain a smaller subset of features with a not-degraded accuracy compared to traditional hybrid feature ranking.

In the following section, we present the correlation measurement used to evaluate the degree of dependence between discrete attributes. Then, we will present an approach which allows to build successive subsets with non-redundant features. In the section 3, we describe our protein classification problem, and present results. The last section concludes the papers.

2 Correlation-Based Feature Ranking

2.1 Correlation Measures

A discrete feature X is relevant for the prediction of a discrete class attribute Y if for one value of X, we can associate one and only one value of Y. In other words, knowing the value of X allows to reduce uncertainty on the value taken by Y. Among the numerous measures which allow to express the feature relevance, we choose correlation measures.

If correlation measures are well known for continuous attributes, there are several interpretations with regard to the correlation between discrete attributes. In this paper, we choose an information theoretical interpretation and used a normalized information gain measure which is not biased in favor of features with more values. In the protein classification process where we have only binary attributes, this property is not significant, that appears essential if we want to apply this approach to other domains.

The symmetrical uncertainty between two discrete attributes Y and X is defined as

$$SU(Y,X) = 2 \times \left[\frac{H(Y) - H(Y/X)}{H(X) + H(Y)} \right]$$

where $H(Y)$ is the standard Shannon entropy; and $H(Y/X)$ is the conditional entropy i.e. the entropy of Y after observing values of X.

If X is not relevant in the prediction of Y, $SU(Y,X) = 0$, on the other hand, $SU(Y,X) = 1$ if the knowledge of X completely predicts the value of Y. This measure is very popular in the correlation based feature selection [2].

2.2 Combining Standard Feature Ranking and Wrapper Evaluation

The main pitfall of the correlation-based feature selection is the determination of the stopping rule in the selection process. The relevance of features being evaluated independently of the characteristics of the learning method, it is very hard to determine the optimal size of feature subset. If it is too restrictive, available information will be insufficient and the produced classifier not very powerful; if it is too permissive, irrelevant features can involve and degrade also the accuracy of the classifier.

Hybrid feature ranking approach allows to solve this pitfall. It proposes to benefit from the quickness of building the successive subsets of features by computation and sorting according to the correlation measurements, while using thereafter a cross-validation accuracy evaluation of candidates features subsets which takes account the characteristics of the implemented classifier [3].

Standard hybrid feature ranking (SHFR) can be described as following: (a) compute the relevance of features using a correlation measurement; (b) sorting the features according their relevance; (c) evaluate the subsets of features of increasing size using a cross-validation; (d) until we have tested all candidate subsets. Let us note that the suggested solutions are overlapping, the subset evaluated at the step j contains the $(j-1)$ features of the preceding solution.

The number of times where we call the learning process is known in advance, if J is the number of subsets and we evaluate the error rate with 10-fold cross-validation, we call the learning method $(J \times 10)$ times. It is also possible to introduce a parameter which limits the number of solutions to be tested [4].

2.3 Avoiding Redundancy in Hybrid Feature Ranking

Standard hybrid feature ranking (SHFR) allows us to avoid a hard fine tuning parameter for determination of the right size of the optimal subset of features. On the other hand, because we explicitly use the classifier characteristics and an unbiased error rate evaluation to detect the right subset, we can hope that the found solution will be powerful. Compared to wrapper approach, because we have dramatically reduced the hypothesis spaces, we can prevent overfitting. In a previous work, we implemented this solution in the protein classification and it gave very encouraging results [5].

Nevertheless, this approach suffers of an important weakness, it does not take account of the redundancy of the features. Thus, if we duplicate 9 times the most relevant feature, the first ten places will be occupied by the same feature. To illustrate this problem, in the protein classification which we will describe below, we choose a 3-grams descriptor which seems a good compromise in the majority of cases [5]. But for one family (”Tool like receptor” family), the best descriptor is in fact a 4-grams ”LDLS”, the 3-grams ”LDL” and ”DLS” which are selected by SHFR procedure are both relevant but highly redundant, one of them is sufficient to obtain an accurate classifier. It is thus necessary to introduce a new constraint in the construction of successive feature subsets: we add a feature if it is relevant to the class attribute, but is not redundant to any other already selected features.

In this paper, we used the CFS correlation based feature selection framework [2]. The first selected feature is always the most correlated feature with the class attribute. But, in the following step, we will classify in second position the feature which is at the same time the most correlated with the class and the least correlated with the already selected features. We use the sequential forward search version of CFS [6]. The real problem in this case is to define an indices which allows to evaluate the addition of a new attribute in the selected attribute by taking account of these compromise, or, it is similar, to directly evaluate the whole quality of a subset of features. Accordingly, CFS proposes the MERIT measure with the following formulae. Its advantage is that it makes possible to compare subsets of features of different sizes, and thus, to evaluate the contribution on an additional feature. It also enables to determine the optimal solution, it is the subset of features which maximizes the MERIT index. It is the approach adopted in the CFS method.

The MERIT of a subset $\aleph$ of j features is defined as

$$MERIT\left(\aleph\right) = \frac{j \times \overline{r_{YX}}}{\sqrt{j + j \times (j-1) \times \overline{r_{XX}}}}$$

where $\overline{r_{YX}}$ the average feature-class correlation, and $\overline{r_{XX}}$ the average feature-feature intercorrelation.

Contrary to the CFS filter algorithm, we removed this standard stopping rule in our hybrid approach. We evaluate each subset of features of increasing sizes, proposed by CFS, with a cross-validation error rate evaluation using the learning algorithm. Thus, we test all the sequences even if the MERIT measure decreases, our goal is to find the best subset of features which minimizes the error rate. We call this approach RBHFR (Redundancy-based Hybrid Feature Ranking), we hope that while eliminating as much as possible the redundancy between features in the successive subsets, the optimal subset which minimizes the error rate will be significantly small compared to standard hybrid feature ranking. Let us note that, compared to SHFR, RBHFR is slower in the first step when we rank the features, because we must compute the intercorrelation between candidate feature with the already selected features. But, in the second step, the number of error rate evaluation using the learning algorithm is the same.

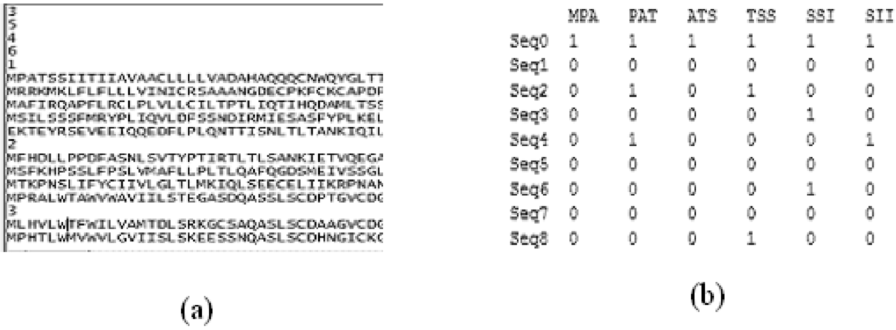
3 Experiments on a Proteins Classification Problem

3.1 The Proteins Classification Problem

In this paper, we use the text mining framework for a protein classification problem from their primary structures. The analogy with text classification is relevant in our case, indeed the original description of the dataset is very similar. A protein is described by a suite of characters which represents amino acids. There are 20 possible amino acids. We show below an example of file describing few proteins (Figure 1.a).

However, unlike the text classification, there is no "natural" separation in the character sequences, it is not possible to extract "words" for which we can easily attach semantics properties. Therefore, we have used the n-grams, a sequence of n characters, extraction techniques in order to produce descriptors.

Previous works showed that n = 3 (3-grams) is a good compromise to produce accurate classifier [5]. We obtain a Boolean attribute - value dataset with several thousands of descriptors (Figure 1.b). Theoretical maximum number of



```
3
5
4
6
1
MPATSSIITIIAVAACLLLLVADAHAQQQCNWQYGLTT
MRRKMKLFLFLLLVINICRSAAANGDECPKFCKCAPDF
MAFIRQAPFLRCLPLVLLCILTPTLIQTIHQDAMLTSS
MSILSSSFMRYPLIQVLDFSSNDIRMIESASFYPLKEL
EKTEYRSEVEEIQQEDFLPLQNTTISNLTLTANKIQIL
2
MFHDLLPPDFASNLSVTYPTIRTLTLSANKIETVQEGA
MSFKHPSSLFPSLVMAFLLPLTLQAFQGDSMEIVSSGL
MTKPNSLIFYCIIVLGLTLMKIQLSEECELIIKRPNAN
MPRALWTAWVWAVIILSTEGASDQASSLSCDPTGVCDG
3
MLHVLWTFWILVAMTDLSRKGCSAQASLSCDAAGVCDG
MPHTLWMVWVLGVIISLSKEESSNQASLSCDHNGICKC
```

(a)

	MPA	PAT	ATS	TSS	SSI	SII
Seq0	1	1	1	1	1	1
Seq1	0	0	0	0	0	0
Seq2	0	1	0	1	0	0
Seq3	0	0	0	0	1	0
Seq4	0	1	0	0	0	1
Seq5	0	0	0	0	0	0
Seq6	0	0	0	0	1	0
Seq7	0	0	0	0	0	0
Seq8	0	0	0	1	0	0

(b)

Fig. 1. Native description of proteins (a) – Boolean 3-grams attribute-value table (b)

3-grams for a protein classification problem is $20^3 = 8000$, experiments showed that we were close to this value. Many 3-grams are irrelevant, others also are redundant. The main challenge of the feature selection is to select among them the appropriate descriptors. There are several reasons for this dimensionality reduction: (1) machine learning algorithms work badly when the dataset is too sparse; selecting a subset of relevant features often improves the classifier performances; (2) the complexity of the learning algorithms always depends on the number of input features; the elimination of the useless attributes allows to a considerable improvement in computing time; (3) a reduced number of features provides a better understanding of the classifier.

3.2 Experiments

Five proteins families have been randomly extracted from the SCOP databank [7], the aim being to discriminate each pair of proteins. In our experiments, we use the naive bayes classifier. In spite of the assumption of conditional independence between attributes which seems erroneous, this method is often accurate [8]. The error rate of each subset of features is evaluated with 10-fold cross-validation.

In this paper, we compare the results of standard hybrid feature ranking (SHFR) and our proposal taking account the redundancy of the features (RBHFR). We use two criteria for our evaluation: the best error rate (ε_{best}) with the associated number of features; the average of the error rate obtained on the first 15 solutions suggested (ε_{avg})(Table 1[1]). The first criterion indicates the capacity of the approach to propose the most powerful solution, the constraint to choose overlapping successive subsets of features can be penalizing if features of bad quality - irrelevant or redundant – are involved in the first positions. The second criterion indicates the quickness of convergence towards a subset of good quality.

The results suggest some comments.

- To take into account the redundancy allows to find complementary features, leading towards powerful solutions quickly. Indeed, the average error rate on the 15 first subsets is always lower for RBHFR, whatever the family of protein studied is (ε_{avg}).
- About the optimal subsets, RBHFR systematically proposes the solutions at least as powerful in terms of error rate, and often reduced the size of this optimal subset (ε_{best} and feature subset size).
- There are two cases where our approach proposes a solution more accurate but with a higher number of descriptors (F_{24} and F_{45}). By studying the detailed results, we note that with a reduced number of descriptors, RBHFR are already more accurate than SHFR (0.024 with 4 features for F_{24}; 0.027 with 3 features for F_{45}).
- The CFS column of table 1 shows the size of the optimal subset computed with the standard CFS filter algorithm [2] which does not use an error rate

[1] We think that $\varepsilon_{best} = 0.000$ on some dataset is an artifact of the cross-validation error rate estimation, "the true error rate is rather very small" would be the correct assertion.

evaluation, and thus is not in relation with the characteristics of the learning algorithm. Results show that the standard filter stopping rule is very hard to adjust, especially in the context of very high number of features. For all discrimination, except the last one, CFS detects a wrong solution with too much features.

These results show that to take account the features redundancy when we build successive subsets of features which are evaluated with cross-validation leads to more accurate classifier and reduces the size of the optimal subset. The additional computing time introduced when we rank the features is justified.

To illustrate the behavior of the algorithm, we show the evolution of the error rate for the discrimination of proteins families F_{23} (Figure 2). The standard feature ranking SHFR is degraded initially by the addition of features which are highly redundant, the error rate is improved when the complementary descriptors are introduced, starting from the ninth feature. On the other hand, RBHFR converges

Table 1. Best error rate and associated feature subset size – Average error rate on 15 first subsets of features

	ε_{best}		Feature subset size			ε_{avg}	
Protein pair	SHFR	RBHFR	SHFR	RBHFR	CFS	SHFR	RBHFR
F_{12}	0.000	0.000	5	5	29	0.038	0.013
F_{13}	0.000	0.000	5	5	16	0.038	0.021
F_{14}	0.016	0.000	8	3	27	0.025	0.013
F_{15}	0.009	0.009	3	3	18	0.029	0.021
F_{23}	0.039	0.000	13	7	56	0.103	0.032
F_{24}	0.031	0.008	6	20	32	0.053	0.031
F_{25}	0.052	0.017	12	8	39	0.096	0.042
F_{34}	0.021	0.021	4	4	12	0.032	0.029
F_{35}	0.057	0.031	16	12	28	0.095	0.052
F_{45}	0.033	0.020	8	12	6	0.046	0.036

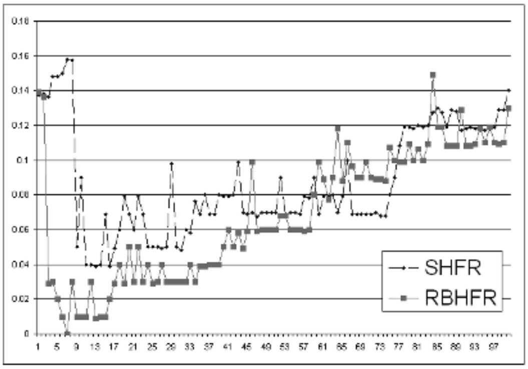


Fig. 2. Error rate evolution according to the size of feature subsets

quickly towards the optimal solution. We also notice that when the number of features is too high, upper 20 in the majority of our proteins families, the feature selection process is absolutely necessary if we want to obtain an accurate classifier.

4 Conclusion

Standard hybrid feature ranking combines the advantage of feature ranking - quickness - and wrapper - accuracy, adapted to the learning method characteristics. In this paper, we show that taking account the features redundancies when we rank the features improves the accuracy of the classifier and reduces the size of the optimal subset of features.

This approach seems a promising way in the context of proteins classification problems where we have few examples and numerous features: compared to classical wrapper approach, because we dramatically restrict the hypothesis spaces which we explore, we avoid overfitting; compared to classical filter approach, it is not necessary to proceed a fine adjustment of a parameter.

References

1. Kohavi, R., John, G.H.: Wrappers for feature subset selection. Artificial Intelligence **97** (1997) 273–324
2. Hall, M.A.: Correlation-based feature selection for discrete and numeric class machine learning. In: ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. (2000) 359–366
3. Xing, E.P., Jordan, M.I., Karp, R.M.: Feature selection for high-dimensional genomic microarray data. In: ICML '01: Proceedings of the Eighteenth International Conference on Machine Learning, Morgan Kaufmann Publishers Inc. (2001) 601–608
4. Duch, W., Wieczorek, T., Biesiada, J., Blachnik, M.: Comparison of feature ranking methods based on information entropy. In: Proceedings of International Joint Conference on Neural Networks (IJCNN), IEEE Press (2004) 1415–1420
5. Mhamdi, F., Elloumi, M., Rakotomalala, R.: Text-mining, feature selection and data-mining for proteins classification. In: Proceedings of International Conference on Information and Communication Technologies: From Theory to Applications, IEEE Press (2004) 457–458
6. Yu, L., Liu, H.: Efficiently handling feature redundancy in high-dimensional data. In: KDD '03: Proceedings of the ninth ACM SIGKDD, ACM Press (2003) 685–690
7. Murzin, A., Brenner, S., Hubbard, T., Chothia, C.: Scop: a structural classification of proteins database for the investigation of sequences and structures. Journal of Molecular Biology (1995) 536–540
8. Domingos, P., Pazzani, M.: On the optimality of the simple bayesian classifier under zero-one loss. Mach. Learn. **29** (1997) 103–130

Predicting Subcellular Localization of Proteins Using Support Vector Machine with N-Terminal Amino Composition

Yan-fu Li[1] and Juan Liu[1,2]

[1] International School of Software, Wuhan University, Wuhan 430072, China
[2] School of Computer, Wuhan University, Wuhan 430079, China
liujuan@whu.edu.cn

Abstract. Prediction of protein subcellular localization is one of the hot research topics in bioinformatics. In this paper, several support vector machines (SVM) with a new presented coding scheme method based on N-terminal amino compositions are first trained to discriminate between proteins destined for the mitochondrion, the chloroplast, the secretory pathway, and 'other' localizations. Then a decision unit is used to make the final prediction based on several SVMs' outputs. Tested on redundancy-reduced sets, the proposed method reached 89.6 % (plant) and 91.9% (non-plant) total accuracies, which, to the best of our knowledge, are the highest ever reported using the same data sets.

1 Introduction

Genome function annotation including the assignment of the function for a potential gene in the raw sequence is now one of the hot topics in bioinformatics and the subcellular location of a protein is closely correlated to its biological function [1]. High-throughput sequencing technology has made it possible for many laboratories to sequence the genomes and proteomes of new organisms. With the rapid increase of sequenced genomic data, the need for an automatic and accurate tool to predict protein subcellular localization becomes increasingly important. Therefore, a fully automatic and reliable prediction system for protein subcellular localization would be very useful.

There are mainly two categories of attempts having been made to automatically predict protein subcellular localization. One is based on amino composition [2, 3, 4]. Nakashima and Nishikawa [5] have pointed out the significant difference between intracellular and extracellular proteins in the amino acid composition. Based on their result, Reinhardt and Hubbard [3] built a prediction system using neural networks, which obtained a total accuracy of 81% for three subcellular locations in prokaryotic sequences and 66% for four locations in eukaryotic sequences. Hua and Sun [4] proposed a method using SVMs to achieve accuracy 91.4% for three subcellular locations in prokaryotic organisms and 79.4% for four locations in eukaryotic organisms. It is true that these methods have reported high total accuracy and acceptable robustness in N-terminal sequence. However, they are not good enough to predict some kinds of proteins. For instance,

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 618–625, 2005.

when used to predict the mitochondrion protein, the neural network method only achieved an accuracy of 61% and the SVM method obtained an even lower accuracy of 56.7%. Furthermore, it is pointed out by Nakai [6] that isoforms cannot be well localized by the methods based on composition.

The other attempts are focused on the recognition of protein N-terminal sorting signals [7, 8]. Emanuelsson and his colleagues have developed an integrated subcellular localization prediction system, called TargetP, with two layers of neural networks using N-terminal sorting information [7]. They dealt with plant and non-plant sequences respectively to obtain the total accuracies of 85% (plant) and 90% (non-plant). The accuracies of the prediction of mitochondrion protein are 82% for plant and 89% for non-plant, which show that the method based on N-terminal amino acid sequences may achieve higher accuracies than those depended on amino acid composition when it is used to predict mitochondrion protein. The advantage of this method is that it can imitate the real sorting process to a certain extent. However, the input sequences of this system do not contain much useful structural or compositional information. Moreover, the system with two layers of neural networks and a decision unit is more complicated than Subloc [4] and other methods based on amino acid composition.

To address above problems, we propose a novel approach for protein subcellular localization by combining the amino composition method with the N-terminal sorting signals method, the N-terminal amino composition method. On one hand, it can take the advantage of N-terminal sorting information that imitates the real sorting process; on the other hand, the input sequences also contain some compositional information. Similar to [4], we use several $1-v-r$ SVMs to build the prediction model. The SVMs are first trained to discriminate between proteins destined for different localizations. Then a decision unit is used to make the final prediction based on several SVMs' outputs.

2 Materials and Methods

2.1 Data Sets

We use the data sets established by Emanuelsson and his colleagues [7] to evaluate our method. All sequences were extracted from SWISS-PROT. The sequences belong to plant and non-plant proteins and are further divided into 5 categories: mitochondrion, the chloroplast, the secreted proteins, nucleus and cytosolic proteins. For the plant data sets, release 36 was used; for the non-plant data sets, release 37 was used, except that the upgrades of release 38 also were included for mTP set. Just as mentioned in [7], nuclear and cytosolic proteins were assembled together to form the "Other" category. After the redundancy reduction, the plant data set contained 368 mTP, 141 cTP, 269 SP, and 162 "Other"[1] sequences, and the non-plant data set contained 371 mTP, 715 SP and 1652 "Other" sequences.

[1] Abbreviations: SP, signal peptide; mTP, mitochondrial targeting peptide; cTP, chloroplast transit peptide.

2.2 Support Vector Machine

The SVM [9] is a novel machine learning method that has been widely used in many kinds of pattern recognition problems. It is based on the following concepts: mapping the input vectors into a high-dimension Hilbert space and establishing a separating hyperplane in this space. The mapping is performed by a kernel function defining an inner product in the high-dimension space. The separating hyperplane, called the optimal separating hyperplane (OSH), is selected to maximize its distance from the nearest training samples of each class. The kernel function is critical to the performance of SVM. Preliminary tests show that the Radial Basis Function (RBF) kernel can usually obtain better results. Therefore, the RBF kernel was used for all the experiments in our work.

2.3 Coding Scheme

Different from TargetP or SubLoc, both the classical amino composition coding method and protein N-terminal sorting signals coding method are utilized in our work. The input vector $V$of a k protein is defined by the following formula:

$$V = \begin{bmatrix} v_{1,k} \\ v_{2,k} \\ \vdots \\ v_{20,k} \end{bmatrix}$$

$$v_{i,k} = \frac{n_{i,k}}{N_k} (i = 1, 2 \cdots 20)$$

Where N_k is the total number of amino acid residues of the N-terminal sorting signals in protein k and $n_{i,k}$is the number of ith amino acid residue of the N-terminal sorting signals in protein k.

The length of N-terminal amino has great influence in the results. It has been mentioned in [10] that the typical length of cTP is from 40 to 70, mTP is from 25 to 45 and SP is from 20 to 30. In this paper, we simply take the averages and set the lengths to 55, 35, 25 for cTP, mTP and SP respectively.

As to the category “Other”, it contains nuclear protein sequences and cytosolic protein sequences. The nuclear localization signals (NLSs) do not generally show any particular consensus sequence, it is rather hard to discriminate a NLS from a non-NLS region [10]; and, to the best of our knowledge, the cytosolic protein does not contain special localization signals. So it is hard to determine the length of N-terminal amino for category “Other”. To avert this problem, during the training process, we first chose a certain range of the length (the upper limit roughly follows the average length of forepart of the sequence), then increased the length by adding one position every time to observe the variation of total accuracy. The length that made the SVM get the highest output value was accepted. According to the results shown in Fig.1. and Fig.2., the length of “Other” is set to 31 for non-plant set, 45 for plant set.

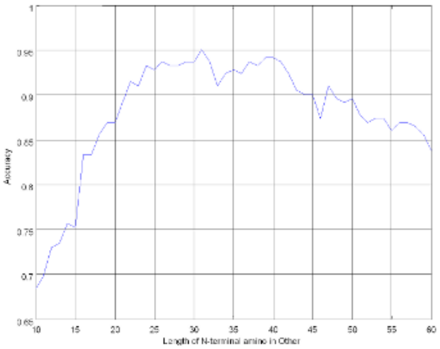


Fig. 1. The length of category "Other" chose in non-plant set is 31

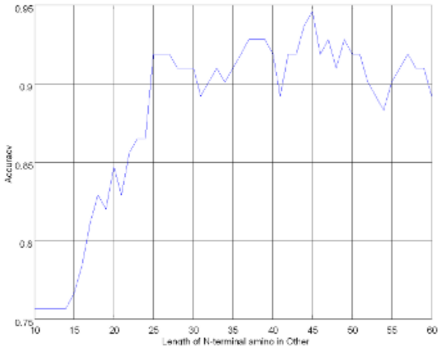


Fig. 2. The length of category "Other" chose in plant set is 45

2.4 Construction of Predicting System

Initially, SVMs can only deal with the binary classification problems. Recently, there are also several approaches having been proposed to solve multi-class problems, such as [11], which presents two methods of implementing multi-class SVM in one step [12]. compared several approaches of multi-class SVM, the results show that, when using the RBF kernel, $1-v-r$ method performed better than others in 4-class classification and almost the same with others in 3-class classification. Therefore, $1-v-r$ is selected in this work. Fig.3 illustrates the architecture of the predicting system. For a k-class classification, k SVMs are established. Each of them is called $1-v-r$ SVM. The ith SVM is trained to recognize the ith class. The decision unit finally determines the final class label of the test sample as the result of the $1-v-r$ SVM with the highest output value.

2.5 Measurement of Prediction Performance

In this paper, the sensitivity, the specificity and the total accuracy are used to evaluate the performance the methods:

$$Total\ accuracy = \frac{\sum_{i=1}^{k} tp(i)}{N}$$

$$sensitivity = \frac{tp(i)}{tp(i)+fn(i)}$$

$$specificity = \frac{tp(i)}{tp(i)+fp(i)}$$

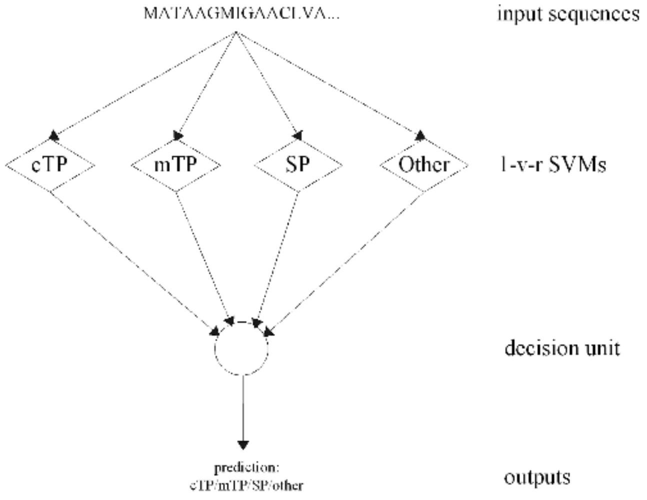


Fig. 3. The predictor architecture is built from one layer of 1-v-r SVMs and a decision-making unit. The non-plant version lacks the cTP SVM

Where N is the total number of proteins, k is the category number, $tp(i)$ is the number of true positives of location i, $fn(i)$ is the number of false negatives of location i, and $fp(i)$ is the number of false positives of location i. Moreover, the MCC (Matthews Correlation Coefficient) [13], defined by the following formula, is also used to measure the balance of the prediction among different classes.

$$MCC = \frac{tp(i) \times tn(i) - fp(i) \times fn(i)}{\sqrt{(tp(i) + fn(i)) \times (tp(i) + fp(i)) \times (tn(i) + fp(i)) \times (tn(i) + fn(i))}}$$

Where $tn(i)$ is the number of true negatives of location i.

3 Experiments and Results

We did experiments to evaluate our method and compared our results with those obtained from some popular subcellular localization prediction methods listed in [10]. Among which, iPSORT and TargetP are based on N-terminal signal method, Subloc is based on amino acid composition method. In order to be fair, we followed the procedures same as [10]. In our C=20, γ =80 in all SVMs.

3.1 Comparison with Other Methods

The comparison results for non-plant and plant data set are shown in Table 1. All data except for our method come from [10]. Among all, our method obtained the highest total accuracies for plant set 89.6% and non-plant set 91.9%. Also our method performed better than Subloc in terms of sensitivity and specificity for all sets. When compared to TargetP, our method performed better or as the same as it in most categories. Our method also performed better than iPSORT in terms of sensitivity and specificity for most sets.

These results indicate that with the N-terminal amino composition coding scheme, the prediction accuracy can be significantly improved by using the same machine-learning tool.

Table 1. Comparison of localization predictor performances on plant (940 proteins) and non-plant (2738 proteins) test sets

Predictor set	Total accuracy (%)	Category	Sensitivity	Specificity	MCC
A. This work					
Plant	89.6	cTP	0.99	0.75	0.84
		mTP	0.81	0.99	0.84
		SP	0.92	0.94	0.90
		Other (*Nuclear+Cytosolic*)	0.98	0.83	0.88
Non-plant	91.9	mTP	1.00	0.78	0.87
		SP	0.90	0.89	0.86
		Other (*Nuclear+Cytosolic*)	0.91	0.97	0.85
B. TargetP					
Plant	85.3	cTP	0.85	0.69	0.72
		mTP	0.82	0.90	0.77
		SP	0.91	0.95	0.90
		Other (*Nuclear+Cytosolic*)	0.85	0.78	0.77
Non-plant	90.0	mTP	0.89	0.67	0.73
		SP	0.96	0.92	0.92
		Other (*Nuclear+Cytosolic*)	0.88	0.97	0.82
C. iPSORT					
Plant	83.4	cTP	0.68	0.71	-
		mTP	0.84	0.86	-
		SP	0.91	0.98	-
		Other (*Nuclear+Cytosolic*)	0.83	0.70	-
Non-plant	88.5	mTP	0.74	0.68	-
		SP	0.96	0.92	-
		Other (*Nuclear+Cytosolic*)	0.90	0.92	-
D. Subloc					
Non-plant	77.4	mTP	0.67	0.61	-
		SP	0.50	0.74	-
		Nuclear	0.84	0.79	-
		Cytosolic	0.64	0.46	-

3.2 Reliability Index of the Prediction

It is important to know the prediction reliability of a machine learning method. The Reliability Index (RI) is a useful indicator of the level of the certainty in the prediction for a sequence, and is usually assigned according to the difference (noted asΔ) between the highest and the second output value [3, 7, 14]. In this paper, The RI is calculated by the following formula:

$$RI = \begin{cases} INTEGER(\frac{\Delta}{0.6}) + 1 & if\ 0 \leq \Delta \leq 2.4 \\ 5 & if\ \Delta > 2.4 \end{cases}$$

Table 2 shows the statistical results for non-plant data set. Similar results were obtained for plant data set (not shown in this paper). For example, the expected accuracy for a sequence with $RI = 3$ is 97.8%. About 80% of all sequences have $RI \geq 2$ and of these sequences above 94.7% were correctly predicted.

Table 2. The performance within the RI

RI	1	2	3	4	5
No. sequences with the RI	546	700	535	862	95
% of the sequences with the RI in all non-plant sequences	19.9%	25.6%	19.5%	31.5%	3.5%
No. correctly predicted sequences with the RI	379	663	523	858	92
% of correctly predicted sequences with the RI in sequences with the RI	69.4%	94.7%	97.8%	99.5%	96.8%

4 Conclusions and Discussion

In this paper, we combined amino acid composition coding method with N-terminal sorting signals coding method and proposed a novel N-terminal composition coding method. The aim of this work is to provide a subcellular localization predictor for proteins potentially destined to the chloroplast, the mitochondrion, or the secretory pathway. Comparing to widely used tools, We have managed to increase the discriminant ability between target sequences, especially in terms of sensitivity. In particular, the poor discrimination of cTPs and mTPs has been clearly improved when compared to TargetP and iPSORT(Table 1). Furthermore, the statical results shown in Table 2 illustrates that the prediction results of our method are reliable.

In conclusion, the work in this paper demonstrates that, with protein sorting signals composition, a subcellular localization predictor for proteins with reasonable reliability can be constructed only from amino acid sequence alone. Current research results show that it is likely that other useful features can also be incorporated into the predictor. For example, Bhasin and Raghava [15] have localized eukaryotic proteins using various features of proteins, such as physicochemical properties, amino acid composition, and dipeptide composition. In the future, other useful information will also be addressed in our method.

Acknowledgements

This work was supported by the National Nature Science Foundation of China under grant 60301009, Chenuang Project of Wuhan city under grant 211121009, and the Open Foundation of State Key Laboratory of Software Engineering of Wuhan University under grant SKLSE03-09. We thank Dr. Emanuelsson for providing us with the experiment data, thank Dr. Liu Gai for his valuable suggestions.

References

1. Jensen, L.J., Gupta, R., and Blom, N., et al.: Prediction of human protein function from post-translational modifications and localization features, J. Mol. Biol. **319** (2002) 1257–1265
2. Chou,K.C. and Elrod,D.: Protein subcellular location prediction, Protein Eng. **12** (1999) 107–118
3. Reinhardt,A. and Hubbard,T: Using neural networks for prediction of the subcellular location of proteins. Nucleic Acids Res **26** (1998) 2230–2236
4. Hua S. and Sun Z.: Support vector machine approach for protein subcellular localization prediction, Bioinformatics **17** (2001) 721–728
5. Nakashima,H. and Nishikawa,K.: Discrimination of intracellular and extracellular proteins using amino acid composition and residue-pair frequencies, J. Mol. Biol. **238** (1994) 54–61
6. Nakai,K.: Protein sorting signals and prediction of subcellular localization, Adv. Protein Chem. **54** (2000) 277–344
7. Emanuelsson,O., Nielsen,H., and Brunak,S, et al.: Predicting subcellular localization of proteins based on their N-terminal amino acid sequence, J. Mol. Biol, **300** (2000) 1005–1016.
8. Nielsen, H., Engelbrecht, J., and Brunak, S., et al.: Identification of prokaryotic and eukaryotic signal peptides and prediction of their cleavage sites, Protein Eng. **10 (**1997)1–6
9. Vapnik,V.: The Nature of Statistical Learning Theory, Springer, New York (1995)
10. Emanuelsson, O.: Predicting protein subcellular localisation from amino acid sequence information, Briefings in Bioinformatics **3** (2002) 361–376
11. Westion, J. and Watkins, C.: Multi-class support vector machines, In: Verleysen, M. (ed.): Proceedings of ESANN99, Brussels, D. Facto Press (1999)
12. Hsu, C.-W. and Lin, C,-J.: A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks, **13** (2002) 415-425
13. Matthews,B.W.: Comparison of predicted and observed secondary structure of T4 phage lysozyme, Biochim. Biophys. Acta **405** (1975) 442–451
14. Rost,B. and Sander,C.: Prediction of secondary structure at better than 70% accuracy, J. Mol. Biol. **232** (1993) 584–599
15. Bhasin,M. and Raghava, G.P.S.: ESLpred: SVM Based Method for Subcellular Localization of Eukaryotic Proteins using Dipeptide Composition and PSI-BLAST, Nucleic Acids Reasearch **32** (2004) W383 - W389

The Dynamic Character Curve Adjusting Model of Electric Load Based on Data Mining Theory

Xiaoxing Zhang, Haijun Ren, Yuming Liu, Qiyun Cheng, and Caixin Sun

The Key Laboratory of High Voltage Engineering and Electrical New Technology
of Ministry of Education，Chongqing University，Chongqing 400044，China
mikezxx@tom.com

Abstract. There are a number of dirty data in the load database produced by SCADA system. Consequently, the data must be adjusted carefully and reasonably before being used for electric load forecasting or power system analysis. This paper proposes a dynamic and intelligent curve adjusting model based on data mining theory. Firstly the Kohonen neural network is meliorated according to fuzzy soft clustering arithmetic which can realize the collateral calculation of Fuzzy c-means soft clustering arithmetic. The proposed dynamic algorithm can automatically find the new clustering center, namely, the character curve of data, according to the updating of swatch data. Combining an RBF neural network with this dynamic algorithm, the intelligent adjusting model is introduced to identify the dirty data. The rapidness and dynamic performance of model make it suitable for real-time calculation. Test results using actual data of Jiangbei power supply bureau in Chongqing demonstrate the effectiveness and feasibility of the model.

1 Introduction

Load data from SCADA system has some dirty data because of information channel transmission error, RTU faults, impack load etc. Directly using these load data can influent the accuracy of electric load forecasting and power system analysis, thus it is necessary to adjust the dirty data [1].

Recently, various methods are proposed to recognize and adjust the dirty data from load data. In [2], the ranges of load changing rate in two points at one historical load day were counted. By comparing the ranges of two load changing rate to the normal ranges, dubious point could be found. However, if the first data was dirty or the dirty data existing in a long interval, it misjudged or failed to judge. Reference [3] proposed a method to the detection and identification of mal-data by combining gray estimation and parameter evaluation. Because the parameter evaluation was a complicated non-linear optimal problem, this approach was prone to be troubled by local extreme and less effective. In [4] artificial neural network was introduced to realize the identification and adjustment of dirty data, which was well conceived and could subtly locate dirty data. Meanwhile, it had two weak points: firstly the used Kohonen NN can only fulfill hard clustering of spherical data and the adjustment of dirty data by using hard clustering center vector as characteristic curve was coarse; secondly the swatch data for load forecasting should be dynamic and updating due to the elapsing time.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 626 – 633, 2005.

In this paper, a dynamic and intelligent model based on data mining theory is proposed which has three layers. The first layer will extract the characteristic curve from load using Kohonen neural network improved by fuzzy soft clustering arithmetic. RBF neural network; in the second layer, RBF neural network is used to construct a pattern classifier for dirty data; In the third layer, the value of dirty data can be adjusted which are replaced by the weighted sum of corresponding value in two characteristic curves with maximal membership grade.

2 Principle of Intelligent Adjusting Model of Dirty Data

Similarity and smoothness are the two important characters of power load curves. Several peak times in daily curve are generally the same and the neighboring points usually have not much variation, the existence of dirty data will obviously destroy the smoothness, but similarity remains unchanged because the amount of dirty data is small. Therefore, characteristic patterns can be extracted from many load curves that may contain the information of dirty data by using clustering algorithm in data mining theory. Then, the characteristic curve can be separated from the load curves by using classification algorithm. After this step, dirty data will be recognized. The structure of model is shown in Fig 1.

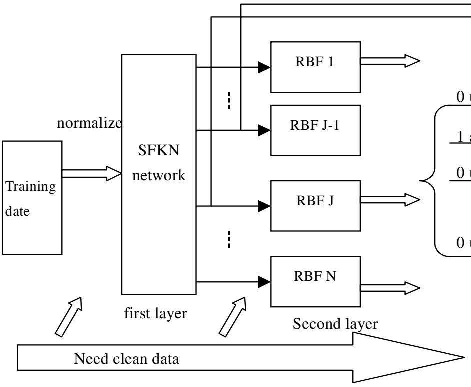


Fig. 1. The intelligent adjusting model of dirty data

The first layer is improved Kohonen network (SFKN-Soft Fuzzy Kohonen Network). The under checked curve $\boldsymbol{x}_{\mathrm{j}}$, is the input of SFNN. If the characteristic curve corresponding to a nerve cell has greatest similarity to $\boldsymbol{x}_{\mathrm{j}}$, the nerve cell will output 1 and excite the corresponding RBF sub-network. The second layer is RBF sub-network related to each clustering center. After training, it is ready to identify the dirty date and locate them accurately. If the output cell of RBF is close or equal to 1, the corresponding input cell stands for dirty data. The third layer adjusts dirty data. Detailed principles in term of model layer are as follows:

2.1 Load Data Clustering (The First Layer)

Data clustering is used for extracting the characteristic curve from load. The clustering analysis algorithm can be divided into two parts: hard clustering algorithm and fuzzy clustering algorithm [7]. The hard clustering algorithm has a fast convergence rate but is prone to fall into local extremum; while the later is more practical and accuracy.

2.1.1 Fuzzy C-Means Clustering [5]-[7](FCM)

FCM is the most widely used clustering algorithm for research and application. At first it gets an object function that is the weighted sum of distance between each data and clustering center (weight is the exponent of membership function), then the iterative formula of clustering center is obtained by calculating the optima of object function. $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_n\}$ denotes a data set , each element in $\boldsymbol{X}$ is a vector with P dimension, and C stands for the number of class. Center of the i-th class is expressed by $\boldsymbol{v}_i=\{v_{i1},v_{i2},\cdots,v_{ip}\}$. In $\boldsymbol{X}, u_{ij}$ and $d_{ij}^2 = \| \boldsymbol{x}_j - \boldsymbol{v}_i \|^2$ stand for membership grade and distance of the j-th element toward the i-th clustering center, respectively.

Then membership grade and class center of iterative expressions are:

$$\text{When } \boldsymbol{x}_j \neq \boldsymbol{v}_i \quad u_{ij} = \left(\sum_{k=1}^{c} \left(d_{ij}^2 / d_{kj}^2 \right)^{\frac{1}{m-1}} \right)^{-1} . \tag{1}$$

$$\text{Otherwise:} \quad u_{ij} = \begin{cases} 1, & j=i \\ 0, & j \neq i \end{cases} . \tag{2}$$

$$\boldsymbol{v}_i = \sum_{j=1}^{n} u_{ij}^m \boldsymbol{x}_j \left(\sum_{j=1}^{n} u_{ij}^m \right)^{-1} . \tag{3}$$

Where parameter m is ambiguity or smoothness factor, which control the share degree of patterns in different fuzzy classes. Usually, m ranges from 1.5 to 2.5.

2.1.2 Characters of Kohonen NN and the Fault of FCM

In Kohonen self-organization mapping network, neural cells interact and compete with each other by which simulate the human brain's ability of clustering, self-organization and learning, but, the Kohonen NN can only fulfill the hard clustering of spherical data. In Kohonen NN, the Mexico strawhat function is used in competition layer to suppress the nearby cells and excite the far cells.

Soft clustering means the adjustment of membership grade matrix by using extra information in the iterative process of fuzzy clustering. Adjusted matrix makes the selection of clustering center value more reasonable in the next iterative step and accelerates convergence of FCM algorithm. Therefore, formula (3) may not be reasonable because all the membership grade influence $\boldsymbol{v}_i$. Data should greatly adjust the clustering center that has the highest membership grade, and may not strongly effect the clustering center with sub maximal membership grade or it will slow the convergence of FCM. Meanwhile, data should slightly adjust the center if membership grade is small, and thus avoid dead point in clustering.

According to the side-restraining feature of Kohonen NN and its conformance with the views above, the paper propose SFKN network by using KN to realize FCM soft clustering, SFKN can quickly get center point of clustering and thus increases the convergence rate.

2.1.3 SFKN Network Combined with FCM and KN

Ambiguity and membership grade are introduced into the calculation of Kohonen's learning rate, Nodes near the cluster center are adjusted greatly, and the points far away have less adjustment. Weight updating formula can be expressed as:

$$\Delta w_{ij}(t) = \eta(t)_{ij}(\boldsymbol{x}_j - w_{ij}(t))\,. \tag{4}$$

Where learning rate:

$$\eta_{ij}(t) = (u_{ij}(t))^{m(t)}\,. \tag{5}$$

$$m(t) = m_0 - (m_0 - 1)t/T\,. \tag{6}$$

Where T is the maximal iterative steps; constant m_0 (m_0>1) is initial iterative ambiguity.

In FCM algorithm, all the membership grade of data will influence the clustering center which is unreasonable. Hence, membership grade matrix is amended through Kohonen's side-restraining feature. Membership grade of $\boldsymbol{x}_{\mathrm{j}}$ in the i-th iterative step can be calculated as:

$$u_{ij}^{*}(t) = u_{ij}(t)\left[\frac{1}{k}\left(k_1 e^{-\frac{(i-i^*)^2}{\sigma_1}} - k_2 e^{-\frac{(i-i^*)^2}{\sigma_2}}\right) + k_3\right]. \tag{7}$$

Where k , k_1 , k_2 , σ_1 , σ_2 , k_3 are constant. k is a normalized number, k_3 refers to weak excite strength for the point far away. Expression in square brackets is the Mexico strawhat function consisting two Guassian functions ($k_1 > k_2$, $\sigma_2 > \sigma_1$), i^* is the maximal matching point. Thus, influence of data toward sub nearest clustering center is refrained by amending the membership grade matrix, then convergence rate is accelerated and soft clustering of data is realized.

SFKN algorithm can simply be described as follows:

1) Choosing clustering number C and a small positive constant ε .
2) Initiating $\boldsymbol{V}_0 = (\boldsymbol{v}_1^0, \boldsymbol{v}_2^0, \cdots, \boldsymbol{v}_p^0)$, choosing m_0>1 and maximal iterative steps T.
3) Calculating membership grade matrix U and learning rate according to formula (1)、(2)、(5)~(7).
4) Updating weight vector $\boldsymbol{w}_i(t)$ by using formula (3)(4).
5) If $m(t) > 1.0$ 、 $|\boldsymbol{w}_i(t) - \boldsymbol{w}_i(t-1)| > \varepsilon$ and $t < T$, then t=t+1, turn to step 3, else clustering end.

2.1.4 Dynamic Soft Clustering by Using SFKN

The swatch data is a time sequence and should be updated dynamically with time elapsing. Thus, dirty data adjusting is also a dynamic process. In this paper, detective threshold value u_0 is introduced and the detailed algorithm are as follows:

1) Initializing the dynamic detective threshold value u_0.

2) Introducing $\boldsymbol{x}_j^+$ and $\boldsymbol{x}_j^-$, where $\boldsymbol{x}_j^+$ means the new added swatch data in data set and $\boldsymbol{x}_j^-$ stands for the eliminated swatch data, and current swatch data can be expressed as: $X = \{\{\boldsymbol{x}_j\} + \{\boldsymbol{x}_j^+\} - \{\boldsymbol{x}_j^-\}\}$.

3) Checking $u^*_{i(j-j^-)}$, membership grade of rest data towards the center vector of clustering which data was eliminated except the rest data. If $u^*_{i(j-j^-)} < u_0$ then eliminating v_i, updating c=c-1, and setting all the weights to 0 whose related nodes connect with this node; if $u^*_{i(j-j^-)} \geq u_0$, remain $\boldsymbol{v}_i$.

4) Calculating u_{ij^+} according to formula (1), where u_{ij^+} means the membership grade of $\boldsymbol{x}_j^+$ toward each clustering center $\boldsymbol{v}_i$. Setting $u^*_{ij^+} = \max\{u_{ij^+}\}$, if $u^*_{ij^+} < u_0$, then continue the next step 5; if $u^*_{ij^+} \geq u_0$ then algorithm end.

5) Introducing new added clustering center $\boldsymbol{v}_{i^+}$ with initial value $\boldsymbol{x}_j^+$, setting c=c+1 and keeping other clustering center unchanged, choosing $\boldsymbol{x}_j^+$ as the input of SFKN and calculating new clustering center according to SFKN algorithm.

In step 5, most of clustering centers remain unchanged though the network structure has some variation. At the same time, membership grade of original data toward these cluster centers also keep unchanged, so the parameters of original network can be used in new network, which will converges quickly.

2.2 Pattern Classifying of Dirty Data (The Second Layer)

In the second layer of structure, RBF is used to construct a pattern classifier for dirty data because of its strong ability of fast convergence and classification. Each clustering center corresponds to a RBFNN and the value of clustering center is selected as the center of Gaussian function in every RBF. Suppose that there is only single dirty data and the rest data are normal, and choose sampling number as 96, then the pattern number of dirty data is 96×2=192.

Creating process of I/O swatch data set can be described as:

1) Choosing clustering center $\boldsymbol{v}_i$ as the i-th input of RBFNN, that is, $x_0 = v_i$, and the corresponding output is $\boldsymbol{y}_0 = (0,0,\cdots,0)$.

2) Giving the first element of $\boldsymbol{v}_i$ a deviation, $\boldsymbol{x}_1 = (\boldsymbol{v}_i(1) + e, \boldsymbol{v}_i(2), \cdots, \boldsymbol{v}_i(p))$, and producing a sample containing dirt data, then output $\boldsymbol{y}_1 = (1,0,\cdots,0)$. Continuing this operation to the rest element of $\boldsymbol{v}_i$ and obtaining a swatch data set with positive deviation.

3) Changing the deviation e to $-e$, and replacing 1 in the output vector by –1. Repeat step 2 and obtain a swatch data set with negative deviation.

Network after training can identify and location dirty data accurately no matter how the dirty data existing in the curve: whether single or an interval.

2.3 Recognition and Adjustment of Dirty Data (The Third Layer)

The amendment of dirty data is realized in the third layer. The value of dirty data positioned in the second layer should be adjusted which are replaced by the weighted sum of corresponding value in two characteristic curves with maximal membership grade. If sub maximal membership grade is less than 0.2, then choosing the value of characteristic curve with maximal membership grade. For example, dirty data exists in the curve $\boldsymbol{x}_\mathrm{j}$ from point t1 to t2, and $\boldsymbol{v}_{i1}$ $\boldsymbol{v}_{i2}$ stand for two clustering center with maximal membership grade, the amendment of dirty data can be calculated as:

$$x_j^{'}(t) = v_{i1}^{'}(t)\frac{u_{i1,j}}{u_{i1,j}+u_{i2,j}} + v_{i2}^{'}(t)\frac{u_{i2,j}}{u_{i1,j}+u_{i2,j}} \cdot \tag{8}$$

$$v_{i1}^{'}(t) = v_{i1}(t)\times(\frac{x_j(t1-1)}{v_{i1}(t1-1)} + \frac{x_j(t2+1)}{v_{i1}(t2+1)})/2 \cdot \tag{9}$$

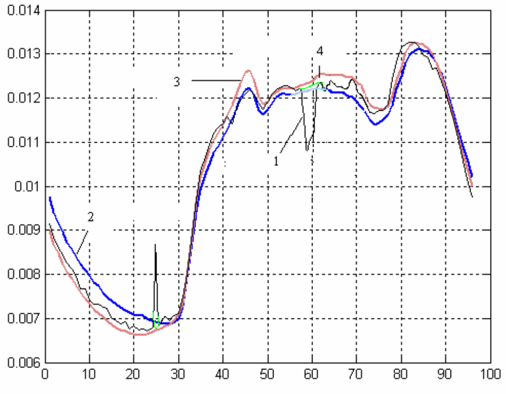
$$v_{i2}^{'}(t) = v_{i2}(t)\times(\frac{x_j(t1-1)}{v_{i2}(t1-1)} + \frac{x_j(t2+1)}{v_{i2}(t2+1)})/2 \cdot \tag{10}$$

3 Result Analysis

Data in workday and weekend are put into SFKN respectively because these two kinds of load curves are obviously different. This operation reduce the amount of training calculation and the number of clustering centers, increase the calculation speed and improve the efficiency of model. The following is an example by using actual load data from April to Sep.2003 of Jiangbei power supply bureau in Chongqing.

3.1 Adjusting Results

Load data in Oct.2003 (out of the training set) are adjusted randomly. Fig. 2 is a typical load curve with dirty data from which we can find the methods of amendment and results clearly.



1: real load curve; 2: clustering center with maximal membership grade
3: clustering center with sub maximal membership grade; 4: curve after adjust

Fig. 2. inspecting curve for dirty data

3.2 Comparison of Accuracy Between Hard Clustering and Soft Clustering

In order to illustrate the advantages of SFKN proposed in this paper, we replace the SFKN in the first layer of model by general Kohonen network. On the basis of a daily load data in Oct.2003, we fabricate some dirty data. Results of two methods are shown in table 1, the accuracy of method proposed is higher than general Kohonen method.

Table 1. The result campare of two kinds of correct models

Dirty data points	Error before adjust	General Kohonen	SFKN
1	26.4%	4.7%	2.8%
2	32.3%	7%	3.3%
3	15.5%	4.2%	2.1%
4	18.7%	3.5%	4.2%
5	8.5%	5.6%	1.9%

3.3 Dynamic Updating Algorithm

In order to verify the efficiency of this algorithm, dirty data of 5 days in Dec.2003 are firstly adjusted without using dynamic updating algorithm (swatch data is in the year 2003 from April to September, 96 points per day). Then the model of dynamic updating algorithm is used and swatch data set is updated till one day ahead of detecting day. Results are shown in as follows in table 2.

Results of dynamic updating algorithm are satisfied because its mistakes are less than non-dynamic algorithm. In dynamic updating algorithm, the latest adjusted clustering center and vectors are used which increase the membership grade of load curves and clustering center. Dynamic updating algorithm improves not only the accuracy of dirty data detection but also precision of dirty data adjustment.

Table 2. The result of random check to load data

Date No	Count of dirty data	Non- dynamic algorithm		Dynamic algorithm	
		Failed to judge	Misjudged	Failed to judge	Misjudged
1	20	3	2	1	1
2	15	5	2	1	0
3	18	2	0	0	0
4	17	1	2	1	0
5	20	3	1	0	2
Total	90	14	7	3	3

4 Conclusion

Examples demonstrate that Kohonen NN is meliorated according to fuzzy soft clustering thinking which obtains the clustering center quickly and reasonably. The proposed dynamic updating algorithm can adjust clustering center automatically according to newly added data. Strong ability of pattern recognition of RBF facilitates the location process of dirty data. The dynamic intelligent adjusting model in this paper can process data dynamically with higher accuracy and faster convergence rate.

References

1. Ming Zhu: Data Mining. China Science and Technology University Press. 2002
2. Weiren Mo, Boming Zhang, Hongbing Sun, Zihang Hu: Application of extended Short-Term Load Forecasting. Power System Technology, 2003,Vol 27(5): 6-9
3. Chongqing Kang, Qing Xia: Parameter estimation and bad data identification of gray systems. Journal of Tsinghua University(Sci& Tech), 1999, 4: 72-75
4. Guojiang Zhang, Jiaju Qiu, Juhong Lee: Outlier identification and justification based on neural network. Proceeding of the CSEE, 2001, Vol 21(8): 104-107
5. Nicolaos B.Karayiannis: An axiomaticn approach to Soft learning vector quantization and clustering. IEEE Transon Neural Networks, 1999 :1015-1019
6. Sato-Ilic, M: Fuzzy regression analysis using fuzzy clustering. 2002 Fuzzy Information Processing Society Annual Meeting of the North American, New Orleans, USA, 2002 :57 – 62
7. Xianghao Lee: Fuzzy clustering analysis and application. GuiZhou Science Press, 1994

Using Boosting Learning Method for Intrusion Detection

Wu Yang[1], Xiao-Chun Yun[1, 2], and Yong-Tian Yang[1]

[1] Information Security Research Center,
Harbin Engineering University, Harbin 150001, China
yangwu@pact518.hit.edu.cn
[2] Computer Network and Information Security Technique Research Center,
Harbin Institute of Technology, Harbin 150001, China
yxc@hit.edu.cn

Abstract. It is an important research topic to improve detection rate and reduce false positive rate of detection model in the field of intrusion detection. This paper adopts an improved boosting method to enhance generalization performance of intrusion detection model based on rule learning algorithm, and presents a boosting intrusion detection rule learning algorithm (BIDRLA). The experiment results on the standard intrusion detection dataset validate the effectiveness of BIDRLA.

1 Introduction

With the rapidly growing connectivity of the Internet, network attacking events have been increasing continually. Intrusion detection systems (IDS) are introduced as a second line of defense and become a research hotspot in the fields of network security. At present, intrusion detection techniques can be categorized into misuse detection and anomaly detection. Misuse detection systems, for example [1], use patterns of well-known attacks or weak spots of the system to identify intrusions. The main shortcomings of such systems are: known intrusion patterns have to be hand-coded into the system; they are unable to detect any future (unknown) intrusions that have no matched patterns stored in the system; anomaly detection systems, such as [2], firstly establish normal user behavior patterns (profiles) and then try to determine whether deviation from the established normal profiles can be flagged as intrusions. The main advantage of anomaly detection systems is that they can detect new types of unknown intrusions. In recent years, the continual emergence of new attacking methods has caused great loss to the whole society. So, the advantage of detecting future attacks has specially led to an increasing interest in anomaly detection techniques.

This paper does in-depth research on anomaly detection method proposed in the article [3], which firstly applies rule learning algorithm (RIPPER) [4] to intrusion detection and acquires good results on identifying intrusion events accurately. Compared with other automatic modeling methods such as neural network and instance based method, rule learning algorithm has obvious advantage in model comprehensibility, adaptability to data types and detection efficiency. In building intrusion detection models with rule learning algorithm, it is often the case that training samples for in-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 634–641, 2005.

trusion detection are insufficient. In this case, generalization performance of intrusion detection model produced by rule learning algorithm is somewhat inferior. In order to enhance classifying performance of weak rule learner, we adopt boosting method [5] for intrusion detection based on rule learning, improve and present a boosting intrusion detection rule learning algorithm (BIDRLA).

This paper is organized as follows: In section 2, we will firstly describe an improved boosting algorithm, and then give the related definitions for BIDRLA algorithm. In section 3, we give the results of experiments and analysis. At last, conclusion and future work is drawn in section 4.

2 Boosting Intrusion Detection Rule Learning Algorithm

The basic idea of BIDRLA algorithm is that it adopts boosting method to enhance classifying performance of weak rule learner (RIPPER) in the case of training example insufficiency. Boosting is based on the observation that finding many rough rules of thumb can be a lot easier than finding a single, highly accurate prediction rule. To apply the boosting approach, we start with a weak algorithm for finding the rough rules of thumb. The boosting algorithm calls this weak or base learning algorithm repeatedly, each time feeding it a different subset of the training examples (or, to be more precise, a different distribution or weighting over the training examples). Each time it is called, the base learning algorithm generates a new weak prediction rule, and after many rounds, the boosting algorithm must combine these weak rules into a single prediction rule that hopefully will be much more accurate than any one of the weak rules.

A big drawback in solving practical problems for the earlier boosting algorithms is that these algorithms must acquire the lower bound of learning correctness rate of weak learning algorithm, but it is difficult to come true in practice. AdaBoost [6] as a boosting algorithm solves many of the practical difficulties of the original boosting algorithms and gains broad applications.

2.1 Improved AdaBoost Algorithm

The AdaBoost algorithm takes as input a training set $(x_1, y_1),....,(x_m, y_m)$ where each x_i belongs to some domain or instance space X, and each label y_i is in some label set Y. For most of this paper, we assume $Y = \{-1, +1\}$. AdaBoost calls a given weak or base learning algorithm repeatedly in a series of rounds $t = 1,....T$. One of the main ideas of the algorithm is to maintain a distribution or set of weights over the training set. The weight of this distribution on training example i on round t is denoted $D_t(i)$ that reflects its importance. Initially, all weights are uniformly distributed, but on each round, the weights of incorrectly classified examples are increased so that the base learner is forced to focus on the hard examples in the training set. The base learner's job is to find a base classifier $h_t : X \to \Re$ appropriate for the distribution D_t (Base classifiers are also called rules of thumb or weak prediction rules). Once the base classifier h_t has been received, AdaBoost chooses a parameter $\alpha_t \in \Re$ that intuitively measures the importance that is assigned to h_t. Pseudo code for AdaBoost algorithm is described in [6].

In intrusion detection application, AdaBoost algorithm sometimes leads to a deterioration in generation performance on some intrusion detection datasets, the main reason for which is that the class distributions across the weight vectors for these datasets become very skewed in the process of generating final classifier [7]. Such skewed weights seem likely to lead to an undesirable bias towards or against predicting some classes, with a concomitant increase in error on unseen instance. This is especially damaging when the classifier derived from the skewed distribution has a high voting weight. It may be possible to modify the weight update process of traditional AdaBoost algorithm so that weights are adjusted separately within each class without changing overall class weights. Detailed description is as follows:

1. Initialization. Proportion of weight distribution is separately set for every class so as to restrict weight update on overall training dataset. For example (x_i, y_i), class $y_i = Y_j, j = 1,...k$, we can set proportion of weight distribution of every class as $a_1 : a_2 : ....a_k$ and $a_1 + a_2 + ...a_k = 1$. This proportion is constant during each iteration.

2. Initialization of weight in the examples with the same class. Suppose that m is the number of examples with class Y_j, the initial weight of every example of class Y_j is a_j / m.

3. During each iteration, weight is adjusted according to proportion of weight distribution of each class. If example (x_i, y_i) satisfies equation $y_i = Y_j$, $\sum D_t(x_i, y_i) = a_j$.

2.2 Definitions and Description for BIDRLA Algorithm

In BIDRLA algorithm, the evaluating criteria for rule growing and rule pruning of the traditional RIPPER algorithm are modified through the formal analysis of AdaBoost algorithm. In order to describe BIDRLA algorithm accurately, firstly we give interrelated definitions and inferences used in BIDRLA:

Definition 1. If a weak hypothesis from weak rule learner is formalized as $h : X \to \Re$, the sign of $h(x)$ is defined as the predicted label and the magnitude $|h(x)|$ as the confidence in the prediction: large numbers for $|h(x)|$ indicate high confidence in the prediction, and numbers close to zero indicate low confidence.

Definition 2. The weak-hypotheses are rules, which are conjunctions of primitive conditions. A rule R can be any hypothesis that partitions the set of instances x into two subsets: the set of instances which satisfy (are covered by) the rule, and those which do not satisfy the rule. If x satisfies R, we will write $x \in R$. Otherwise, we will write $x \notin R$.

Definition 3. In order to make the strong-hypotheses similar to a conventional ruleset, the weak-hypotheses based on a rule R are forced to abstain on all instances unsatisfied by R by setting the prediction $h(x)$ for $x \notin R$ to 0 and predict with the same confidence C_R on every $x \in R$. So, for the t-th rule R_t generated by the weak learner, we define that $\forall x \in R_t$, $\alpha_t h_t(x) = C_{R_t} (C_{R_t} > 0)$; otherwise, $\forall x \notin R_t, \alpha_t h_t(x) = 0$.

Definition 4. We define that each rule R to be one of the two forms: either R is a "default rule" (i. e., $x \in X \Rightarrow x \in R$) or else R is "non-default rule". Each non-default rule R is associated with a single real-value positive confidence C_R.

Inference 1. Z_t is a real value used to normalize the distribution: $Z_t = \sum_i D_t(x_i, y_i) \exp(-\alpha_t y_i h_t(x_i))$. Schapire [8] showed that to minimize training error, the weak-learning algorithm should pick, on each round of boosting, the weak hypothesis h_t and weight α_t which lead to the smallest value of Z_t. Assume that a rule R has been generated by the weak learner and the confidence value C_R is for the rule R. Omitting the dependency on t, Z can be rewritten as

$$Z = \sum_{x_i \notin R} D(x_i, y_i) + \sum_{x_i \in R} D(x_i, y_i) \exp(-y_i C_R) \tag{1}$$

where $C_R = \alpha h(x)$. Let $W_0 = \sum_{x_i \notin R} D(x_i, y_i)$, $W_+ = \sum_{x_i \in R; y_i = +1} D(x_i, y_i)$, $W_- = \sum_{x_i \in R; y_i = -1} D(x_i, y_i)$. We can now further simplify Eq. (1) and rewrite Z as

$$Z = W_0 + W_+ \exp(-C_R) + W_- \exp(+C_R) \tag{2}$$

To minimize Z, we need to solve the equation $dZ / dC_R = 0$, which implies that Z is minimized by setting

$$C_R = \frac{1}{2} \ln(\frac{W_+}{W_-}) \tag{3}$$

In Eq. (3), W_- can be equal to 0, leading to extreme confidence values: to prevent this, in practice, we smooth the confidence by adding $1/2m$ to both W_+ and W_-:

$$\hat{C}_R = \frac{1}{2} \ln(\frac{W_+ + 1/(2m)}{W_- + 1/(2m)}) \tag{4}$$

Plugging the value of C_R into Eq. (2), we get that

$$Z = W_0 + 2\sqrt{W_+ W_-} = 1 - (\sqrt{W_+} - \sqrt{W_-})^2 \tag{5}$$

Thus, a rule R minimizes Z if it maximizes $|\sqrt{W_+} - \sqrt{W_-}|$. According to the constraint of definition 4, the confidence value C_R is positive. Thus the value of $\sqrt{W_+} - \sqrt{W_-}$ is also positive. So, the objective function we attempt to maximize when searching for a good rule is

$$\widehat{Z} = \sqrt{W_+} - \sqrt{W_-} \tag{6}$$

So the *GrowRule* routine in weak learner of BIDRLA begins with an empty conjunction of conditions and considers adding to this conjunction the condition that attains the maximal value $\widehat{Z}$ on *GrowSet*. This process is repeated until the rule covers no negative examples from *GrowSet*, or no further refinement improves $\widehat{Z}$.

```
Given: $S = \{(x_1, y_1), \ldots (x_m, y_m)\}, x_i \in X$,
           $y_i \in Y = \{Y_1, Y_2\} = \{-1, +1\}, i = 1, .. m$
       $S_j = \{(x_i, y_i) | y_i = Y_j, (x_i, y_i) \in S\}$
       $a_j$ for each $Y_j$ and $a_1 + a_2 = 1$
Initialize: for each $(x_i, y_i) \in S$, if $(x_i, y_i) \in S_j, D_1(x_i, y_i) = a_j / |S_j|$
For t=1,.....T:
  Train the weak-learner using current distribution $D_t$
      (I) Split data into GrowSet and PruneSet
      (II) GrowRule: starting with empty rule, greedily add conditions to
           maximize Eq. (6)
      (III) PruneRule: starting with the output $R'$ of GrowRule, delete some final
           sequences of conditions to minimize Eq. (7), where $\widehat{C}_{R'}$ is computed
           using Eq. (4) and GrowSet
      (IV) Return as $R_t$ either the output of PruneRule, or the default rule
  Construct weak hypotheses $h_t : X \to \Re$ :
      Let $\widehat{C}_{R'}$ be given by Eq. (4) (evaluated on the entire dataset), then
           $\alpha_t h_t(x) = \begin{cases} \widehat{C}_{R_t} & if\ x \in R_t \\ 0 & otherwise \end{cases}$
  Update weight of the distribution:
      (I) For each $x_i \in R_t$, set $D_t(x_i, y_i) \leftarrow D_t(x_i, y_i) / \exp(y_i \cdot \widehat{C}_{R_t})$;
      (II) For each class $Y_j$, let $Z_j^t = (\sum_{(x_i, y_i) \in S_j} D_t(x_i, y_i)) / a_j$;
      (III) For each $x_i$, if $(x_i, y_i) \in S_j$, set $D_{t+1}(x_i, y_i) \leftarrow D_t(x_i, y_i) / Z_j^t$;
Output final hypothesis: $H(x) = sign(\sum_{R_t : x \in R_t} \widehat{C}_{R_t})$
```

Fig. 1. The BIDRLA algorithm

Definition 5. The resulting rule is immediately pruned using the *PruneRule* routine because the growing rule is often too specific and overfits the training data. Assume that each candidate rule for pruning is written as $R^{'}$, rule pruning is operated on *PruneRule* by deleting any final sequence of conditions from this rule. Similar to the definition of W_+ and W_-, let $V_+ = \sum_{x_i \in R^{'}; y_i = +1} D(x_i, y_i)$, $V_- = \sum_{x_i \in R^{'}; y_i = -1} D(x_i, y_i)$. Denote by $\widehat{C}_{R'}$ the smoothed prediction confidence obtained by evaluating Eq. (4) on the W_+, W_- associated with *GrowSet*. *PruneRule* minimizes the formula

$$(1 - V_+ - V_-) + V_+ \exp(-\widehat{C}_{R^{'}}) + V_- \exp(+\widehat{C}_{R^{'}}) \qquad (7)$$

Pseudo code for BIDRLA algorithm (for Boolean classification) is given in Fig. 1. AdaBoost method can reduce the size of the training dataset required for intrusion detection model based on RIPPER algorithm, enhancing weak learner to improve the detection accuracy of intrusion detection model.

3 Experiment Results and Analysis

Experiments have been carried out on a subset of the database created by DARPA in the framework of the 1998 Intrusion Detection Evaluation Program. This subset has been pre-processed by the Columbia University and distributed as part of the UCI KDD Archive [9]. The available dataset is made up of a large number of network connections related to normal and malicious traffic. Each connection is represented with a 41-dimension feature vector. Connections are also labeled as belonging to one out of five classes, i.e. Normal, Denial of Service (DOS) attacks, Remote to Local (R2L) attacks, User to Root (U2R) attacks, and Probing attacks (Probe).

The whole dataset is composed of the training dataset (Train_Data) and testing dataset (Test_Data). In the original datasets, all classes of attack examples mix up together, which makes the original datasets include much noise. So, in order to acquire accurate experiment results, the original datasets should be partitioned according to attack classes of examples. For four attack classes in Train_Data and Test_Data, we separately mix four classes of attack examples with normal examples and proportionally select parts of these four mixed datasets into four training and testing datasets for four attack classes, i.e. (DOS_Train_Data, DOS_Test_Data), (Probe_Train_Data, Probe_Test_Data), (U2R_Train_Data, U2R_Test_Data), (R2L_Train_Data, R2L_Test_Data).

1. After training BIDRLA, RIPPER and C4.5Rules on training datasets for four attack classes, we separately test these three algorithms on the corresponding testing datasets to compare their detection rates (DR%) and false positive rates (FPR%).

We perform five tests on different subsets of training and testing dataset for every algorithm. The experiment results are the average of five testing values and are given in Table 1, 2, 3 and 4 (Experiment parameter setting: the number of iterations in BIDRLA algorithm is set as 20; the number of rule optimization in RIPPER is set as 1; other parameters in three algorithms are set to the default value).

From these tables, the detection rates and false positive rates of RIPPER and C4.5Rules are nearly equal, but the experiment results of RIPPER are slightly better than those of C4.5Rules. The detection rates of BIDRLA are higher than those of RIPPER and C4.5Rules, and its false positive rates are lower, which shows that optimized AdaBoost algorithm can improve generalization performance of original rule learning algorithm. For the training dataset of U2R class, because the number of training examples is few, there is over-fitting with RIPPER on this dataset and the detection precision of RIPPER is relatively low. But generalization performance of weak rule learner is greatly improved with boosting BIDRLA algorithm in this case of training example insufficiency.

Table 1. Detecting results for DOS attack with C4.5Rules、RIPPER and BIDRLA

	C4.5Rules	RIPPER	BIDRLA
DR(%)	91.89	92.34	93.16
FPR(%)	2.41	2.36	2.29

Table 2. Detecting results for Probe attack with C4.5Rules、RIPPER and BIDRLA

	C4.5Rules	RIPPER	BIDRLA
DR(%)	81.29	81.42	82.73
FPR(%)	0.54	0.47	0.45

Table 3. Detecting results for R2L attack with C4.5Rules、RIPPER and BIDRLA

	C4.5Rules	RIPPER	BIDRLA
DR(%)	78.61	78.65	80.37
FPR(%)	1.12	1.14	1.05

Table 4. Detecting results for U2R attack with C4.5Rules、RIPPER and BIDRLA

	C4.5Rules	RIPPER	BIDRLA
DR(%)	17.53	17.56	22.12
FPR(%)	0.31	0.35	0.24

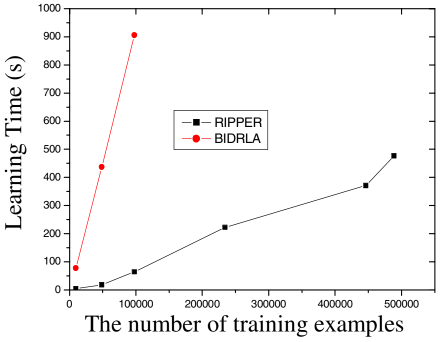


Fig. 2. Learning time of BIDRLA and RIPPER

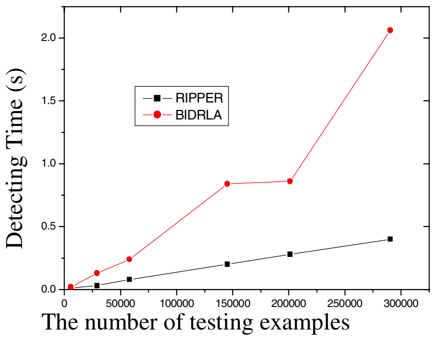


Fig. 3. Detecting time of BIDRLA and RIPPER

2. We test the learning time and detecting time on different training and testing datasets for RIPPER and BIDRLA. The configuration of machines in the experiments are: dual 1GHz PIII CPU, 2G main memory, Intel 1000Mbps Ethernet NIC and 18G SCSI hard disk. The Linux kernel version is 2.4.10. The results are shown in Figs. 2 and 3. The learning time of BIDRLA is long (when the number of training examples reaches 200000, its learning time exceeds 75 minutes) and the detecting time is also longer than that of RIPPER. This indicates that although BIDRLA can improve the intrusion detection precision, it is achieved at a cost of computation efficiency. We conclude that BIDRLA is applicable for building intrusion detection model on small dataset.

4 Conclusion and Future Work

In the case of lacking training examples for intrusion detection, the generalization performance of intrusion detection model by the traditional RIPPER algorithm is inferior. This paper adopts AdaBoost algorithm to enhance the classification performance of rule learner. To avoid the deterioration in generation performance on some intrusion detection datasets, the update process of weight distribution in AdaBoost is improved. The criteria of rule growing and rule pruning are modified in traditional RIPPER algorithm. So, boosting intrusion detection rule learning algorithm (BIDRLA) is presented. The experiment results confirm the effectiveness of BIDRLA, but its detection precision is improved at a cost of running efficiency. Future work includes how to improve the learning and detecting efficiency of BIDRLA.

References

1. Illgun K., Kemmerer R., Philips A.: State Transition Analysis: A Rule-based Intrusion Detection Approach. IEEE Transaction on Software Engineering. 2 (1995) 181-199
2. Karlton S., Mohammed Z.: ADMIT: Anomaly-based Data Mining for Intrusions. In Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, Edmonton Alberta Canada (2002) 386–395
3. Wenke L., Stolfo S. J., Mok K. W.: A Data Mining Framework for Building Intrusion Detection Models. In Proceedings of the 1999 IEEE Symposium on Security and Privacy. IEEE Press, Oakland CA USA (1999) 120-132
4. William W. C.: Efficient Rule Induction. In Proceedings of the 12th International Conference on Machine Learning. Morgan Kaufmann (1995)
5. Schapire R. E.: The Strength of Weak Learnability. Machine Learning. 2 (1990) 197-227
6. Freund Y., Robert E. S.: A Decision-Theoretic Generalization of On-line Learning and an Application to Boosting. Journal of Computer and System Sciences. 1 (1997) 119-139
7. Quinlan J. R.: Bagging, Boosting and C4.5. In Proceedings of the Thirteenth National Conference on Artificial Intelligence. Menlo Park (1996) 725-730
8. Schapire, R. E., Singer Y.: Improved Boosting Algorithms Using Confidence-rated Predictions. In Proceedings of the Eleventh Annual Conference on Computational Learning Theory. ACM Press, Madison Wisconsin United States (1998) 80-91
9. KDD CUP 1999. http://kdd.ics.uci.edu/database/kddcup99/kddcup99.html (1999)

RoleOf Relationship and Its Meta Model for Design Pattern Instantiation

Chengwan He[1,2], Fei He[3], Keqing He[1], Jin Liu[1], and Wenjie Tu[2]

[1] State key laboratory of Software Engineering, Wuhan University,
Wuhan, 430072, China
hechengwan@mail.whict.edu.cn

[2] Department of Computer Science and Technology, Wuhan Institute of Chemical Technology, Wuhan, 430073, China

[3] Major in Computer Science, Graduate School of Science and Engineering,
Waseda University, Tokyo, 169-8555, Japan

Abstract. This paper states that the RoleOf Relationship can provide a general approach to resolve instantiation problems of design patterns. The problems come from the fact that pattern logic scatters across multiple business classes (classes specific to each application). This causes problems such as decreasing reusability of pattern logic, and losing of the instantiation information of pattern (traceability and overlapping problem) etc. To resolve these problems in design level, an approach for design pattern instantiation based on RoleOf relationship is proposed. In our approach, roles of pattern are treated as the independent modeling elements and RoleOf relationship is used to associate a role with a business class. The meta model of RoleOf relationship for pattern instantiation and its semantics are proposed as well. Examples are used to illustrate this approach. Implementation and behavior description of RoleOf relationship are also presented in the paper.

1 Introduction

Design patterns [1] have been recognized to be very important and useful in real software development. Especially, for software reuse. However, several problems still exist when applying pattern: implementation, documentation and composition [2]. Documentation problem can be called traceability problem [3][4], which is caused by the scattering of pattern logic(attributes and methods in role of a pattern). Composition problem can be called overlapping problem [5] of pattern, which is a special case of the documentation problem.

In Unified Modeling Language (UML) [6], the parameterized collaboration is used to present design patterns. The template parameters of a collaboration are specified using ClassifierRoles, which are placeholders for objects that will interact to achieve the collaboration's goal. In each use of the pattern, actual classes are substituted for the parameters in the pattern definition, the business logic and pattern logic are bound together. This makes it difficult to distinguish them and the functionality of the business class is unclear. It also reduces the reusability of the business class. In addition, when the role of a pattern is bound with multiple classes, the pattern logic scatters

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 642 – 653, 2005.

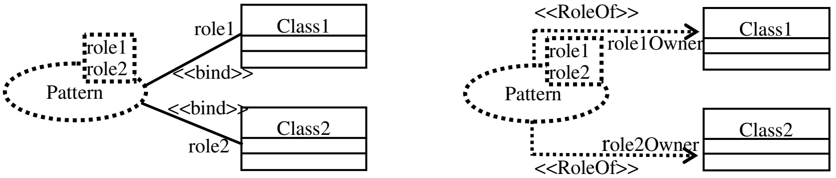
across numerous business classes. This reduces the reusability of the pattern logic and makes the maintenance of application very difficult.

This paper presents that RoleOf Relationship can provide a general approach resolving instantiation problems of design patterns in design level. The concept of role is widely applied to the conception modeling of business system containing object reclassification [7][8]. Dahchour brought forward the conception of RoleOf relationship [9] explicitly to describe the relationship between object and role. In our approach, roles in pattern are treated as the independent modeling elements(called role class) and RoleOf relationship is used to associate a role with a business class. This helps to improve the reusability of the business logic and pattern logic.

We propose the meta model of RoleOf relationship for design pattern instantiation and its semantics using UML extension mechanisms. We illustrate this approach with examples and discuss the implementation method of RoleOf relationship. Through analysis and comparison of design pattern instance model before and after using RoleOf relaionship, our conclusion is validated.

2 Design Pattern Instantiation Based on RoleOf Relationship

When instantiating design pattern, generally, the business class is bound with a certain role in the pattern. As Fig. 1(a) shows, Class1 and Class2 are two business classes, which are bound with role1 and role2 respectively. After being bound, Class1 and Class2 contain the attribute and method of role1 and role2 respectively. It is not only difficult to distinguish the business logic and the pattern logic but also make the function of the business class implicit.



(a) Binding of business class and role

(b) RoleOf relationship between business class and role

Fig. 1. Design pattern instantiation

In order to improve the reusability of pattern logic in design level, it is essential to separate the business logic with the pattern logic. We use RoleOf relationship [9] to describe the relation between role and business class. Fig. 1(b) shows that role1 and role2 are roles of Class1 and Class2 respectively. After using RoleOf relationship, role and the business class can be regarded as separated modeling elements. In order to model RoleOf relationship, we extend UML and provide the meta model and its semantics of RoleOf relationship.

The meta model of RoleOf relationship for design pattern instantiation is shown in Fig. 2. RoleOf relationship is a dependency relationship. Two ends of RoleOf relationship are classes, where *roleOwner* is the business class and *role* is the role class.

But the role class needs the stereotype <<PatternRole>>. In addition, two tagged values *patternName* and *roleName* are defined to present the pattern and the role names that the business class and the role class belongs to. For example, if {patternName=Observer}, {roleName=Subject} are applied to a role class, then such role is *Subject* role in *Observer* pattern.

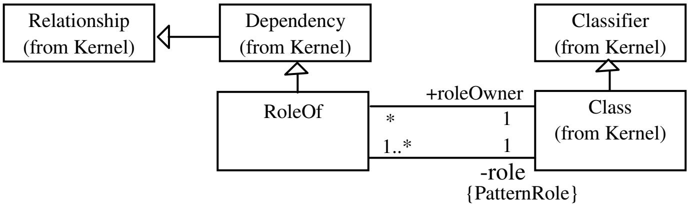


Fig. 2. Meta model of RoleOf relationship

Semantics of RoleOf relationship can be described as the follows:

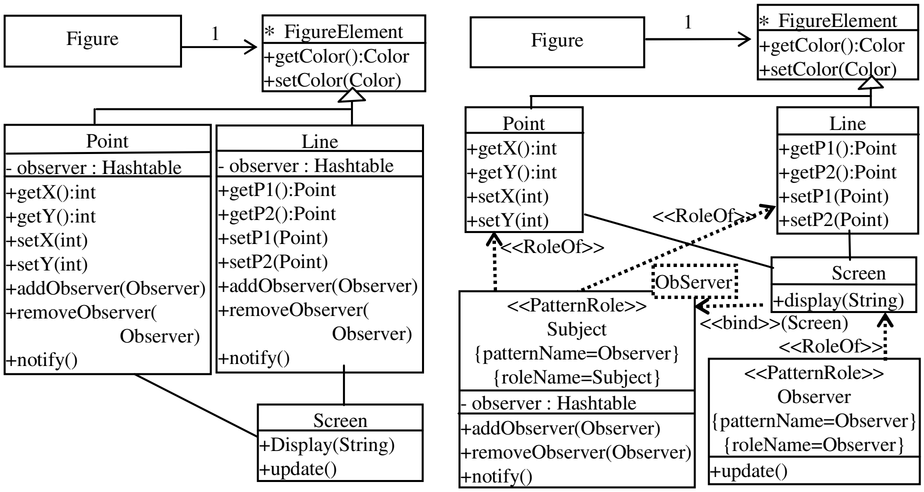
- RoleOf relationship belongs to dependency relationship. Such dependency relationship can be interpreted from two aspects: (i) the object's behavior relies on the role's behavior when one object's role is in operation; and (ii) the role can't exist alone and only takes effect after being related with an object.
- A business class may have more than one RoleOf relationship(multiplicity is *), which may relate to zero or more than one role class(PatternRole). On the other hand, a role class may also relate to one or more business classes(multiplicity is 1..*). This means that the role class can't exist alone and it has not any meaning until being related with one or more business classes.
- The definition method of the role class is the same as the business class. But, if the attribute and the method of a role class have reference type parameter, for example there is an *addObserver (Observer)* method in *Subject* role of *Observer* pattern [1] and its parameter type is *Observer* class, it needs to define the role class as the template class and bind the actual parameter with the template class.
- RoleOf relationship can only relate business class with role class.

3 Example

The *Observer* pattern [1] is used as the example. Application and implementation of RoleOf relationship in pattern instantiation is discussed with analysis on how to resolve the reuse of the role class and the documentation problem at the same time.

Observer pattern is very commonly used. It maintains information consistency between multiple objects, i.e. when the state of an object changes, it will notify other relevant objects. [2] introduces a simple figure element system using *Observer* pattern: *Point* class and *Line* class inherit the *FigureElement* class, which correspond to the *Subject* role in *Observer* pattern, and both contain *addObserver(), removeObserver(), notify()* methods etc. *Screen* class, corresponding to the *Observer* role in *Observer* pattern, contains *update()* method. See Fig. 3(a).

Class structure based on RoleOf relationship is shown in Fig. 3(b). *Subject* role and *Observer* role in *Observer* pattern are defined as two independent classes identified by stereotype <<PatternRole>>, each containing attributes and methods of the *Subject* role and the *Observer* role. And *patternName* and *roleName* tagged values are used to identify the pattern name and the role name. RoleOf relationship exists between *Subject* role, *Point* class and *Line* class. Equally, it still exists between *Observer* role and *Screen* class. In addition, *addObserver()* and *RemoveObserver()* methods in *Subject* role have reference type parameters, so *Subject* role class is defined as template class and *Screen* class is bound with *Observer* parameter.



(a) Before using RoleOf relationship (b) After using RoleOf relationship

Fig. 3. A figure element system based on *Observer* pattern

3.1 Static Implementation

The main purpose of using RoleOf relationship is to reuse the pattern logic scattering across multiple business objects, so it is an appropriate choice to use AOP [10] technology to implement RoleOf relationship. AspectJ [11] is a kind of AOP language most widely used at present. In AspectJ, *instruction* mechanism is used to insert members and relationships to base class. It is used to implement role class.

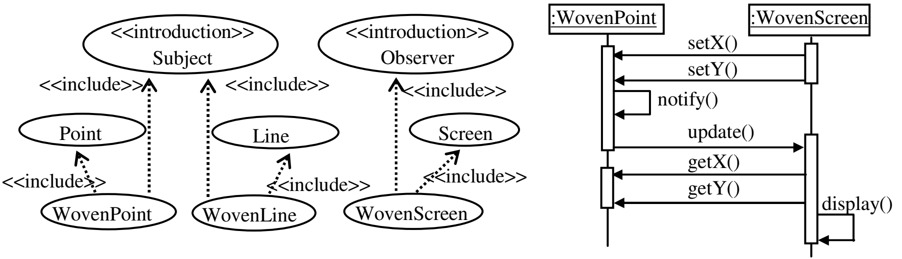
An implementation of *Subject* role class using AspectJ is shown below. Two aspects are defined, in which *PointSubject* is a role class related with *Point* class and *LineSubject* is the one related with *Line* class.

```
public aspect PointSubject {
    private Hashtable Point.observer;
    public void Point.addObserver(Screen observer){
        this.observer.add(observer);}
    public void Point.removeObserver(Screen observer){
        this.observer.remove(observer);}
    public void Point.notify(){//…}
}
```

```
public aspect LineSubject {
    private Hashtable Line.observer;
    public void Line.addObserver(Screen observer){
        this.observer.add(observer);}
    public void Line.removeObserver(Screen observer){
        this.observer.remove(observer);}
    public void Line.notify(){//…}
}
```

To the static implementation method above, pattern logic is added into the business class at compile time, so the instance of role class will not be created at run time. With this implementation method, the behaviors of the pattern instance include weaving of Aspect and interaction between the instances of the business class after weaving Aspect. Dominik Stein [12] proposed a UML profile for AspectJ modeling, which uses the *Use Case* diagram of UML to describe the weaving of *instruction*.

Fig. 4(a) uses the UML *Use Case* diagram to describe the weaving process of Aspect: *Subject* use case is woven into the use case *Point* and *Line*, and *Observer* use case is woven into the use case *Screen*. Fig. 4(b) uses UML *Sequence* diagram to describe the interaction between *Point* object and *Screen* object after weaving Aspect, and this interaction is just the protocol depicted by *Observer* pattern.



(a) Weaving introductions with UML Use Cases

(b) The interaction between *Point* and *Screen* object after weaving Aspect

Fig. 4. Behavior description of example introduced in Fig. 3

3.3 Dynamic Implementation

The dynamic implementation can realize the dynamic binding between the business object and the role object at run time [13][14]. We propose a reflective approach to model dynamic object behavior [14]. It structures a system into base level and meta level. The meta objects in meta level manage the base objects and roles in base level. The collaboration between *Point* object and *Subject* role object is shown in Fig.5.

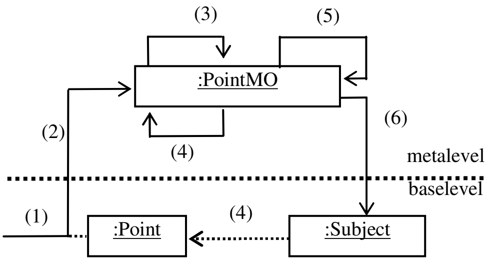
We propose a linguistic extension of Java, RoleJava [14] where roles are defined using the keyword *ROLE*, business objects are defined as subclasses of *ReflectiveObject*, and meta objects are created automatically by RoleJava compiler. For example, *Point* class and *Subject* role class can be built by RoleJava as follows.

```
//Client.rjava
public class Client{
   public static void main(String args[]){
      Point point = new Point();
      Screen screen = new Screen();
```

```
        //invoke addObserver() method of Subject role
        point.INVOKEROLE(“Subject”,”addObserver(screen)”);}
}

//Point.rjava
public class Point extends ReflectiveObject{
    public int getX(){//…}
    public void setX(int x){//…}
}

//Subject.rjava
ROLE Subject{
    private Hashtable observers;
    public void addObserver(Observer o){
        observers.add(o);}
    public void removeObserver(Observer o){
        observers.remove(o);}
    public void notify(){//…}
}
```



(1) client invokes a role method of *Point* object; (2) the invocation is intercepted by meta object *PointMO*, and the execution control jumps from base level to meta level; (3) *PointMO* checks whether the requested role can be bound with its object; (4) *PointMO* creates the role instance and binds it with *Point* object; (5) *PointMO* checks the pre-condition; (6) *PointMO* invokes the method of the role, and the execution control jumps back to base level.

Fig. 5. The collaboration between object and its role

3.4 Quantitative Evaluation

We choose the metrics applied to UML model that is brought forward by Kim [15] to evaluate the design pattern instance model after using RoleOf relationship. The metrics of Kim can be divided into four kinds: Model, Class, Message, and Use Case, totally including 23 kinds of metrics. Class Metrics contain 13 kinds: NATC1(Number of the attributes in a class – unweighted), NATC2(Number of the attributes in a class – weighted), NOPC1(Number of the operations in a class – unweighted), NOPC2(Number of the operations in a class – weighted), NASC(Number of the associations linked to a class), CBC(Coupling between classes), DIT(Depth of inheritance tree), NSUPC(Number of the superclasses of a class), NSUPC*(Number of the elements in the transitive closure of the superclasses of a class), NSUBC(Number of the subclasses of a class), NSUBC*(Number of the elements in the transitive closure of the subclasses of a class), NMSC(Number of messages sent by the instantiated objects of a class), NMRC(Number of messages received by the instantiated objects of a class). From Table 1 we can conclude that,

– After using RoleOf relationship, NATC1, NOPC1 and NOPC2 are smaller than those that don't use RoleOf relationship, which shows that a class is easy to reuse and maintain. The value of NATC2 is 0, because we suppose that all the attributes of the classes are private and the weights for private attributes is 0.

- After using RoleOf relationship, NASC and CBC become higher, which shows that the coupling between the classes increases and it has an effect on the maintenance of system. But in a complicated system, much more the number of the classes is, much smaller the effect is.
- After using RoleOf relationship, DIT, NSUPC, NSUPC*, NSUBC, NSUBC* are smaller than those that don't use RoleOf relationship, which indicates that the model is easy to maintain, and the reusability is improved.
- NMSC and NMRC measure the quantities of the messages transferred between the objects after instantiation. Because we use Aspect to realize role class, the attribute and method of role class are put into the corresponding business class at compile time and the number of the classes as well as its attributes and methods are all consistent no matter using or not using RoleOf relationship when the system is running, which causes that the values of NMSC are equal in both cases, and NMRC is the same too.

Table 1. The values of metrics and change rate of example introduced in Fig. 3

Metrics	V1*	V2*	Percentage	Metrics	V1*	V2*	Percentage
NATC1	0.4	0.14	-65	NSUPC	0.4	0.29	-27
NATC2	0	0	0	NSUPC*	0.4	0.29	-27
NOPC1	4.4	2.71	-38	NSUBC	0.4	0.29	-27
NOPC2	4	2.71	-38	NSUBC*	0.4	0.29	-27
NASC	1.2	2	+67	NMSC	4	4	0
CBC	1.6	2.14	+34	NMRC	4	4	0
DIT	0.4	0.29	-27				

V1*: The values before using RoleOf relationship
V2*: The values after using RoleOf relationship

4 Another Example: Pattern Overlapping

In this section, we take *Composite* pattern and *Strategy* pattern [1] as the example, analyzing how to resolve the pattern overlapping and traceability problem, and how to improve the reusability of pattern instance model.

[16] introduces a grade recording system applied to school or company: including the business classes denoting *teacher*, *student*, and *lecture* as well as *test* etc. Teachers can make use of report, written examination, or oral examination and so on to test. This structure can be presented with *Composite* pattern. In addition, different teachers have different grade calculating methods. Even to the same teacher, different courses may have different grade calculating methods. *Strategy* pattern can be used to model the structure. The class model of the system is shown in Fig. 6 (The figure comes from Literature [16]). *Text* class is bound with *Component* role of *Composite* pattern, as well as the *Context* role of the *Strategy* pattern at the same time, including not only the business logic implemented by *compute()* method, but also the pattern logic: *add()* method in the *Component* role and *setAlgorithm()* method in the *Context* role.

In Fig. 6, the business logic and pattern logic are not differentiated, and because *Test* class is bound with two roles in two patterns, that which method belonging to which pattern is not pointed out though the roles of *Test* class in the two patterns are identified.

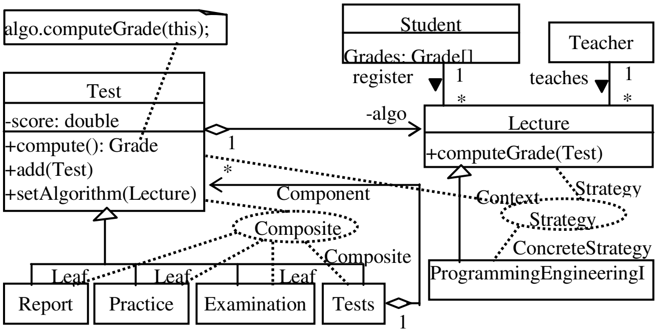


Fig. 6. The grade recording system based on *Composite* pattern and *Strategy* pattern

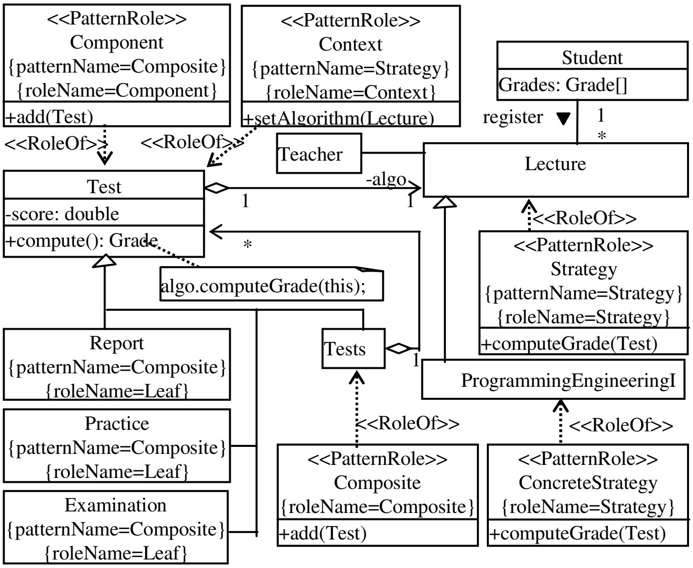


Fig. 7. The grade recording system based on RoleOf relationship

Class model after using RoleOf relationship is shown in Fig. 7. From the figure it can be observed that *Test* class contains two RoleOf relationships, which plays not only *Component* role of *Composite* pattern but also *Context* role of *Strategy* pattern. Moreover, after the pattern logic is separated from the *Test* class, *Test* class becomes lighter, only containing the business logic. This can enhance the reusability of business logic and pattern logic.

After analyzing the model in Fig. 6 and Fig. 7 with the method in section 3.4, we get the result in Table 2. From the result, the conclusion similar to the quantitative evaluation in section 3.4 can be gained.

Table 2. The values of metrics and change rate of example introduced in Fig. 6 and Fig. 7

Metrics	V1*	V2*	Percentage	Metrics	V1*	V2*	Percentage
NATC1	0.56	0.36	-36	NSUPC	0.56	0.35	-38
NATC2	0	0	0	NSUPC*	0.56	0.35	-38
NOPC1	1.89	0.71	-62	NSUBC	0.56	0.35	-38
NOPC2	1.89	0.71	-62	NSUBC*	0.56	0.35	-38
NASC	0.89	1.21	+36	NMSC	0.89	0.89	0
CBC	1	1.29	+29	NMRC	0.67	0.67	0
DIT	0.56	0.35	-38				

V1*: The values before using RoleOf relationship
V2*: The values after using RoleOf relationship

5 Discussion

5.1 Description of the Pattern Element Without Pattern Logic

The constitution elements of the majority of patterns all contain the pattern logic, for example *Subject* class in *Observer* pattern contains *addObserver()* and *notify()* methods, *Observer* class contains *update()* method etc. These pattern logics are necessary for the pattern and they constitute the pattern protocol [2]. To these pattern elements with the pattern logics, they are presented by the role classes.

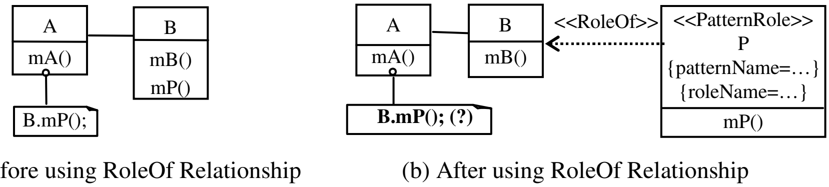
There are some pattern elements which don't contain the pattern logic, for example, though *Component* class and *Composite* class of *Composite* pattern contain not only pattern logic(*add* and *remove* method) but also business logic, *Leaf* class only contains the business logic. To the elements without the pattern logic, it is not necessary to define them as the role class because the reusability problem of the pattern logic doesn't exist. But it is necessary to point out which role of which pattern this business class plays with the tagged values *pattrnName* and *roleName* in the corresponding business class. See *Report*, *Practice* and *Examination* class in Fig. 7.

5.2 Invocation of the Pattern Logic

Different implementation methods of RoleOf relationship may affect the invocation method of the pattern logic. As Fig. 8 shows, before using RoleOf relationship, method *mA()* of class *A* invokes the pattern logic *mP()* of class *B*. After using RoleOf relationship, how does method *mA()* invoke *mP()*?

If using static implementation, it is legal to invoke *B.mP()*, because method *mP()* defined in aspect will be inserted to class *B* at compile time. If making use of dynamic implementation, it is illegal to invoke *B.mP()*, because the instance of *B* and the one of role *P* are bound dynamically at run time, and the type checking error will be reported at compile time.

In addition, the semantics of RoleOf relationship indicates that role class can't exist alone and it must be related to business class with RoleOf relationship. Therefore no matter static or dynamic implementation method is used, the methods in the role class can't be invoked directly. For example, it is illegal to invoke *P.mP()*. Any invocation of the pattern logic must be done through the associated business class.



(a) Before using RoleOf Relationship (b) After using RoleOf Relationship

Fig. 8. Invocation of the pattern logic

5.3 Role Class and Aspect

Role class and Aspect are alike in the structure. A role class is a UML class identified by stereotype <<PatternRole>>, while Aspect, as same as the standard UML class, is a container which may contain attributes, methods, pointcuts, advices and introductions and so on. They are common logics separated from the business class and can't exist alone. Furthermore, role class can be defined as Aspect and be woven into the business class. The main differences between role class and Aspect are:

- A role class denotes a role in the pattern and the collaboration relation exists between the roles. Although such collaboration relation is not expressed explicitly between the role classes, it exists between the corresponding business classes; Aspect focuses only on how to improve the modularity of the system. So the conception of the role can describe the relation between the system and the used design pattern relevantly, such as the role played by the business object, collaboration between the business objects etc. Those relations are just the structures and behaviors of the pattern instances needing to be described.
- The main purpose of using Aspect is to make use of advice to modify the behavior of the base objects, such as adding logging and transaction function etc, putting an emphasis on changing the behavior of a certain method (though the *introduction* mechanism can be used to insert the attribute and the method into the base object). The role class emphasizes on role and behavior played by the business object not on modifying the behavior of a certain method of the business object.
- RoleOf relationship is based on the concept of role. Besides using Aspect implementation, it can also use some dynamic implementation methods of the role, which can realize the dynamic binding and separation between the business object and the role, improving the reusability of the pattern logic furthest. But dynamicity is not mandatory for Aspect [7].

6 Related Work

A design pattern composition method is proposed in [4], which uses the UML extension mechanism– tagged value to identify the information of instantiated pattern. The method can mark the pattern instantiation information definitely, and has good traceability, but some drawbacks still existing: the business logic and the pattern logic are not separated. When a business class is involved in multiple patterns, although the tagged values can be used to distinguish the patterns, it is difficult to reuse this class because it contains too much functionality. In our approach, business and pattern logics are separated completely to improve their reusability.

Yacoub proposes Pattern-Oriented Analysis and Design (POAD) [5] that uses UML to compose design pattern in different abstract levels. POAD defines three level logic views: Pattern-Level, Pattern Interfaces and Detailed Pattern-Level views. The interior detailed construction to the pattern is hidden in the high level and described in the low level. This method focuses on the pattern-oriented system developing process, and doesn't address the problem of how patterns can be combined with parts of design that are not expressed as patterns [5].

[17] proposes composition patterns for the reusable Aspect design. The method defines pattern as template package and binds it with application package to produce a composition output. Although this approach can reuse composition patterns in design level and improve the traceability of the design, it's not possible to reuse roles in pattern respectively because its reusable unit is subject. Our approach defines roles of pattern as role classes, which can reuse role class respectively.

[18] proposes a visual modeling language - DPML(Design Pattern Modeling Language). This language provides a set of expression approaches to the pattern modeling and instantiation. For not according with UML standard, the approach is difficult to be spread and can't support the pattern overlapping well [18]. Our approach is based on the standard UML, and can be easily integrated into existing UML tools.

7 Conclusion

When Instantiating design patterns in design level, designers are faced with problems, such as reusability, traceability and overlapping problem. In this paper, we present that RoleOf Relationship can provide a general approach resolving these problems.

We present a static implementation of RoleOf relationship, this method is intuitive and easy to realize. But because every RoleOf relationship needs to define different aspects, the reusability of Aspect is not realized furthest in implementation level. As the improved solution, parametric introductions [19] can be used, which permits defining template parameter in introduction and being bound with the actual parameter when being woven into introduction. This can enhance the reusability of introduction. In addition, using the dynamic implementation method can realize the dynamic binding between business object and role object at run time.

Acknowledgements. This research project was supported by the National Natural Science Foundation of China under Grant No. 60373086; Wuhan Science & Technique Key Project under Grant No. 20021002043; Hubei Province Key Project under Grant No. 902130819.

References

1. Erich Gamma, Richard Helm, John Vlissides, et al: Design Patterns: Elements of Reusable Object Oriented Software. TOKYO:SOFT BANK, PUBLISHING (2001)
2. J. Hannemann, G. Kiczales: Design Pattern Implementation in Java and AspectJ. Proc. of the ACM Conf. on Object-Oriented Programming Systems, Languages, and Applications (2002) 161-173
3. M. Torchiano: Documenting Pattern Use in Java Programs. IEEE Int. Conference on Software Maintenance (ICSM 2002) (2002)
4. J. Dong: UML Extensions for Design Pattern Compositions. J. of Object Technology, vol. 1, no. 5 (2002) 149-161
5. Sherif M. Yacoub, Hany H. Ammar: UML Support for Designing Software Systems as a Composition of Design Patterns. In UML 2001 - The Unified Modeling Language. Modeling Languages, Concepts, and Tools, pages 149-165, Springer, LNCS, Vol. 2185 (2001)
6. Object Management Group: Unified Modeling Language Specification. Version 1.4. http://www.omg.org (2001)
7. Stefan Hanenberg, Rainer Unland: Roles and Aspects: Similarities, Differences, and Synergetic Potential. 8th International Conference on Object-Oriented Information Systems (2002)
8. Bent Bruun Kristensen: Object-Oriented Modeling with Roles. In Proceedings of the 2nd International Conference on Object Oriented Information Systems (OOIS'95) (1995)
9. M. Dahchour: Integrating Generic Relationships into Object Models Using Metaclasses[Ph.D Thesis]. D´epartement d'ing´enierie informatique, Universit´e catholique de Louvain, Belgium (2001)
10. Kiczales, G., Lamping, J., Menhdhekar, A., Maeda, C., Lopes, C., Loingtier, J., Irwin, J.: Aspect oriented programming. In: Proceedings of ECOOP'97. Number 1241 in Lecture Notes in Computer Science, Springer Verlag (1997) 220-242
11. AspectJ Team: The AspectJ^{TM} Programming Guide. http://eclipse.org/aspectj/.
12. D. Stein, S. Hanenberg, R. Unland: A UML-based Aspect-Oriented Design Notation For AspectJ. 1st Int. Conf. on Aspect-Oriented Software Development(AOSD) (2002)
13. Lorenzo Bettini, Sara Capecchi, Betti Venneri: Extending Java to dynamic object behaviors. Electronic Notes in Theoretical Computer Science 82 No. 8 (2003)
14. HE Cheng-Wan, HE Fei , HE Ke-Qing: A Dynamic Object Behavior Model and Implementation Based on Computational Reflection. Wuhan University Journal of Natural Sciences. Vol.10, No.2 (2005) 358-362
15. H. Kim, C. Boldyreff: Developing Software Metrics Applicable to UML Models. 6th ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (QAOOSE 2002) (2002)
16. Y. Sanada, R. Adams: Representing Design Patterns and Frameworks in UML-Towards a Comprehensive Approach. J. of Object Technology. Vol. 1, No. 2, (2002)
17. S. Clarke, R.J. Walker: Composition Patterns: An Approach to Designing Reusable Aspects. Proc. of the 23rd International Conference on Software Engineering (2001) 5-14
18. David Mapelsden, John Hosking, John Grundy: Design Pattern Modelling and Instantiation using DPML. In Proceedings of the Tools Pacific 2002. CRPIT Press (2002)
19. Stefan Hanenberg, Rainer Unland: Parametric Introductions. 2nd Int' Conf. on Aspect-Oriented Software Development (AOSD-2003), ACM Press (2003) 80-89

Automatic Inspection of Industrial Sheetmetal Parts with Single Non-metric CCD Camera

Yongjun Zhang

School of Remote Sensing and Information Engineering, Wuhan University, China
zhangyj@whu.edu.cn, yongjun_zhang@hotmail.com

Abstract. A novel approach for three-dimensional reconstruction and inspection of industrial parts with image sequence acquired by single non-metric CCD camera is proposed. The purpose of the approach is to reconstruct and thus inspect the producing imprecision (of deformation) of industrial sheetmetal parts. Planar control grid, non-metric image sequence and CAD-designed data are used as information sources. Principles of least squares template matching to extract lines and points from the imagery are presented. Hybrid point-line photogrammetry is adopted to obtain the accurate wire frame model. Circles, connected arcs and lines on the part are reconstructed with direct object space solution. The reconstructed CAD model can be used for inspection or quality control. Experimental results are very satisfying.

1 Introduction

Nowadays, Computer Aided Design (CAD) is widely used in industries. Most industrial parts have their CAD-designed data. Precision evaluating and quality control of parts with reference to CAD data receive attention in industrial communities. Reducing manpower, maintaining high precision and consistency and the time of inspection are the main foci of researchers.

Along with the development of computer vision, automated two-dimensional (2D) visual inspection has been widely used in Printed Circuit Board product lines (Choi 2003). Stereo vision technique with two CCD cameras and two infrared LED lamps is used by (Kosmopoulos 2001) in inspection of gaps on the automobile product line. Precision of 0.1mm within a planar area of 80mm × 80mm is obtained. Although automated vision metrology getting more and more mature (c.f. Fraser 1999, Chang 2001), three-dimensional (3D) automatic inspection has been limited due to the complexity of the problem. Precission of about 10 to 20 ppm can be achieved by the V-Stars system (GSI 2003). However, special targets have to be attached onto the surface of the interested object, which makes the system les efficient. Precision of 0.07mm for inspection of industrial parts has been achieved by Zhang (Zhang 2004).

The general purpose of this paper is to inspect the industrial sheetmetal parts automatically and accurately through CAD data and information extracted from the imagery. A planar grid is used to calibrate the CCD camera and provide initial values of camera parameters during reconstruction. World coordinate system is chosen the same as that of the grid. CAD-designed data represents the initial model and the topology of the part. Least-Squares Template Matching (LSTM) is discussed in

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 654 – 661, 2005.

section 2. Afterwards, detailed approaches of how to reconstruct industrial parts are presented. The reconstructed CAD model can be used to inspect the producing imprecision or deformations of the part. Section 4 discusses the experimental results. Conclusions and future work are outlined in section 5.

2 Least Squares Template Matching

2.1 Two-Dimensional Line Template Matching

Image points and lines are the most effective features for 3D reconstruction of objects. If "Minimization of the squared sum of grey-difference" is chosen as the criteria, the image matching equation is $\sum vv = \min$. This is the basic principle of least squares template matching (Schenk 1999). Line template matching is a 2D technique that attempts to match a standard template with a real image patch. The real image patch is rotated into horizontal to facilitate matching. As shown in Fig. 1, the level rectangle represents the standard template, while the dashed rectangle is the image patch to be matched. Two unknowns dy_1 and dy_2 are essential to fit the small rotation angles between the image patch and the template.

The line template matching technique can be used for the extraction of image line features. Grid points are detected as the intersection of two matched lines of each corner, as shown in Fig. 2. The black crosses are the predicted image corners, and the white crosses are the matched ones. The precision of image matching results is higher than 0.05 pixels. Light blue lines in Fig. 3 shows the initial projections of several line segments of the part. Although rust exists on the part and can be seen clearly, the matched image lines are well fitted to the real image features (green lines in Fig. 3). Actually, the matching precision is also higher than 0.05 pixels.

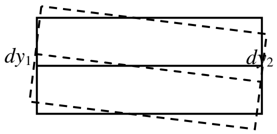


Fig. 1. 2D line template matching

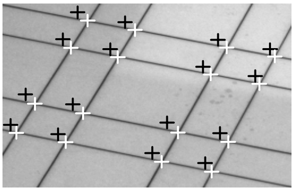


Fig. 2. Matching of points

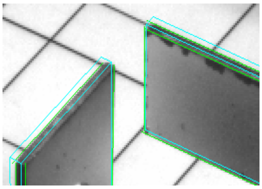


Fig. 3. Matching of lines

2.2 One-Dimensional Point Template Matching

A line can be represented by a group of small colinear segments. If the image window of line template matching is subdivided into small segments, each with a length of 2-5 pixels, named point segment, the rotation angle between standard template and small point segment can be neglected. Matching between the point segment and the template is called point template matching in this paper. As it differs from line template matching, angle is assumed to be not existed, since the length of point segment is usually very short. There is only one unknown *dr* for one-dimensional (1D) point template matching, as shown in Fig. 4.

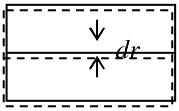


Fig. 4. 1D point template matching

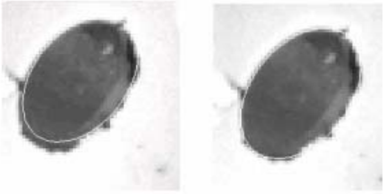


Fig. 5. Matching of circle

Fig. 5 shows the initial projection and matched result of a circle by 1D point template matching. Although rust is evident, the matched circle is well fitted to the image, showing the well potential of 1D point template matching.

3 Reconstruction of Industrial Sheetmetal Parts

The topology of CAD data of sheetmetal part is assumed to be correct since it is designed on computer and often checked many times before the part is produced. But the geometry of the sheetmetal part is usually not identical to the CAD-designed data due to mis-operation during producing. The correct geometric model is reconstructed by two steps, firstly the wire frame model is obtained and then the complex shapes.

3.1 Reconstruction of Wire Frame Model

The world coordinate system is chosen the same as that of the grid. Generally speaking, the coordinate system of the part defined in CAD-designed data will not be identical to the world coordinate system. There are six elements (rotation and translation) to convert the CAD coordinate system into the world coordinate system.

Sheetmetal parts are mostly composed of line segments. This is the reason that we choose line photogrammetry to reconstruct and inspect them. A image line *pq*, space line *PQ* and the projection center *S* should be coplanar (Heuvel 1999, Zhang 2004), while *p* and *P*, *q* and *Q* are not necessarily correspondences, which is the most important advantage of line photogrammetry. Two end points are used to present a line, because it is singularity-free and easy to setup error equations.

The coplanar equation among *p, S, P* and *Q* is:

$$\begin{vmatrix} u_p & v_p & w_p \\ X_P - X_S & Y_P - Y_S & Z_P - Z_S \\ X_Q - X_S & Y_Q - Y_S & Z_Q - Z_S \end{vmatrix} = 0 \tag{1}$$

where (u_p, v_p, w_p) is the model coordinate of image point *p*, (X_S, Y_S, Z_S) the coordinate of camera center *S*, (X_P, Y_P, Z_P) and (X_Q, Y_Q, Z_Q) the coordinates of points *P* and *Q*. The error equation of line photogrammetry can be written as (Zhang 2004):

$$\begin{aligned}&A_1 dx_p + A_2 dy_p + A_3 d\varphi + A_4 d\omega + A_5 d\kappa + A_6 dX_S + A_7 dY_S + A_8 dZ_S +\\ &A_9 d\varphi^0 + A_{10} d\omega^0 + A_{11} d\kappa^0 + A_{12} d\Delta X^0 + A_{13} d\Delta Y^0 + A_{14} d\Delta Z^0 + A_{15} dX_P^0 +\\ &A_{16} dY_P^0 + A_{17} dZ_P^0 + A_{18} dX_Q^0 + A_{19} dY_Q^0 + A_{20} dZ_Q^0 + F_X = 0\end{aligned} \tag{2}$$

where $A_1 \sim A_{20}$ are the coefficients of unknowns and F_X the constant item. Besides the coplanar equation among *p, S, P* and *Q*, there exits another equation among *q, S, P* and *Q*. The linearized form is similar to that of equation (2).

For parts that are very simple or for those that have a few line segments, grid points should be combined into the adjusmtent model to ensure the reliability of reconstruction. The error equations of grid points are (Schenk 1999):

$$\begin{aligned}v_x &= B_1 d\varphi + B_2 d\omega + B_3 d\kappa + B_4 dX_S + B_5 dY_S + B_6 dZ_S + B_7 dX + B_8 dY + B_9 dZ - l_x\\ v_y &= C_1 d\varphi + C_2 d\omega + C_3 d\kappa + C_4 dX_S + C_5 dY_S + C_6 dZ_S + C_7 dX + C_8 dY + C_9 dZ - l_y\end{aligned} \tag{3}$$

where l_x, l_y are constant items, B_1, ... ,B_9, C_1, ... ,C_9 coefficients of unknowns. If the coordinates of grid points can be treated as known, terms of (dX, dY, dZ) should be removed. The model of hybrid point-line photogrammetry is composed of equation (2) and (3) and can be used to reconstruct the wire frame model of industrial part.

3.2 Reconstruction of Complex Shapes

For lots of board-like industrial parts especially sheetmetal parts, the reconstruction of complex shapes is also very important but hard to deal with in practice. Up to now, there is few publication or system that can automatically reconstruct circles, connected arcs and lines without attaching any marks on the surface.

Camera parameters, which can be obtained with hybrid point-line photogrammetry, are treated as known. The end points of small line segments are the result of template matching and also functions of circles or lines. Parameters of circles or lines and image features are related by mathematical models. Thus parameters of circles, arcs and lines can be obtained from several images by least squares template matching.

To facilitate the reconstruction, suppose the plane where the circle or arc lies in is known. The camera parameters of the images can be rotated to generate a level plane. So the circle equation in the level plane is very simple:

$$\begin{aligned}X &= X_0 + R \cdot \cos\theta\\ Y &= Y_0 + R \cdot \sin\theta\end{aligned} \tag{4}$$

where X_0, Y_0 and *R* are the center and radius of circle or arc, θ varies from 0 degree to 360 degree for circle, and from start angle to end angle for arc. In this paper, circles and arcs are represented by a number of points with different angle θ. The top of Fig. 6 is a space circle, and the bottom is the projected ellipse with known camera parameters. Each point *A* on the space circle defined by θ has its corresponding point *a* in the image. If equation (4) is substituted into collinearity equations, the unknowns are the center and radius of circle or arc. For certain angle θ, the object point is projected onto image and the tangential vector with angle α can be easily determined. The displacement *dr* determined by template matching can be rotated back to the image coordinate system according to α. The error equations of circle or arc reconstruction can be written as:

$$\begin{aligned} v_x &= A_1 \cdot dX_0 + A_2 \cdot dY_0 + A_3 \cdot dR - dx \\ v_y &= B_1 \cdot dX_0 + B_2 \cdot dY_0 + B_3 \cdot dR - dy \end{aligned} \tag{5}$$

where A_1, A_2, A_3, B_1, B_2, B_3 are the coefficients of unknowns, *dx, dy* constant items. Thus the parameters of space circles or arcs can be obtained directly from one or several images. The obtained parameters of circles or arcs should be rotated back to the world coordinate system according to camera parameters.

Arcs in parts are usually connected to lines. As shown in Fig. 7, two arcs c_1, c_2 and three line segments l_1, l_2 and l_3 are connected to each other. For convenience of reconstruction, line segments are also rotated into a level plane:

$$\begin{aligned} X &= X_s + i \cdot \Delta L \cdot \cos\beta \\ Y &= Y_s + i \cdot \Delta L \cdot \sin\beta \end{aligned} \tag{6}$$

where X_s, Y_s is the start point of line segment, β the direction of the line, ΔL the length of small segment approximately equal to the length of point window in circle and arc matching. The error equations of line reconstruction are:

$$\begin{aligned} v_x &= M_1 \cdot dX_s + M_2 \cdot dY_s + M_3 \cdot d\beta - dx \\ v_y &= N_1 \cdot dX_s + N_2 \cdot dY_s + N_3 \cdot d\beta - dy \end{aligned} \tag{7}$$

where M_1, M_2, M_3 and N_1, N_2, N_3 are coefficients of unknowns, *dx, dy* constant items. The circle or arc reconstruction equation (5) can be combined with line reconstruction equation (7) to get solution of connected arcs and lines. To ensure the stability of reconstruction, geometric constrains should be added, such as the center of arc c_1 should lie on the bisector of line $l_1$and l_2, the center of arc c_2 should lie on the bisector of line $l_1$and l_3 and three lines should be tangential to the two arcs, etc.

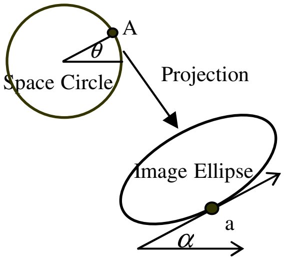


Fig. 6. Projected image point of circle

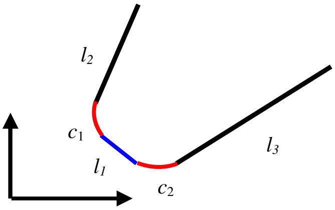


Fig. 7. Connected arcs and lines

4 Experiments

4.1 Overview of the System

To reduce the cost of the inspection system, only one non-metric CCD camera with 1300×1030 pixels resolution is used. Fig. 8 shows the hardware configuration of the system. A planar grid is fixed on the rotation table. The part to be reconstructed is put approximately on the center of the grid. To generate diffused illumination for the grid and the part, four home-used lamps are encapsulated in a semi-transparent plastic box. Image sequence is obtained while the table rotates under computer control.

The developed software runs fully automatically. It can be used to reconstruct and inspect the imprecision of industrial parts mainly composed of lines, circles, connected arcs and lines. The system is composed of 4 steps. Firstly, Image sequence is acquired by CCD camera while the table turns around its center controlled by computer. Image points and lines are obtained by LSTM simultaneous with image acquiring. Then 3D wire frame model of the part is reconstructed. Afterwards, circles, connected arcs and lines are reconstructed by direct object space solution. Finally, inspection can be done automatically or interactively.

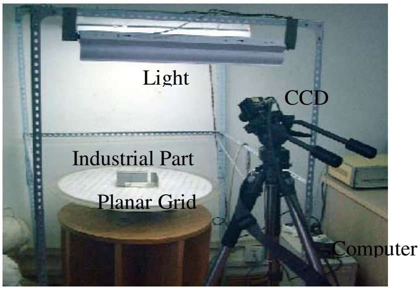


Fig. 8. Hardware of the inspection system

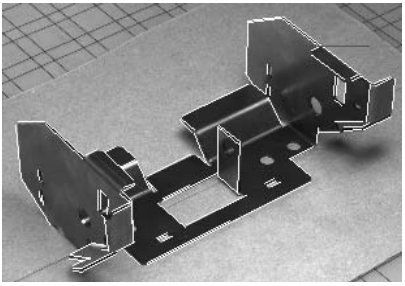


Fig. 9. One image of the part to be inspected

4.2 Real Data Experiments

The inspection system has been tested with real image data of many parts taken by a pre-calibrated CCD camera (Zhang 2003). Experiment of a part with dimension of about 150mm×100mm×80mm will be presented. The part to be reconstructed is put on the center of the planar grid, which is fixed on the turntable. The CCD camera is fixed on a tripod 600mm away from the part. A sequence of 25 images for the part is taken with equal angle intervals while the table turns around, one image is shown in Fig. 9. Image matching is made simultaneous with image acquiring. Grid points are detected as the intersection of two line segments fitted to each corner. Lines are obtained by LSTM with initial values projected by the CAD-designed data.

White lines in Fig. 9 are the matched lines of the part. There is nearly no mismatch for points of planar grid. But for part that are very thin, there maybe some mismatched lines. Most of them can be removed by trifocal tensor (Hartley 2000) computed with camera parameters. The remained mismatches can be eliminated by the iterative least squares adjustment. The system can generate a final CAD model for each industrial part within 3 minutes (includes image acquiring) in a PIV personal computer.

In order to evaluate the precision of the inspection system, 25 distances between lines and planes on the part are measured by calipers and compared with which computed by the reconstructed model. In Fig. 10, "Producing imprecision" means distances between lines, planes or line to plane measured by calipers subtracting the corresponding designed distances. "Computed imprecision" means distances computed with the reconstructed model subtracting the designed distances.

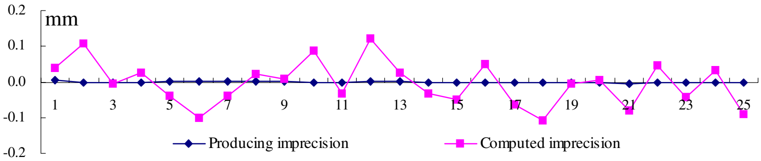


Fig. 10. Inspecting result of the fisrt part

It can be seen from Fig. 10 that the measured distances are very close to that of the CAD data, i.e., the producing imprecision is nearly zero. Deviations of computed imprecision show a well normal distribution. The root mean square (RMS) error of deviation is 0.067mm. The relative precision, which can be calculated as the ratio of RMS against the distance between the camera and the part, is higher than 1/8500 (0.067mm/600mm = 1/8900), which shows the precision of the proposed system when manually measured distances are treated as errorless.

The proposed circle reconstruction approach is also tested with several real image data. Left of Fig. 11 shows the projection of a circle with CAD designed data and camera parameters obtained from hybrid point-line photogrammetry. As can be seen, the rust is very clear. The reconstructed projection (right of Fig. 11) is well fitted with the image feature. The diameter of reconstructed circle is 9.952mm, very close to 10.00mm that measured by calipers.

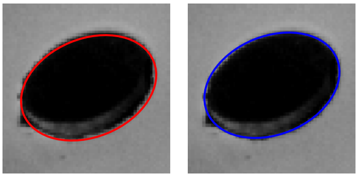


Fig. 11. Reconstruction of circle

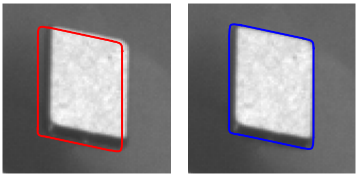


Fig. 12. Reconstruction round rectangle

Left of Fig. 12 shows the initial projection of a round rectangle. The length of the short line segments is only about 12 pixels in the image, and even a few pixels for arcs. They are very difficult to reconstruct with common strategies. But for the proposed technique of object space solution with geometric constraints, they can be reconstructed successfully and stably. The projection of reconstructed model is also well fitted to the image feature (right of Fig. 12).

5 Conclusion

An effective approach for 3D reconstruction and inspection of industrial parts mainly composed of line segments, circles, connected arcs and lines with non-metric image sequence and CAD data is proposed. Wire frame model of industrial part can be reconstructed with hybrid point-line photogrammetry. Circles, connected arcs and lines are reconstructed by point template matching and direct object space solution.

A relative precision of higher than 1/8500 has been obtained. As can be seen, the precision of our system is higher than that of Kosmopoulos (2001), which can achieve a precision of 0.1mm within an planar area of 80mm × 80mm. The proposed technique also has the advantages of low cost of hardware and fully automatic. It shows a promising potential in automatic 3D reconstruction and inspection of industrial parts mainly composed of lines, circles, connected arcs and lines.

However, the proposed system still requires further improvements. Relation between illumination condition and precision of measurement has to be analyzed. Furthermore, the main limitation of the proposed system is that there must be point and/or line features on the surface of the industrial part. Otherwise, artificial patterns have to be projected onto the surface.

Acknowledgement. This work is supported by National Natural Science Foundation of China with project number 40301041.

References

1 Chang M., Fuh C., 2001. Fast search algorithms for industrial inspection. International Journal of Pattern Recognition and Artificial Intelligence. Vol. 15, No. 4: 675-690.

2 Choi K. -S., Pyun J. -Y., Kim N. -H., et al, 2003. Real-time inspection system for printed circuit boards. Lecture Notes in Computer Science, Vol. 2781: 458-465.

3 Fraser C., 1999. Automated vision metrology: a mature technology for industrial inspection and engineering surveys. 6th South East Asian Surveyors Congress Fremantle, Western Australia, 1-6 November 1999

4 Hartley R., Zisserman A., 2000. Multiple view geometry in computer vision. Cambridge University Press, UK.

5 Heuvel F. A., 1999. A line-photogrammetry mathematical model for the reconstruction of polyhedral objects, Proceedings of SPIE, Vol. 3641: 60-71.

6 GSI, 2003. V-STARS main brochure. http://www.geodetic.com/vstars.htm.

7 Kosmopoulos D, Varvarigou T., 2001. Automated inspection of gaps on the automobile production line through stereo vision and specular reflection. Computers in Industry, Vol. 46: 49-63

8 Schenk T., 1999. Digital photogrammetry. TerraScience, USA.

9 Zhang Y., Zhang Z., Zhang J., 2003. Camera calibration technique with planar scenes. Proceedings of SPIE. Vol. 5011: 291-296.

10 Zhang Y., Zhang Z., Zhang J., 2004. Deformation visual inspection of industrial sheetmetal part with image sequences. Machine Vision and Applications. Vol. 15, No.3: 115-120.

An Advanced Implementation of a Distributed Control Scheme Based on LonWorks System over IP Networks

Il-Joo Shim[1], Kyung-Bae Chang[1], Ki-Hyung Yu[2], Dong-Woo Cho[2],
Kyoo-Dong Song[3], and Gwi-Tae Park[1]

[1] School of Electrical Engineering Korea University,
1, 5-ka, Anam-dong, Seongbuk-gu, Seoul, 136-701, South Korea
[2] Building Research Div., Korea Institute of Construction Technology, 2311, Daehwa-Dong,
Ilsan-Gu, Koyang-Shi, Kyonggi-Do, South Korea, 411-712
[3] Major in Architectural Engineering, Hanyang University, 1271, Sa-1-Dong, Ansan,
Kyonggi-Do, South Korea, 425-791
{ijshim, lslove}@korea.ac.kr, {raytrace, dwcho}@kict.re.kr,
kdsong@hanyang.ac.kr, gtpark@korea.ac.kr

Abstract. In this paper, an advanced distributed control scheme that connects the control networks to the IP networks, based on LonWorks technology, which is one of the control networks, are presented. The proposed approach is implemented by using a simple programmable basic Lon node (BLN) and a IBM PS/2 (PC) compatible computer as an Internet server. BLN is a developed electric board in this research and it physically contains a transceiver, Neuron chip and some memory devices. To perform various functions as an Internet server of PC, control software of Lon on internet system (LOIS) with C-language for GNU/LINUX environment is also developed. Our approach makes system designers to easily implement their various specific applications, only with the download of a control program from serial port (RS-232) of PC.

1 Introduction

These days, many control networks systems have been distributed, and LonWorks technology by Echelon Corporation is one of the most promising solutions due to its intelligent and decentralized network nodes [1]. LonWorks technology uses the LonTalk as its communication protocol, which supports a seven-layer model of open system interconnection (OSI, ISO 7498) network protocol [4]. However, since LonTalk is different from IP networks' protocol, for the connection of LonWorks system to the Internet, LonTalk adapter merges two other protocols, with devices such as iLon. However, this application is restrictive and highly expensive, therefore a newly flexible and cost effective approach is demanded.

To serve the problems mentioned above, an advanced programmable method that connects the control networks to the IP networks based on LonWorks technology is presented. The contents of this paper are as follows. Firstly, a brief description of the LonWorks technology as a control network and its connection to the Internet are briefly described in Section II. In Section III, the proposed programmable hardware, BLN, and software, LOIS, are described in detail, and an advanced implementation of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 662 – 669, 2005.

a distributed control scheme based on LonWorks system, using BLN and LOIS is presented. In this section, to demonstrate the proposed approach, implementations of two operation modes are also included. Finally, conclusions are given in Section IV.

2 LonWorks Technology: Distributed Control Network

Control networks are designed for the specific requirements of control system such as controlling and monitoring for the various kinds of facilities in buildings or factories. The objectives of control networks are the efficient control and management of overall network equipments. These days, the Internet is spread world widely, and if the con-trol networks are to be connected to data network of the Internet, more progressive solutions for intelligent remote control can be realized. Since LonTalk is different from IP network protocol, LonTalk itself cannot support IP network automatically without a network adapter for merging two protocols such as iLON1000 by Echelon Corporation. By using a network adapter, LonWorks network can be easily connected to the IP networks, and its simple connection diagram is shown in Fig. 1. However, when using this adapter, the following weak points are to be considered:

i. Expensive charge: Network adapters are generally expensive.
ii. Narrow application band: Network adapters are inflexible hardware implemented devices, so that wide band applications are restrictive.
iii. Waste of IP address: One adapter must have its own address.
iv. Technical dependency: System designer can not know the inner structure of a network adapter, and it is difficult to extend the function of this adapter.

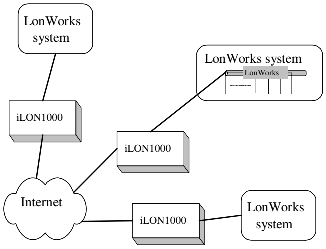


Fig. 1. Connection diagram of control network to the internet by iLon1000

In this paper, to serve the problems mentioned above, an advanced new approach that flexibly connects LonWorks control networks to the IP networks is presented. First of all, to remove the technical dependency on a particular vendor for the network adapter, a programmable solution is proposed, and it can be easily and flexibly applied to any control network. And a IBM PS/2 compatible computer (PC) is engaged as an internet server. Because PCs can be easily connected to the Internet with its own IP address, extra cost for another IP address is not needed. To accomplish the connection between LonWorks system and the Internet server of PC, a programmable BLN, which contains a transceiver and Neuron chip is implemented. The BLN is a simple electric board for the connection LonWorks to the IP networks, and can replace the expensive network adapter such as iLon device.

3 An Advanced Implementation of Distributed Control Scheme Based on LonWorks System

An advanced implementation of the distributed control scheme base on the LonWorks system is accomplished by using BLN, IBM PS/2 compatible PC (internet server) and LOIS which is the operation software on the PC. And then, an advanced distributed control scheme, based on LonWorks system is implemented by using these BLN and the LOIS on the PC. The physical connection between control network and IP network is accomplished by connecting between serial ports on the programmable device in BLN and RS-232 port in PC.

3.1 Basic Lon Node (BLN) for LonWorks System

BLN is designed as a control node for the connection of control networks to the Internet. Its main functions are to transmit all of the information on connected devices to Internet servers, or to receive and execute the control instruction from the Internet server. BLN is physically composed of the programmable device, the Neuron chip, to communicate with the internet server, the transceiver for the direct interface to control networks and some memory chips to store data or application program codes. The Neuron chip by Echelon corporation is a large-scaled integrated (LSI) circuit, which performs control and communication functions based on three-microprocessors with 11 I/O ports. In this paper, the programmable device on BLN board is implemented by a Neuron TMPN3150 chip with the external memory devices of EEPROM and RAM for application program that includes the function of connection between LonWorks system and internet server. The Neuron chip approaches to LonWorks system through the transceiver of FTTA-10A with a twisted-pair communication port [2-3]. The universal input & output ports in the Neuron chip, IO8 and IO9, are used to communicate with the Internet server by serial communication, RS-232.

Fig. 2 shows the implemented BLN board, and connection diagram between LonWorks and the Internet server using BLN is shown in Fig. 3. In Fig. 3, LonWorks system contains three controlled devices, however, the number of them can be increased. BLN has logically two operation modes, such as local and global modes as for transmitting and receiving the information on the control network system. In the local operation mode, BLN has no exchange of information with the Internet server. In this mode, the server cannot sense the status of nodes, but there is an advantage that time to communicate with BLN can be removed. On the other hand, in the global mode, various kinds of work can be done through the Internet server operation, but the operation time gets a little longer due to the communication between BLN and the Internet server.

The operation of BLN in the view of software is simply shown in Fig. 4. In this figure, a simple LonWorks system, BLN and the PC is displayed. This LonWorks system includes one switch node and one lamp node, and BLN has two operation modes, the local mode (Fig. 4(a)) and the global mode (Fig. 4(b)).

In switch node, “nv_switch_state” is defined as the output network variable, and “nv_input” is defined in BLN as the input network variable, which is bound with “nv_switch_state”. The switch node transmits the switch state, “nv_switch_state”, to bound input variable in BLN, “nv_input”. In lamp node, “nv_lamp_state” is defined as the input network variable, and “nv_output” is defined in BLN as the output network variable, which is bound with “nv_lamp_state”. The lamp node receives and executes switching instructions for the lamp, “nv_lamp_state”, from bound output variable in BLN, “nv_output”.

3.2 Lon on Internet System (LOIS) on GNU/LINUX

LOIS is the developed operation software on the PC that serves the part as an Internet server in this paper. In LOIS, basic functions of performing the connection between control networks and IP networks are included. System designers can develop any application programs for their own purposes with this LOIS.
The main functions of LOIS can be briefly described in two categories:

1. LOIS has the function for the Internet server to be able to understand the status of control networks and, if necessary, to control connected devices through the BLN.
2. LOIS communicates bi-directionally with LOIS on the other Internet server over the Internet. Namely, LOIS receives the information on its own connected control networks from BLN, and transmits the received information to LOIS on their Internet server, or receives the information that its counterpart transmitted over the Internet. Furthermore, BLN must have the function that properly executes the received instruction or information.

In this paper, for a simple implementation of the function of an Internet server, a IBM PS/2 compatible computer (PC) has been used. Because a PC can be easily connected to the Internet with its own IP address, extra costs for another IP address is not needed. Two main functions, which are mentioned above, are analyzed into the following four threads for specific implementation:

1. Main thread: Controlling the whole flow, such as start, work and end of processing.
2. Serial communication thread: Performing the serial communication between PC and BLN through RS-232 port. This thread transports a specific instruction or receives the status information on LonWorks system from BLN to control any devices connected to the control networks.
3. Server thread: Waiting any requests from LOIS on other PCs over the Internet, and accomplishing the received requests.
4. Client thread: Performing necessary requests for other PCs over the Internet, and executing them if the appropriate response was received from other PCs.

The functional block diagram of LOIS with these four threads is shown in Fig. 5. Development cost is maximally reduced and LOIS is free from any restriction by using free-ware software GNU/LINUX. When LOIS is launched, it receives the status information on the connected LonWorks system through the BLN. That is, the current mode of BLN, names and the numbers of input/output network variables defined in BLN and the information of standard network variable type (SNVT) index are structured to be a data-base. And it preserves the current information about LonWorks system by continuously updating for the status information. LOIS provides the server and client threads simultaneously, and has the function of bi-directional communication with a number of connected PCs on the Internet. In this paper, these server and client threads are implemented by using Berkeley software distribution (BSD) socket on GNU/LINUX environment. BSD is a version of a unix system and from version 4.1, it uses the concept of 'socket' as the application program interface (API) using TCP/IP [6].

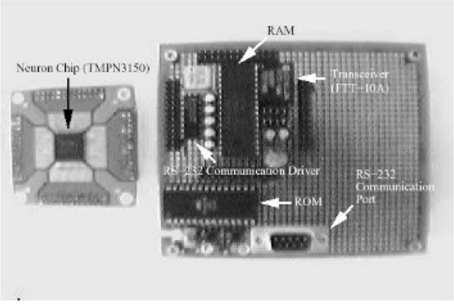


Fig. 2. Neuron device and implemented BLN board

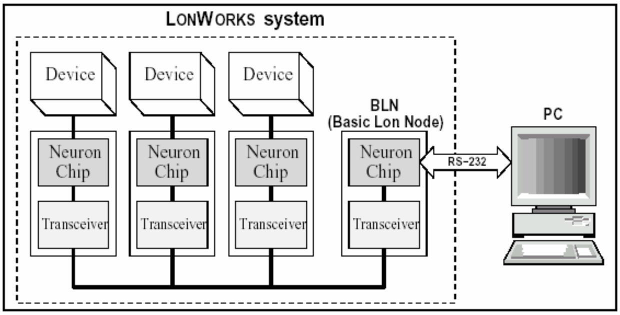
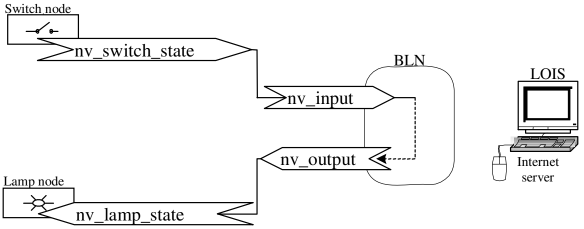


Fig. 3. Connection diagram between LonWorks and internet server using BLN

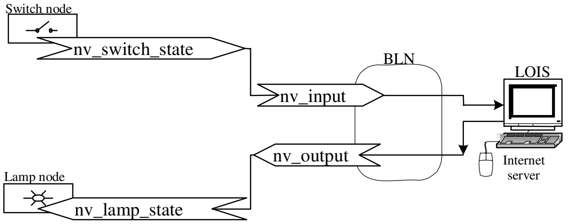
3.3 Implementation of Distributed Control Scheme on LonWorks System

By using BLN, a large number of local control networks can be connected to any PCs as an Internet server, and LOIS operating on PC can access to its own connected control network. Moreover, by imposing specific functions on BLN or LOIS for the system designers, a newly advanced distributed control system can be implemented. Connection diagram of some local control networks on the Internet with LOIS and BLN is shown in Fig. 6. Compared with Fig. 1, the expensive and inflexible network adapters, such as iLon1000 are replaced by the PC with LOIS and BLN.

In this paper, the proposed scheme is demonstrated using simple equipments of three lamps controlled by their power switches.



(a) Local mode in BLN



(b) Global mode in BLN

Fig. 4. Simple block diagram of two operation modes in BLN

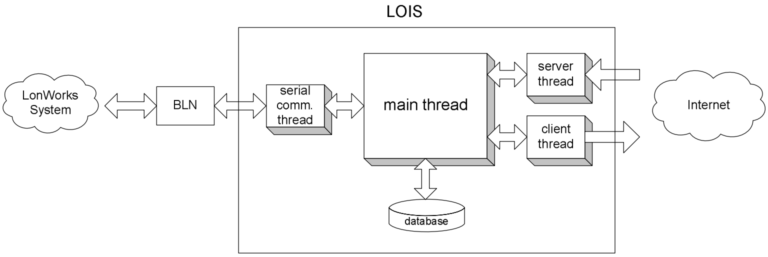


Fig. 5. Structure and functional block diagram of LOIS with four threads

i. Example 1: Global operation mode

In this example, LOIS, which is connected to local LonWorks system via BLN is glob-ally connected to LOIS on another PC over the Internet, and this connection diagram is shown in Fig. 7. In this demonstration of the global operation mode, three lamps and their switch nodes are used same as the example 1, and LonWorks system-1 is remotely controlled over the Internet by LOIS-2 with keyboard-2 input. Fig. 8(a) shows the turn on operation for lamp node-1 in LonWorks system-1 by an operator with key board-2 input at other PC, and Fig. 8(b) and Fig. 8(c) show the turn on operation for lamp node-2, and node 3 over the Internet, respectively.

3 Conclusion

In this paper, an advanced distributed control scheme that connects the control network to the IP networks based on LonWorks system, is presented. LonWorks technology uses the LonTalk as its communication protocol. However, LonTalk is different from IP networks' protocol, and for the connection of LonWorks system to the Internet, this merging solution is high priced, it not easy for system designers to implement their various specific applications. In this paper, the connection between LonWorks system and IP networks is accomplished by using a programmable BLN. To perform various functions as an Internet server of PC, control software, LOIS with C-language for GNU/LINUX environment is also developed. In this paper, the implementation of a distributed control scheme, based on Lon-Works system is demonstrated, using three lamps controlled by their power switches with two operation modes. Finally, with the proposed approach, flexible and cost reduced distributed control systems over the Internet are to be accomplished with a lower cost.

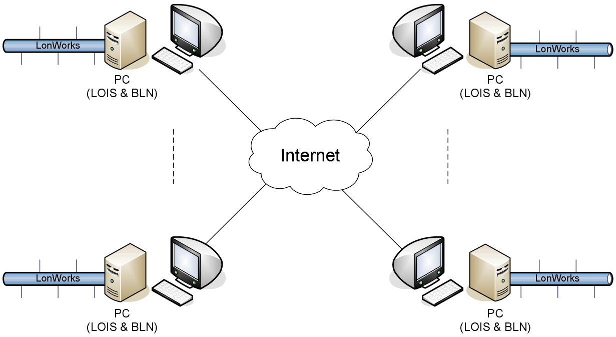


Fig. 6. Connection diagram of control network on the internet by PC with LOIS and BLN

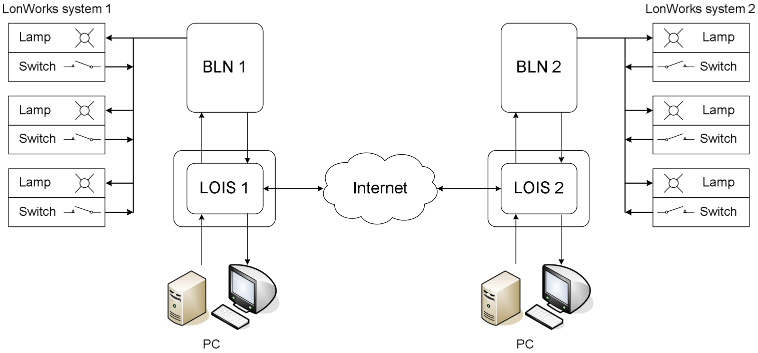
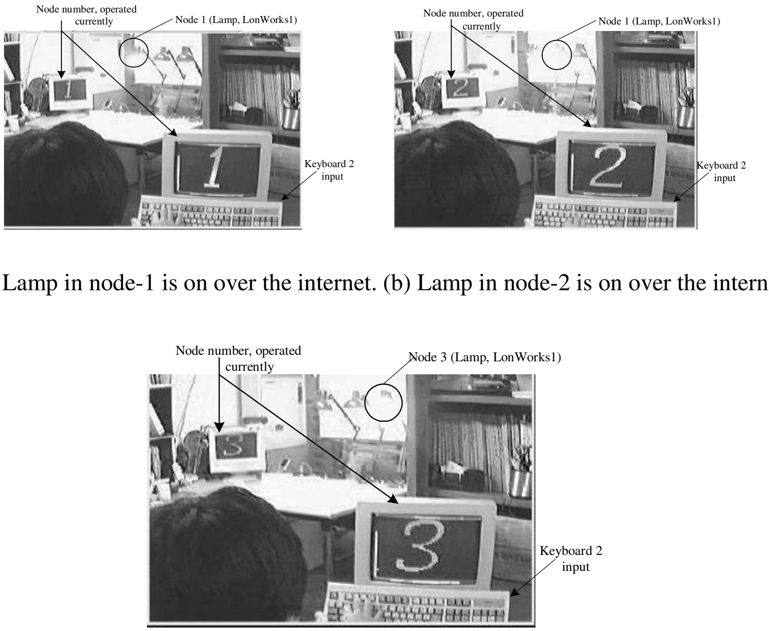


Fig. 7. Simple connection diagram of the global operation mode for three lamps



(a) Lamp in node-1 is on over the internet. (b) Lamp in node-2 is on over the internet

(c) Lamp in node-3 is on over the internet

Fig. 8. Operation results in the global operation mode

References

1. Echelon Corporation. Introduction to the LonWorks system. Echelon Corporation, 1999.
2. Echelon Corporation, LonWorks FTT-10A Free Topology Transceiver User's Guide, Echelon Corporation, 2001.
3. Echelon Corporation, LonWorks PLT-22 Power Line Transceiver User's Guide (110kHz - 140kHz Operation), Echelon Corporation, 1999.
4. ELECTRONIC INDUSTRIES ALLIANCE, EIA STANDARD, Control Network Protocol Specification, EIA-709.1-A, ELECTRONIC INDUSTRIES ALLIANCE, Engineering Department 2500 Wilson Boulevard Arlington, VA 22201, 1999.
5. TOSHIBA Corporation, Neuron Chip, Local Operating Network LSIs (TMPN3150 / TMPN3120), TOSHIBA Corporation, 1999..
6. W. Richard Stevens, UNIX Network Programming, vol. 1 Second Edition. Prentice Hall, 1998.

Structural Damage Detection by Integrating Independent Component Analysis and Support Vector Machine

Huazhu Song[1], Luo Zhong[1], and Bo Han[2]

[1] School of Computer Science and Technology, Wuhan University of Technology,
Wuhan, Hubei 430070, P.R. China
shuazemail@yahoo.com
[2] Center for Information Science and Technology, Temple University
Philadelphia, PA 19122, USA

Abstract. Structural damage detection is very important for identifying and diagnosing the nature of the damage in an early stage so as to reduce catastrophic failures and prolong the service life of structures. In this paper, a novel approach is presented that integrates independent component analysis (ICA) and support vector machine (SVM). The procedure involves extracting independent components from measured sensor data through ICA and then using these signals as input data for a SVM classifier. The experiment presented employs the benchmark data from the University of British Columbia to examine the effectiveness of the method. Results showed that the accuracy of damage detection using the proposed method is significantly better than the approach by integrating ICA and ANN. Furthermore, the prediction output can be used to identify different types and levels of structure damages.

1 Introduction

Structural stiffness decreases due to aging, damages, and other harmful effects. These adverse changes lead to abnormal dynamic characteristics in natural frequencies and mode shapes. By instrumenting structures with a vibration sensor system, structural health monitoring (SHM) aims to provide reliable and economical approaches to monitor the performance of structural systems in an early stage so as to facilitate the decisions on structure maintenance, repair and rehabilitation [1, 11].

In the exciting field, researches have been studied on detect whether or not damage exists in a structure with varied approaches. A comprehensive literature review was made by Doebling and some successful methodologies were shown in his report [4, 5], such as employing changes in the natural frequencies and mode shapes, using measurements of flexibility, constructing statistical model, applying model-updating techniques and artificial neural network. From the view of datasets, these approaches used either time domain data or frequency domain data, measured from sensors in a structure. From the view of constructed models, they applied either physics-based models or data mining models (non physics-based models).

Pothisiri presents a physical damage detection model and its algorithm based on a global response of a structure [12]. This algorithm can locate one or more damaged members in a structure. However, it is not sufficient since it requires that the vicinity

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 670 – 677, 2005.

of the damage is known prior to the experiment and that the portion of the structure is easily accessible. As structures become larger and more complex, this method becomes unfeasible. Domain experts expect more efficient methods.

As the development of data mining techniques, it is possible to classify input signals or discovery patterns from large size of dataset without any prior background knowledge. In SHM, data mining techniques are used to identify the potential damage in a structure by using the variation of the dynamic response continually measured by sensors. Specifically, the first category problem is solved in two steps: 1) feature reduction from measured dynamic sensor data; 2) structure classification based on selected features. For the first step, ICA, mostly used in feature reduction from time series data, is a statistical and computational technique for revealing hidden factors that underlie sets of random variables, measurements, or signals. Recently, Zang [2] applied ICA to model damaged structures. Their results showed that ICA is a more robust method for feature selection and leads to more accurate classification. However, they didn't make deep and detailed analysis on the classification output. For the second step, both artificial neural network (ANN) and support vector machine (SVM) are active classifiers in this area. Especially, SVM, as a powerful kernel-based learning machine [9], has shown practical relevance for classification in various fields, such as object recognition [3], time-series prediction [10].

In this paper, we integrate ICA and SVM together to detect damages. By analyzing the independent components, we extract some features which include the information about the damage level and type. Next, the obtained components are input into SVM to classify structural damage. Our experiments, based on the benchmark data from the University of British Columbia, showed the prediction output can be used to identify different types and levels of structure damages.

2 Methodology

2.1 Feature Selection

ICA techniques provide statistical signal processing tools for optimal linear transformations in multivariate data and these methods are well-suited for feature extraction, noise reduction, density estimation and regression.

From a mathematical view, the ICA problem can be described as follows, each of h mixture signals $x_1(k)$, $x_2(k)$, …, $x_h(k)$ is a linear combination of q independent components $s_1(k)$, $s_2(k)$, …, $s_q(k)$, that is, X = AS where A is a mixing matrix. Now given X, we hope to compute A and S. Obviously, this is a difficult question since both A and S are unknown. Based on the following two statistical assumptions, ICA successfully gains the results: 1) the components are mutual independent; 2) each component observes nongaussian distribution.

The first one is a strong assumption about signals, even stronger than uncorrelated in PCA. It brings two advantages: we can compute the components in any order without considering the involvement of other components; uncorrelated is just partly independent, so PCA can be used as a pre-processing to whiten data and reduce the dimensionality, which greatly simplified the further processing work.

The second assumption is critical to separate signals. We see that gaussian signal looks like a nearly symmetric shape from all angles. If components observe gaussian distributions, by center limit theory, their linear combination of components should be more like gaussian, which becomes more difficult to separate. Hence, nongaussian is a necessary condition to extract components by detecting non-symmetric mixtures.

Specifically, with the second assumption, there are the following solution to solve the signal separation problems:

By $X = AS$, we have $S = A^{-1}X = WX$ (where $W = A^{-1}$). Hence, the task is to select an appropriate W which applied on X to maximize the nongaussianity of components. This can be done in an iteration procedure.

Different ICA algorithms measure nongaussianity by different methods. Some use Kurtosis function: $\mathrm{Kurt}(y) = E[y^4] - 3(E[y^2])^2$, which approaches 0 for a Gaussian random variable; some use negentropy: $\mathrm{negentropy}(y) = H(y_{gauss}) - H(y)$ (H is entropy); some use approximations of negentropy for speeding up the computation: $J(y) = E[y^3]^2 / 12 + \mathrm{Kurt}(y)^2 / 48$.

FastICA algorithm is applied in our application. The non-quadratic function $g(y) = \tanh(a_{1*}y)$ is used to compute nongaussianity. The detailed algorithm steps are listed in [8].

2.2 Support Vector Machine Classifiers

It is known that SVM, proposed by Vapnik in 1995, has been achieving great success on classifying high-dimensional data. In the practice from many engineering fields, its accuracy is even better than neural networks.

Assuming data $D = \{(\vec{x}_i, y_i), i = 1...N\}$ with label $y_i \in \{-1, +1\}$, SVM transformed the attribute $\vec{x}$ into a higher-dimension attribute set $\vec{x}'$, and then separate data by a hyperplane in this hyper space. SVM assume the best linear classifier of the type $f(\vec{x}') = \vec{w}^T\vec{x}' + b = (w_1x_1' + w_2x_2' + ... + w_nx_n' + b)$ is the hyperplane in the middle of the gap (that is, maximize the margin between two classes of samples) shown in Fig.1.

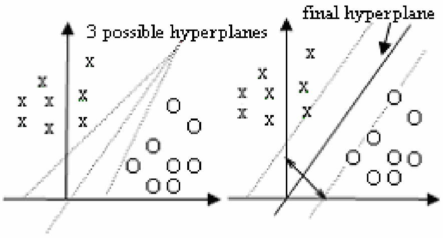


Fig. 1. The hyperplane for classification

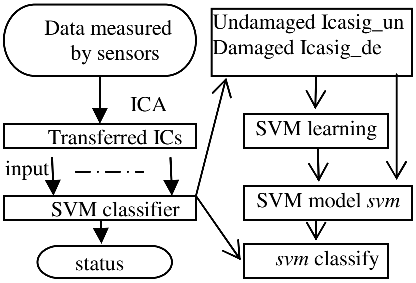


Fig. 2. Frame of integrating ICA and SVM

To seek the optimal **w** and **b** in $f(x) = (\mathbf{w}^T x + \mathbf{b})$ with maximal margin, SVM let the points closest to the separating hyperplane, $|\mathbf{w}^T x_i + \mathbf{b}| = 1$, called the support vectors, and for other points, $|\mathbf{w}^T x_i + \mathbf{b}| > 1$. Given f(x), the classification is obtained as +1 if f(x)>0, otherwise -1.

2.3 Frame Integrated ICA and SVM

The frame of integrating ICA and SVM is shown in Fig.2. The original time domain data measured by the sensors are first used as the input to ICA, and result in the independent component matrix. The matrix serves as the input attributes for SVM model.

3 Experiments

In this section, we will use both undamaged and damaged data as training data to construct a SVM model, and then apply it to test unseen data, exploring that if they are correctly recognized. In addition, the parameters of SVM were previously reported as an important influence over the classification accuracy. Consequently, we designed an experiment to see how they affect the performance in our case. Furthermore, we applied a trained SVM model on different types and levels of damaged data sets and analyzed if SVM model can distinguish them.

3.1 Data Sets

The data set, from the University of British Columbia, is a popular benchmark to testify the classification accuracies. They were developed by The IASC-ASCE SHM Task Group. The structure (Black and Ventura, 1998) is a 4-story, 2-bay by 2-bay steel-frame scale-model structure in the Earthquake Engineering Research Laboratory at the University of British Columbia. It has a 2.5 m × 2.5m plan and is 3.6m tall [7]. The detail Phrase II data can be reached by [6]. In our experiments, we mainly use 7 data sets in the ambient data from this benchmark, in which C01 data is undamaged data, C02-C07 data are different damaged data. For undamaged data, the structural status is '1' (undamaged), and for damaged data, the structural status is '-1' (damaged). The configuration key is attached in the last page, and the description is as following in [8].

Firstly, we input damaged and undamaged sensor data directly into SVM, though this is reasonable design, we could not obtain any suitable outputs since the computing could not converge at all. Hence, in the following experiments, we report our results by integrating ICA and SVM together for damage detection. C01 and one group data from C02 to C07, worked as input to ICA, whose size is 60000x15 (60000 examples with 15 features), then 10 independent components *Icasig_un* were computed and shown in Fig.3. (X axis presents the number of data and the unit is 10^4, Y axis presents the frequency).

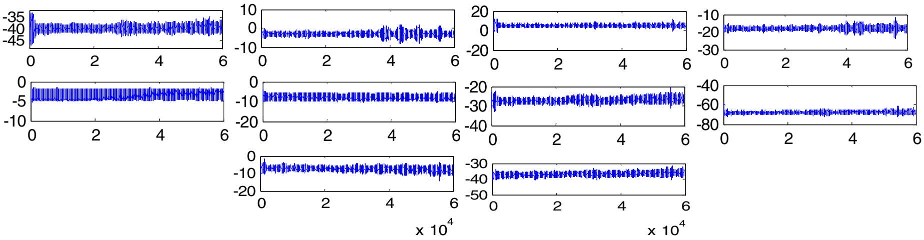


Fig. 3. Independent components of C01

3.2 Experimental Steps

(1)Experiment 1

For choosing the kernel function in SVM, the following experiments were done.
Experiment: Input data: undamaged data (*C01*), damaged data (*C02*)
Output data: structural status (undamaged 1 or damaged -1)

Step 1: Let undamage data ← *C01*; Let damaged data ← *C02* ;

Step 2: By using tanh as the non-gaussian function in FastICA algorithm, we compute the independent components from undamage data, the resulted ICs are denoted as *Icasig_un*;

Step 3: Select a number *t* as the type of different SVM kernel functions, initialize t←0.

Step 4: Randomly select *N* examples from *Icasig_un*, and 50% of them work as train set *Traind_de*, and the other 50% work as test set *Testd_de*.

Step 5: With the same settings on FastICA in all iterations at experiment 1, we compute the independent components from damaged data *C02*, the results are denoted as *Icasig_de*.

Step 6: Randomly select *N* examples from *Icasig_de*, 50% of them work as train set *Traind_de*, and the other 50% work as test set *Testd_de*.

Step 7: Traind ← combine *Traind_un* with *Traind_de*, build SVM model *svm*.

Step 8: Testd ← *Testd_de*. Use *svm* and *Testd* to predict the value for test data, if such value is beyond a scope s, the example will be classified as outlies, otherwise as undamage data.

Step 9: t ← t+1, repeat Step7 until t=4.

The results are shown in Table 1, in which, 'trainCpusec' means the cost CPU second in training data; 'w' is Norm of weight vector; 'VCdim' is estimated VC dimension of classifier; 'testCpusec' means the cost CPU second in classifying data. After consideration, t is assigned 0 in our experiments for training SVM model.

In addition, for trade-off between training error and margin, the corresponding parameter was changed from 0.5 to 2.5, but they did not affect the SVM model. We choose 1 as the trade-off value. The results showed that the liner kernel function and learning trade-off parameter 1 are optimal and will be used in the following experiments.

Table 1. Experiment for choosing Kernel function

t	0	1	2	3
meaning	linear function	polynomial	Radial basis	Sigmoid function
trainCpusec	0.08	0.02	0.05	0.02
$\|w\|$	0.435	0.078	2.777	1.000
VCdim	2.235	3.575	16.421	1.238
testCpusec	0	0	0.01	0
accuracy	0.999	0.618	0.992	1.000

(2) Experiment 2

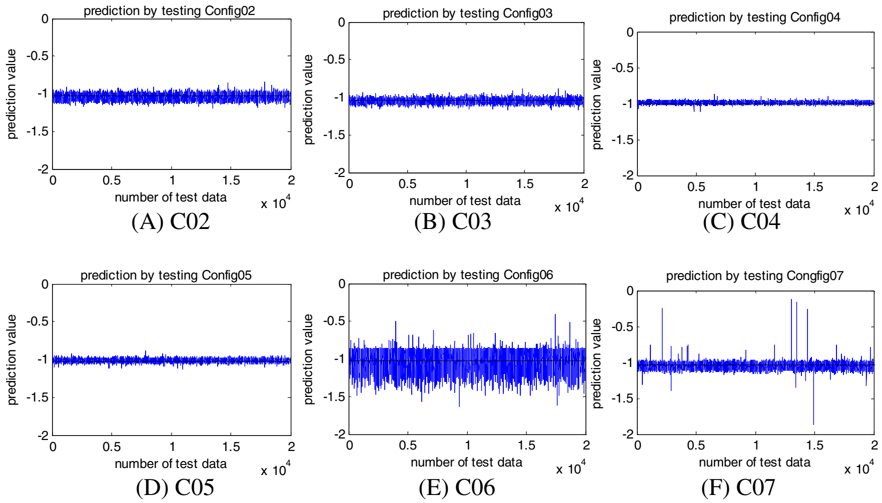
In the experiment, ICA and SVM are combined to build a classification model by using undamaged data and damaged data. Then only damaged data is tested to see if they are classified correctly. The output data are structural status (undamaged 1 or damaged -1)

The classification accuracy is measured as Eq.1. Table 2 shows the experimental results, in which, each experiment use 40000 undamaged data C01 and 20000 damaged data from C02-C07 as train data, corresponding 20000 damaged data from C02-C07 as test data.

$$\text{accuracy} = \frac{\text{number of correct classification samples}}{\text{number of examples}} \qquad \text{(Eq.1)}$$

Table 2. Prediction value by ICA and SVM with some C02-C07 as test data

	C02	C03	C04	C05	C06	C07
accuracy	0.986	0.979	0.983	0.992	0.962	0.996



(A) C02 (B) C03 (C) C04

(D) C05 (E) C06 (F) C07

Fig. 4. Prediction by different data set

For looking for the difference damaged types, the prediction results are analyzed in details.

We observed that different types of damaged data result in the prediction value in different range. The prediction of C07 is nearest to -2 in negative direction and 0 positive direction, and has the biggest wave area. Among the damaged data from C02 to C05, and their wave areas are much smaller than C07. Therefore, C07 data might have biggest damaged level. It is proved by domain experts that all braced removed on all faces in C07 and should have the biggest damage level. In addition, from Fig.4A to Fig.4D, the wave area is reduced, especially in Fig.4C and Fig.4D, most of prediction values are bigger than -1, maybe C04 and C05 have similar damage level. According to the configuration key, Config04 removed braces on 1st and 4th floors in one bay on SE corner, and Config05 removed braces on 1st floor in one bay on SE corner. Therefore, the prediction value can show some information about damage level and damage type, and it help us to identify the different structural damage.

(3) Experiment 3

In the experiment, we aim to analyze the accuracy on classifying two classes of samples: undamaged and damaged data. Hence, the training and testing data are both sampled from two class data.

Table 3. Definitions of tp, fp,tn and fn

	True value=1	False value=-1
Prediction=1	tp	fp
Prediction=-1	tn	fn

Table 4. Accuracy using C01-C07 train and test

traind	C02	C03	C04	C05	C06	C07
1000	0.990	0.987	0.989	0.990	0.986	0.990
2000	0.993	0.990	0.990	0.992	0.989	0.992
4000	0.995	0.991	0.994	0.993	0.991	0.992
8000	0.996	0.994	0.995	0.992	0.992	0.993
10000	0.998	0.995	0.997	0.995	0.991	0.993
20000	0.998	0.995	0.997	0.996	0.990	0.994
40000	0.998	0.996	0.998	0.996	0.992	0.995

Since there are two classes of samples here, we measure the classification accuracy by Eq.2, in which, tp, fp,tn, fn are defined in Table 3. Table 4 shows the experimental results.

$$\text{accuracy} = \frac{\text{tp} + \text{fn}}{\text{tp} + \text{fp} + \text{tn} + \text{fn}} \qquad \text{(Eq.2)}$$

The above results are similar as the results in experiment 2. This showed that, there are obvious difference between the undamaged data and damaged data. Consequently, in experiment 1, the damaged data is classified as outliers; in experiment 2, such data is classified with correct class label. Hence, our results proved that by integrating ICA and SVM, we can extract the distinctive features for undamaged data and damaged data, and further effectively classify unseen data.

(4) Compare ICA-SVM and ICA-ANN

With the same experimental settings in experimental 3, we compared the performance achieved by integrating ICA and SVM with that achieved by integrating ICA and ANN [8]. Results in table 5 showed that ICA-SVM obtains better classification accuracy than ICA-ANN.

Table 5. Accuracy in ICA-ANN and ICA-SVM

data set	C01-C02	C01-C03	C01-C04	C01-C05	C01-C06	C01-C07
ICA+ANN	0.983	0.966	0.979	0.984	0.973	0.953
ICA+SVM	0.998	0.996	0.998	0.996	0.992	0.995

4 Conclusion

In this paper, we proposed an approach of integrating ICA and SVM for structure damage detection. In the first step, independent components are extracted from structure sensor data, which included signals about damage level and type. Next, the obtained components were input into SVM for structural damage classification. We evaluated our approach on the benchmark data from the University of British Columbia. The results from 3 experiments, all used both damaged data and undamaged data for training, showed that the accuracy of damage detection by the proposed method achieved significantly better accuracy than that obtained by use of ICA and ANN. Furthermore, the detailed analysis showed that we could identify different types and levels of damage from the prediction output, which are very useful conclusions for application in SHM.

In next step, we will continue to analyze the independent components for detecting the damage location and consequently support the repair decisions.

References

1. Kiremidjian, A.S., Straser, E.G., Law, K.H., Sohn, H., Meng, T., Redlefsen, L., Cruz, R.: Structural Damage Monitoring for Civil Structures. International Workshop on Structural Health Monitoring. Stanford University, Stanford, CA, USA, September (1997) 18-20
2. Zang, C.: MI Friswell, M Imregun, Structural Damage Detection using Independent Component Analys, Structural Health Monitoring, An Int. J. 3(1), March (2004) 69-84.
3. DeCoste, D., Schölkopf, B.: Training invariant support vector machines. Machine Learning, (2001)
4. Doebling, S.W., Farrar, C.R., Prime, M.B., and Shevitz, D.W.: Damage Identification and Health Monitoring of Structural and Mechanical Systems from Changes in Their Vibration Characteristics: a Literature Review. Los Alamos National Laboratory. The Shock and Vibration Digest, Vol. 30 (2), (1998) 91-105
5. Fritzen, C.-P., Jennewein, D., Kierer, Th.: Damage Detection Based on Model Updating Methd., Mechanical systems and Signal Processin. Vol. 12 (1), January (1998) 163-186
6. http://www.bc.cityu.edu.hk/asce.shm
7. http://wusceel.cive.wustl.edu/asce.shm/benchmarks.htm
8. Song, H., Zhong, L., Moon, F.: Structural Damage Detection by Integrating Independent Component Analysis and Artificial Neural Networks, accepted. Int. Conf. MLMTA 2005
9. Müller, K., Mika, S., Rätsch, G., Tsuda, K., Schölkopf, B.: An Introduction to Kernel-Based Learning Algorithm. IEEE Transactions on Neural Networks, Vol.12 (2) March (2001)
10. Müller, K.-R., Smola, A. J., Rätsch, G., Schölkopf, B., Kohlmorgen, J., Vapnik, V. N.: Predicting time series with support vector machines. Artificial Neural Networks—ICANN'97. ser. Springer Lecture Notes in Computer Science, W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, Eds. Berlin, Germany: Springer-Verlag, vol. 1327 (1997) 999–1004
11. Rytter, A.: Vibration Based Inspection of Civil Engineering Structures, Ph. D. Dissertation. Department of building Technology and Structural Engineering, Aalborg University, Denmark (1993)
12. Masri, S.F., Smyth, A.W., Chassiakos, A.G., Caughey, T.K., Hunter, N.F.: Application of Neural Networks for Detection of Changes in Nonlinear System. J. of Engineering Mechanics, July (2000) 666-676

An LZ78 Based String Kernel

Ming Li and Ronan Sleep

University of East Anglia, Norwich, NR4 7TJ, UK
{mli, mrs}@cmp.uea.ac.uk

Abstract. We have shown [8] that LZ78 parse length can be used effectively for a music classification task. The parse length is used to compute a normalized information distance [6,7] which is then used to drive a simple classifier. In this paper we explore a more subtle use of the LZ78 parsing algorithm. Instead of simply counting the parse length of a string, we use the coding dictionary constructed by LZ78 to derive a valid string kernel for a Support Vector Machine (SVM). The kernel is defined over a feature space indexed by all the phrases identified by our (modified) LZ78 compression algorithm. We report experiments with our kernel approach on two datasets: (i) a collection of MIDI files and (ii) Reuters-21578. We compare our technique with an *n*-gram based kernel. Our results indicate that the LZ78 kernel technique has a performance similar to that obtained with the best *n*-gram performance but with significantly lower computational overhead, and without requiring a search for the optimal value of *n*.

1 Introduction

The support vector machine (SVM) classifier has become a popular tool in the last few years. It achieves good performance in a wide range of pattern recognition tasks. It was originally developed for fixed dimension data vectors, and was later applied to variable length sequence inputs e.g. for protein homology detection [1,2,3], recognition of handwritten characters [4,5] and document classification [14,15]. SVM is driven by a kernel function, which measures the pairwise similarity of two sequences in a high-dimensional feature space, but which can be computed in R^2. Much of SVM theory relies on a choice of a kernel which satisfies Mercer's condition[1]. This requires the kernel to be both symmetric and positive semi-definite (PSD).

In this paper we develop a new kernel suitable for sequence classification. Our kernel builds on the concept of normalized information distance (NID). NID is informally a scalar measure of the information shared between two sequences. We introduce our distinctive approach to using Kolmogorov Complexity (KC) as an inspiration for a similarity measure, and develop a novel string kernel, called the LZ78 kernel. Our kernel is based on mapping sequence input to a high-dimensional feature space which is indexed by all the phrases identified by an LZ78 parse of a sequence.

The performance of our LZ78 kernel is tested empirically on two distinct application areas, namely music style classification and text categorization. The results are compared with standard techniques which use *n*-gram information.

[1] Though some promising results have been obtained with non-compliant kernels [4,5].

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 678 – 689, 2005.

2 The Normalized Information Distance

A Normalized Information Distance (NID) as proposed in [6] is a metric measuring similarity relations between sequences. Informally, it is the ratio of the information shared by the two sequences to the total information content of the pair of sequences. This is illustrated in Figure 1 where circle A represents the Kolmogorov complexity *K(x)* of object *x*, circle B represents *K(y)* and the total area of two circles (*A*+*B*+*C*) is *K(xy),* i.e. the Kolmogorov complexity of the concatenation of the sequences *x* and *y*. Two identical sequences will have NID=0, whilst two sequences with no common information content will have NID=1.

2.1 Conditional Kolmogorov Complexity

Given an object encoded as a binary string *x*, its Kolmogorov complexity *K(x)* is the minimum number of bits into which the string can be compressed without losing information [9]. Intuitively, Kolmogorov complexity indicates the descriptive complexity contained in an object. It is defined as the length of the shortest program for some universal machine which, when run without any input, outputs that string. This is an idealized notion, because it is not computable. However, any compression algorithm gives an upper bound and this can be taken as an estimate of the Kolmogorov complexity.

A random string has relatively high complexity since no structural pattern can be recognized to help reduce the size of program. Strings like structured texts and musical melodies should have lower complexity due to repeated terms and musical structure.

For our application, objects will be sequences of descriptive symbols such as musical notes or words in a text. In our early work [8], we develop a measure of the distance between two such sequences. We base this on the conditional Kolmogorov complexity *K(x|y)*, which is defined as the shortest program that can generate the sequence *x* given *y* as input. *K(x)* is the special case *K(x|λ)* where *λ* is the empty sequence. Following [6], We estimate *K(x|y)* as the difference of the unconditional complexity estimates *K(xy)* and *K(y)*:

$$K(x \mid y) = K(xy) - K(y) \tag{1}$$

When using sequence scanning algorithms to estimate KC, the order of concatenation affects the size of the compressed concatenation, so that the relation *K(xy)* = *K(yx)* may not hold for our estimates. This issue can be handled by using the average of the two orderings.

2.2 Normalized Information Distance

The information distance [11] between two sequences *x* and *y* can be defined as the length of a shortest binary program that computes *x* given *y*, and also computes *y* given *x*. However, such a distance does not take the length of the sequence into account. This motivates the desire for a relative (or normalized) measure that takes account of sequence length. If two pairs of sequences have the same information distance but with different lengths, the longer pair should be given a smaller distance measure than the shorter pair, reflecting the fact that more information is shared between longer sequences.

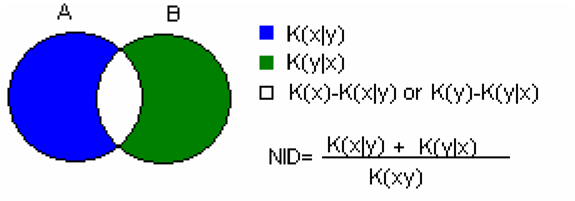


Fig. 1. Illustration of normalized information distance. Circle *A* is *K(x),* circle *B* is *K(y)* and the total area of two circles is *K(xy)*

The authors in [6] use conditional Kolmogorov complexity as the basis of a normalized information distance *D(x,y)* for measuring the similarity relations between sequences. For a well behaved estimate of *K(x,y)*, their distance measure should have values in [0,1] for all sequence pairs and should satisfy the following relations: 1) $D(x,y) = 0$ iff $x = y$; 2) $D(x,y) = D(y,x)$ (symmetry); 3) $D(x,y) \leq \mathrm{D}(x,z)+D(z,y)$ (triangle inequality).

Two versions of the similarity metric are proposed in [6]:

$$d1(x, y) = \frac{K(x|y) + K(y|x)}{K(xy)} \tag{2}$$

$$d2(x, y) = \frac{max(K(x|y), K(y|x))}{max(K(x), K(y))} \tag{3}$$

The second definition (equation 3) can be shown to satisfy the conditions enumerated above without qualification, and in that sense is more satisfactory than the first. But conditional Kolmogorov complexity is not computable, so we have to rely on estimates. This makes it less clear that strict application of the mathematical elegance criterion is the most important guideline for practical classification work. Indeed, in our earlier work in [8], we found the practical performance of the *d1* measure is slightly better than that of equation 3.

3 Support Vector Machines

SVM is a powerful supervised learning algorithm, which maps data into a high dimensional feature space where data is more likely to be linearly separable. A linear decision boundary is obtained by maximizing the geometric margin in order to make it as far away from the data as possible whilst separating the two classes correctly. At the heart of SVM is a constraint quadratic optimization solver which works with the dual problem using only the inner product of input data pairs - this is the similarity measure. This inner product may be specified in terms of a kernel function which does the mapping to the high dimensions, but by working with the dual problem only inner products are needed and these may be evaluated in R^2 or even specified directly as a similarity measure between pairs. When the data is not linearly separable in the high dimensions, a modified SVM procedure can be used, which finds a trade-off between maximizing geometric margin and minimizing the cost of misclassification. This is achieved by introducing “slack” variables, which allow the margin constraint to be violated.

The kernel matrix with entries of similarity score measured by pairwise inner product is known as the Gram Matrix. The decision boundary learnt by SVM is entirely based on the information provided by this matrix. SVM theory requires that the matrix is positive semi definite (PSD), but lots of sophisticated similarity functions not consistent with this requirement are applied in real-world tasks and generate some promising results e.g. [4,5]. Issues like non-convexity of the optimization problem can be handled by a SVM implementation called libsvm [10], which makes simple modification for SMO-type decomposition methods that guarantee the convergence to stationary points for non-PSD kernels. However, the theoretical justification for those approaches is unclear, and the resulting SVM performance may not be ideal. One attempt to get out of above dilemma is the work in [2]. The authors convert a protein sequence into its vector form where the i^{th} element corresponds to the pairwise similarity score between current protein and the i^{th} protein in the data. For each entry in kernel matrix, it corresponds to the inner product of two feature vectors. Note that with this technique the dimension of feature space is set by the number of examples.

4 The LZ78 Kernel

We will now introduce a new class of string kernel, called LZ78 kernel, which is motivated by the idea of NID but deviates from it radically:

- *NID.* Given two sequences x and y, NID uses estimates of the Kolmogorov complexity of x, y and xy to compute a scalar similarity measure;
- *LZ78 Kernel.* Given two sequence x and y, we first use the LZ78 algorithm to construct dictionaries D(x) and D(y). We now interpret a dictionary as indexing an infinite feature space of all possible words over the sequence alphabet. As both D(x) and D(y) are finite for finite sequences, we then use a joint membership test to compute the (infinite) dot product of the two feature vectors. The basic technique may be embellished by pruning longer words from the dictionary, and incorporating relative frequency measures.

4.1 Direct Use of LZ78 to Measure Similarity

Before introducing the LZ78 kernel, we recall the more direct use of LZ78 in our early work [8] to compute a similarity measure.

LZ78 is a "pattern-matching" algorithm [12]. It works by identifying patterns, called phrases, of the data and stores them in a dictionary (i.e. encoding table) that defines shorter "entries" that can be used in their stead. In our approach, all the phrases emitted by LZ78 are considered as the features exhibited by a sequence. LZ78 is a lossless method and thus the original sequence can be re-built from the features (i.e. phrases) identified by it.

```
/* count LZ78 phrases */
Clear dictionary;
w = λ[2];
Kest = 0;
while (more input)
  C = next symbol;
  if (wC in dictionary)
    w = wC;
  else
    add wC to dictionary
    (could emit code for w here)
    Kest = Kest+1;
    w = λ;
  endif
endwhile
if (w!= λ)
  Kest = Kest+1;
endif
return Kest;
```

This captures the essence of an LZ78 encoder omitting irrelevant details. It begins with an empty dictionary, and then performs a left to right scan of the symbol sequence extending the length of the uuencoded word *w* until it finds that *w* is not in the dictionary. At this point it stores *w* in the dictionary, resets *w* to 0. The procedure repeat until the whole sequence gets parsed. One example is shown on the right column of Table 1. Notes that 1) unlike *n*-gram approach, our LZ78 kernel technique could produce a set of features with varying length instead of fixed; 2) LZ78 identifies increasingly long initial portions of repeated phrases gradually: that is, it will need to see a phrase of length L on L occasions before it remembers the whole phrase in its dictionary.

Other coding schemes could be considered as a basis for a string kernel. As an illustration, Table 1 shows the features generated by LZ78 and LZ77 for the string 'abcabcabc'.

Table 1. Features (i.e. phrases) identified by LZ77 and LZ78 for 'abcabcabc'

LZ77	LZ78
a, b, c, abc	a, b, c, ab, ca, abc

Normally, the LZ78 algorithm emits code consisting of a compact reference to the each new entry. For direct use of LZ78 to compute a similarity measure, we are only interested in the Kest, the length of the compressed output. In our implementation, we omit the coding step and simply increment variable Kest at each new dictionary entry.

Note that our LZ78 derived Kest is not necessarily an upper bound on the Kolmogorov complexity, because we ignore the number of bits required to represent the codewords for each phrase. However, the length of an LZ78 parse does give a simple

[2] 'λ' represent the empty string.

clear measure of the information size of an object without any of the discontinuities introduced in a practical compressor by details such as code word / window sizes. Further, LZ78 is simple and extremely fast, and its theoretical properties have been extensively studied. This increases the chances that its properties with regard to source separation and classification might be amenable to theoretical analysis.

4.2 LZ78 Kernel

In 4.1 we used LZ78 to compute an estimate of the Kolmogorov complexity of a sequence. In our new kernel approach, we take as output from LZ78 the dictionary constructed during the parse, and regard the phrases in the dictionary as signaling the presence of one of the (infinite) number of possible phrases. Hence the features of a sequence are the set of all the phrases identified by LZ78. Thus, the feature space is indexed by all subsequences (contiguous) of varying length from alphabet ∑. The representation of a sequence $\acute{s}$ under this space is a binary vector Φ_{lz78} with the *i*-th element valued 1 if that feature occurs in $\acute{s}$, 0 if not. The LZ78 kernel similarity measure between two sequences is then defined as the inner product of their feature vectors:

$$K_{lz78}(s,t) =< \Phi_{lz78}(s), \Phi_{lz78}(t) > \tag{4}$$

As mentioned earlier, it is natural to normalize the similarity score in order to take account of sequence length. In the kernel method, this effect can be achieved by normalizing the feature vectors in the feature space:

$$K_{lz78}^{norm}(s,t) = \frac{K_{lz78}(s,t)}{\sqrt{K_{lz78}(s,s)K_{lz78}(t,t)}} \tag{5}$$

Note that while the feature space is potentially huge, it is evident from the description in section 4.1 that the number of non-zero entries in a feature vector is bounded by *length(x)*, which enables various efficient implementations for computing kernels without storing all the features explicitly. In practice, storing the dictionary as a suffix tree allows rapid searching of the existing phrases. The complexity of computing each kernel depends on the underlying patterns. In the worst case, LZ78 identifies no repeated patterns and produces a set of uni-gram phrases and cost $O(n)$ time. Thus, we can estimate that the overall cost of computing LZ78 kernel is bounded by $O(n)$ since a given suffix tree can be used to search for a substring *pattern[1…m]* in $O(m)$ time.

5 Experiments

Two datasets were used for evaluating the performance of our LZ78 kernel. The first one is our own collection of MIDI files downloaded from internet. The second is a benchmark dataset for text categorization, Reuters-21578. All the experiments were conducted using libsvm package [10], which guarantees the convergence to stationary points for non-PSD kernels.

5.1 MIDI Files Dataset

749 MIDI files falling into 4 categories were collected from the internet. Two of the categories were western classical music composed by Beethoven (289 files) and Haydn (255 files). The remaining categories were Chinese music (80 files) and Jazz (125 files). Using the principal track from the MIDI file, a sequence of the pitch interval of the melody is obtained. If two note-on events happened simultaneously, only the one with highest pitch value was preserved. Only pitch information is used, so there is no use of onset time or duration of each note. However, the order of the notes in the sequence is preserved. The aim of this pre-processing was to extract the essence of the melody from the MIDI file, removing unintended clues and MIDI formatting clutter. It also makes the experimental results described here directly comparable with other classification experiments carried out by the authors [13].

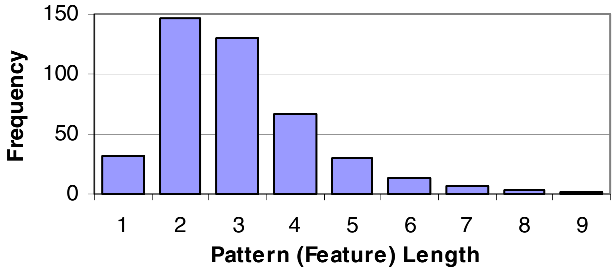


Fig. 2. Averaged phrase length distribution within a single musical melody (generated by LZ78 parsing)

In this task, we take an empirical approach to examine the performance[3] of our LZ78 kernel. More specifically, we focus on three issues: 1) do all compressors from LZ family result in similar classification performance? 2) how does our LZ78 kernel compare with a non-PSD NID kernel[4,5]? 3) is our approach competitive with alternatives (e.g. *k*-spectrum kernel)?

5.1.1 K-Nearest Neighbor (*k*-NN) with Normalized Information Distance

In this experiment, the effect of variant compressors from LZ family on NID approximation is evaluated indirectly by using a simple classifier, i.e. *k*-NN, to check which results in better predictive accuracy compared with others. Two representatives are selected, i.e. LZ77 and LZ78. We do not consider LZW separately here: LZW is a

[3] The performance is evaluated by a standard three-fold stratified cross validation (CV).

[4] The NID approximated by compression algorithm is not symmetric due to the problem mentioned in section 2.1. i.e. the order of the concatenation of two sequences affects the size of compressed concatenation.

[5] To become a valid kernel, the similarity function needs to satisfy Mercer condition (symmetric and PSD) in order for a mapping $\phi(.)$ to exist. In other words, the dot product can be evaluated directly by using a nonlinear function in input space i.e. the kernel trick.

variation of LZ78 and produces results very similar to those for LZ78. The results shown in Table 2 indicate that, LZ78 outperforms LZ77 using *k*-NN as the classifier. It confirms our previous experience which is that LZ78 generally outperform LZ77. Somehow LZ77's focus on maximal length repeated substrings lose out against LZ78's much more myopic approach. Note that, with a simple classifier like *k*-NN and similarity measure NID, this experiment also set a performance baseline for the later experiment.

5.1.2 Comparison of NID Kernel with LZ78 Kernel

Both LZ78 and NID kernel derive from the normalized information distance [6]. The NID kernel is constructed by simply replacing the squared Euclidean distance in an RBF kernel with NID. As mentioned previously, with such a non-PSD kernel, the SMO algorithm is expected to converge but no global optimality can be guaranteed. Here, we confirm this assumption experimentally. As it is shown in Table 2, NID kernel performs better than *k*-NN but it is surpassed by LZ78 kernel.

5.1.3 Comparison with *k*-Spectrum Kernel

The *n*-gram method is a simple and effective approach which has been successfully applied in series of classification tasks for symbolic sequence. One of its variants called *k*-spectrum kernel is implemented here for comparison purpose. The *k*-spectrum of a sequence input is the set of all the *k*-length subsequences (contiguous) contained in it. Given two sequences, the *k*-spectrum kernel is defined as the inner product of their *k*-spectrum feature vectors [1]. The results in Table 2 demonstrate the advantage of our method over *k*-spectrum on this melody dataset. We attribute this to the one of the principal differences between *n*-gram approach and ours, i.e. the restriction over the pattern (i.e. feature) length: *n*-gram approach requires the feature length to be fixed but our method allows it to be variable (see Figure 2). Although a linear combination of spectrum kernels with variant k may improve the performance, it raises the computational cost simultaneously and also leads to the problem known as finite sample size effect or peaking of classification (see Figure 3).

Table 2. Accuracy with different classifiers and similarity metrics

Classifier	Similarity Function	Compressor	Accuracy(STD)
k-NN (*k*=1)	NID	LZ77	64.99(3.26)
		LZ78	69.56(4.65)
SVM	NID	LZ78	71.24(3.76)
	LZ78 Kernel	LZ78	74.35(5.81)
	Combination of 1,2,3,4,5 Spectrum Kernel	-	72.40(7.30)
	Single 2-Spectrum Kernel	-	71.72(2.17)
	Single 3-Spectrum Kernel	-	70.45(6.01)
	Single 4-Spectrum Kernel	-	69.01(4.63)
	Single 5-Spectrum Kernel	-	65.90(3.72)

5.1.4 Issues About Dimensionality and Finite Sample Size Effect

With finite learning samples, the performance of a classifier will improve with the increase of the feature size until an optimal point is reached and then it will deteriorate when adding more features [17,18]. This phenomenon is known as the finite sample size effect or peaking of classification and suggests using the feature sets close to the optimal size. For the *n*-gram-based approach, this corresponds to choice the right pattern length *n*. Although Hua [19] claims that SVM shows strong robustness with respect to large feature sets, their work is based on a small-scaled experiment (dimensionality≤30). For the method mentioned in last section that uses all the co-occurring substrings, the dimension is very much larger.

The finite sample size effect can be seen clearly in Figure 3, with the *n*-gram performance dropping for *n*=4 and above, and continuing to fall. In contrast, the performance of LZ78 kernel is relatively immune from this effect. Informally, our LZ78 kernel method is centred on phrases which actually occur in the data rather than the set of all possible phrases of a given length *n*. Further, this is done with the high degree of efficiency associated with LZ78.

5.2 Reuters Dataset

To explore the LZ78 kernel in a different application setting, we conducted experiments using the Reuters-21578 benchmark for text categorization. Following [14], [15], we use the "ModApte split" of the database and limit attention to the ten most frequent categories only. Documents were preprocessed by removing stop words and punctuation. Further, we performed stemming with a tool called snowball, which is concerned mostly with finding a unique root of a word and does not necessarily produce an existing word or lemma.

For performance evaluation we use the *F1* measure, which gives equal weighting to precision *p* and recall r, i.e. *F1=2pr/(p+r).* The overall performance is evaluated by macro-averaged *F1*, which is influenced equally by all categories instead of dominated by the performance of "large" categories.

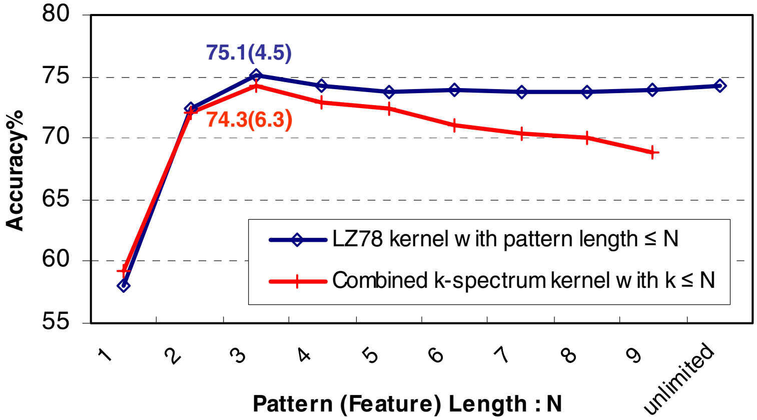


Fig. 3. Peaking of classification or finite sample size effect for melody classifier. The performance with feature size close to the optimal point *D*(3) are shown in the figure

Table 3. *F1* scores for top ten Reuters-21578 categories. All the methods compared are linear kernels measuring the similarity between documents indexed by words, character-based 3-gram and LZ78 phrases respectively. All the features are equally weighted by 1

Categories	Kernel		
	Word	3-Spectrum	LZ78
earn	**98.48**	97.71	**98.48**
acq	95.85	95.33	**96.27**
money-fx	81.40	**83.71**	81.50
grain	**91.93**	90.14	90.71
crude	87.66	87.77	**87.94**
trade	85.71	81.08	**85.97**
interest	73.28	74.24	**77.73**
ship	**81.53**	76.82	77.85
wheat	87.67	**87.94**	86.11
corn	**77.23**	74.75	72.16
macro-average	86.08	84.95	85.47

As shown in Table 3, our LZ78 kernel performs best in five out of ten categories (i.e. earn, acq, crude, trade, interest), the word kernel performs best in four (i.e. earn, grain, ship and corn) and the spectrum kernel in two (i.e. money-fx and wheat). The most obvious conclusion is that the LZ78 kernel, used in a rather crude fashion, is producing a competitive performance in this domain. Overall it does not do quite as well as the Word kernel using the macro-averaged F1 measure, but it comes very close. It is worth noting that substituting words with LZ78 phrases as features improves the precision p at the cost of losing recall r.

6 Conclusion

Previous works have explored various compression-based methods for the tasks of text categorization [16] and sequence clustering [7]. The approaches approximate the cross-entropy or Kolmogorov complexity simply by the compressed length output of a character-based industrial compressor such as gzip or ppm.

In our work, we develop a novel string kernel based on a pure LZ78 parsing algorithm, and use its pattern-matching scheme as the heuristic for extracting informative features efficiently from a sequence of descriptive symbols.

We evaluated our LZ78 kernel on two distinct classification problems: one for musical style classification and another for text categorization. Although, LZ78 kernel incorporates no prior knowledge specific to the problem, it does capture some discriminative features and generate comparable results with some standard algorithms. In musical dataset, melodies always consist of note patterns repeated in different pitch level, which makes our LZ78 method a good candidate for identifying those patterns and outperforming all the other alternatives. In Reuters dataset, our method performs best in 5 out of 10 categories (i.e. earn, acq, crude, trade and interest) though a macro averaging measure puts it a close second behind the word kernel technique.

Our work indicates that adaptive dictionaries of the sort found in an LZ78 compressor provide good features for driving efficient classifiers which perform well over a range of classification tasks. Perhaps more fundamentally, we note that our LZ78 kernel does not suffer from the tail off in performance associated with *n*-gram techniques.

Acknowledgements

We wish to thank the anonymous reviewers for their helpful comments and Stephen Cox for useful discussion.

References

1. Leslie, Eskin et al.: The Spectrum Kernel – A String Kernel for SVM Protein Classification. Proceedings of the Pacific Symposium on Biocomputing, (2002) 564–575
2. Liao, L., Noble, W.S.: Combining pairwise sequence similarity and support vector machines for remote protein homology detection, Proc. 6th Int. Conf. Computational Molecular Biology, (2002) 225–232
3. Jaakkola, T., Diekhans, M., Hausler, D.: A Discriminative Framework for Detecting Remote Protein Homologies. Journal of Computational Biology. 7(2000) 95–114
4. Bernard Haasdonk.: Tangent Distance Kernels for Support Vector Machines. Proc. of the 16th Int. Conf. on Pattern Recognition, 2(2002) 864–868
5. Bahlmann, Haasdonk.: On-line Handwriting Recognition with Support Vector Machines–A Kernel Approach. Proc. of the 8th Int. Workshop on Frontiers in Handwriting Recognition. (2002) 49–54
6. Ming, Li., Xin, Chen., Bin, Ma., Vitanyi, P.: The Similarity Metric. Proc. 14th ACM-SIAM Symposium on Discrete Algorithms. (2003) 863–872
7. R. Cilibrasi, P. Vitanyi.: Clustering by Compression, IEEE Transaction Information Theory. Submitted. See http://arxiv.org/abs/cs.CV/0312044
8. Ming, Li., Sleep, M.R.: Melody Classification Using A Similarity Metric Based on Kolmogorov Complexity. Proceeding of Conference on Sound and Music Computing. Paris. (2004)
9. Ming, Li., Vitanyi, P.: An Introduction to Kolmogorov Complexity and Its Applications. Springer–Verlag, Berlin Heidelberg New York (1997)
10. Lin, H.-T., Lin, C.-J.: A Study on Sigmoid Kernels for SVM and the Training of Non-PSD Kernels by SMO-TYPE Methods. Technical Report. Department of Computer Science and Information Engineering, National Taiwan University
11. Bennett, C.H., Gacs P., Ming, Li., Vitanyi, P.M.B., Zurek, W.: Information Distance. IEEE Transactions on Information Theory, 44:4 (1998) 1407-1423
12. Jacob, Ziv., Abraham, Lempel.: Compression of Individual Sequences via Variable-Rate Coding. IEEE Transactions on Information Theory, Vol. 1T-24, No.5, Sep. (1978)
13. Ming, Li., Sleep, M.R.: Improving Melody Classification by Discriminant Feature Extraction and Fusion. Proc. International Symposium on Music Information Retrieval (ISMIR) Barcelona, 11-14 Oct. (2004)

14. Nicola etc.: Word-Subsequence Kernels. Journal of Machine Learning Research. 3 (2003) 1059-1082
15. Huma, Lodhi., Craig, Saunders., John, Shawe-Taylor., Nello, Cristianini., Chris, Watkins.: Text Classification Using String Kernels. The Journal of Machine Learning Research. 2 (2002) 419-444
16. Teahan, W.J., Harper, D.J.: Using Compression-Based Language Models for Text Categorization. In Workshop on Language Modeling and Information Retrieval. Carnegie Mellon University. (2001) 83-88
17. Kanal, L.: On Dimensionality and Sample Size in Statistical Pattern Classification. Pattern Recognition 3 (1971) 225-234
18. Jain, A.K. and Chandrasekaran, B.: Dimensionality and Sample Size Considerations in Pattern Recognition Practice. In P.R.Krishnaiah, L.N. Kanal (Eds.). Handbook of Statistics. Vol 2. North-Holland, Amsterdam (1982) 835-855
19. Jianping, Hua., Zixiang, Xiong., James, Lowey., Edward, Suh., Edward, R. Dougherty.: Optimal Number of Features as a Function of Sample Size for Various Classification Rules. Journal of Bioinformatics. Vol 21. 8 (2005) 1509-1515

Classifying Class and Finding Community in UML Metamodel Network[+]

Bin Liu[1], Deyi Li[2], Jin Liu[3], and Fei He[4]

[1] The Compute School, Wuhan University, Wuhan, China, 430072
hurricane_liubin@yahoo.com
[2] Beijing Institute of Electronic System Engineering, Beijing 100039, China
ziqin@public2.bta.net.cn
[3] State Key Laboratory of Software Engineer, Wuhan University, Wuhan, China, 430072
jliu@sklse.sklse.org
[4] Major in Compute Science Graduate School of Science and Engineering,
Waseda University, Tokyo, 169-8555
hefei@ruri.waseda.jp

Abstract. Composed of many classes or modules, big software can be represented with network model. By extracting the topology of UML metamodel from the UML metamodel specification, the scale-free, small-world networks properties are revealed. Based on this observation, we come up with our algorithms that can classify all classes in UML metamodel into three kinds: core, general and leaf. Our algorithm can categorize all classes into several subgroups by three factors, i.e., degree, betweenness and weak link. It is illustrated through case study that this algorithm is effective at mining community structure in large software systems.

1 Introduction

With compute service prevails in our daily life, different kinds of compute systems exist widely in human society, e.g., communication systems, the Internet, etc. What these bring to us is huge amount of information with various forms, such as phone-call, or information packets [1], [2], [3]. Complex systems grow and evolve to reveal intricately networked organizations. Further research on complex network in recent years has gained a deep insight into the topological properties of underlying networks that complex network systems share, i.e. ‘‘scale-free’’ and ‘‘small-world’’ qualities [1]. Object-oriented software systems with huge scale can also be regarded as a kind of important complex network, which is composed of many classes or objects. Until recent years, little attention has been paid to this filed. While analysis and design are both critical phases in software development, UML-compatible tools are kind of most influencing CASE tools that are widely been used in almost each develop phase.

[+] This research project was supported by the National Natural Science Foundation of China under Grant No. 60373086, Wuhan Science & Technique Key Project under Grant No. 20021002043, Open Foundation of SKLSE under Grant No. 03-03, the Provincial Natural Science Foundation of Hubei under Grant No. 2002ABB037, Hubei Province Key Project under Grant No. 902130819.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 690 – 695, 2005.

UML is benefit for accelerating the developing process [4], [5], [6]. The extension ability of standard UML makes it can be modified to describe embedded systems, real-time systems, and so forth. UML metamodel is the embodiment of OO modeling techniques. Composted of many classes, UML metamodel can also be regarded as a complex network. With these new viewpoints, we analyze the structure of UML metamodel again for mining the power of UML technique that may be untouched before.

The rest of this paper is organized as follows. Section 2 specifies how to get topology of UML metamodel. Section 3 explains the concept betweenness and the algorithm to calculate it. Section 4 puts forward the concept weak link. We also advance algorithms that classifies classes in metamodel and clusters them to several communities in this section. Section 5 summarizes our work.

2 UML Metamodel

UML is widely used to modeling all artifacts during almost each phase in software develop. More and more software engineers and software companies benefit from UML. Thus nowadays UML is de facto industry standard [5], [6]. The current version of metamodel is 2.0. In UML metamodel specification, class has three type relations: generalization, aggregate and association. Each class is a node in UML metamodel network. If class A and class B has one of forenamed relations, there is an edge between Class A and Class B. The result is shown in figure 1.

Table 1. Calculate parameters of UML metamodel

Name	Nodes	Edges	Average Diameter	γ
MOF	29	44	6.4	-0.8
UML1.5	175	452	6.8	-0.9
UML2.0	224	622	7.1	-0.9

Table 2. Summary of various symbols used throughout the text

Symbol	Meaning
G	complex network
N	number of nodes in a network
V	serial number of node in a network
K(v)	degrees of vertex V
L	shortest path between node Vi and Vj in network
GL	all shortest paths in network
B(v)	betweenness of node V
C(v)	importance of node v: leaf, general, core
Mean(k)	average degrees of all nodes in network
E(p), E(q)	two points of a edge
D	average diameter of network
a, b, c	adjustable parameters in algorithm

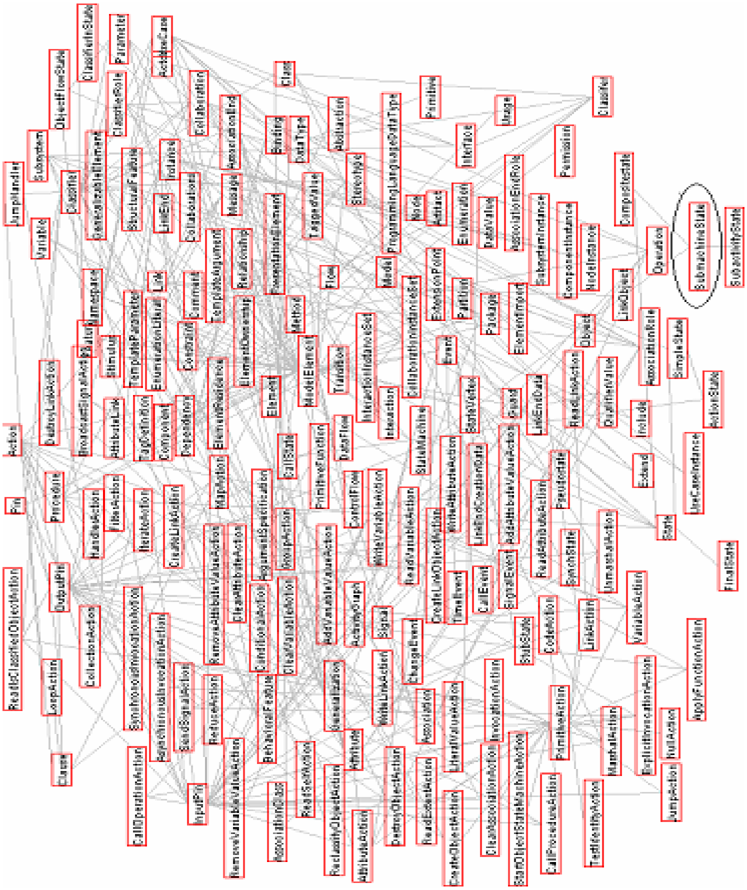


Fig. 1. The network extracts from UML1.4 metamodel

3 Betweenness

Since message is the bridge of classes in software system, different behavior represents different message pass path. In order to enhance the software robustness, more redundancy classes or messages are added in software. The larger the network is, the more difficult it is to assure that task execution is effective.

The most important thing in UML metamodel is to find optimal shortest path between source class and destination class [3]. Different task has different the shortest path. If a class receives more kinds of message or sends more kinds of message, this class is considered to be critical important in the whole network. This seems be in a general way, but the case is not always right. Take transportation centers as an example, the new transportation one is much important than old one. In this case, betweenness is used to describe the importance of nodes [7].

Definition 1: Node Betweenness. If L is the shortest path between node Vi and Vj in complex network, B(Vi)=0,B(Vj)=0. If L pass through node Vm (Vm!=Vi and Vm!=Vj), B(Vm)=1.

Definition 2: Node Betweenness of Network. Node v betweenness is number of the shortest path that passes through node v. Program CalculateBetweenness gives a method how to calculate Network Betweenness of node v.

```
program CalculateBetweenness(output )
{assuming all shortest paths are calculated according
  to Dijkstra algorithmic and stored in GL set};
var  MaxNodes=10000;
     B:array[1..NetNodes] of integer;
     GL:Set;
begin
 for i= 1 to MaxNodes do
   B[v]:=0
 for each L in GL
  for each v in G
  if v in L then
   B[v]:= B[v]+1;
end;
```

4 Weak Link

Is a node is important if it has high betweennesses? This is not always true. The class SubmachineState circled by an oval is an example to prove it. If node SubmachineState is deleted from network, it only effects interactions among the other nodes with SubmachineState, or SubactivityState, but not the interactions among them. A primary reason is that SubmachineState has low degrees. So we adopt both degree and betweenness to classify all classes in UML metamodel. All nodes are classified into three types: leaf, general, core. The algorithm making for the method is listed as follows [2], [3].

```
program ClassifyingNodes(output)
var  K:array[1..NetNodes] of integer;
     C:array[1..NetNodes] of integer;
     MeanDegree:Integer;
begin
  CalculateBetweenness;
  MeanDegree=CalculateMeanDegree;
  for each e in GL
    begin
     if B[v]=0 then
       C[v]=leaf ;
     if(K(v)<a*MeanDegree) then
       C[v]=general
     else
     if (B[v]> K[v])
       C(v):=core ;
```

```
      else
      if(abs(B[v]- K[v])<b)
        C[v]:= core;
     else
        C[v]:= general;
   end ;
 end;
```

Weak link originated from a social investigation [8] creates a bridge between two disconnected communities to exchange information effectively. According to complex network theories [1], the new edge called as rewired edge in [1] decreases the average diameter of the whole network, although this new link is seldom. Also there may exist some weak links in UML metamodel. If they are broken, the average diameter of the whole network will increase acutely. And the network would collapse in the worst condition. The FindWeakLink algorithm gives the process how to choose weak link in network.

```
program FindWeakLink(output)
begin
  for each e in GL
   if (E(p)= core) and (E(q) =core) then
    e:=WeakLink
end;
```

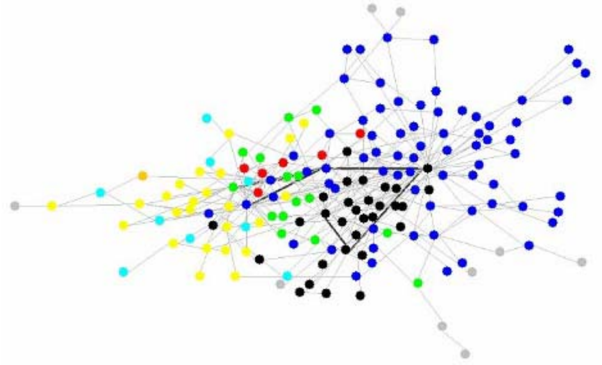


Fig. 2. Classify all classes in UML metamodel. The light gray and black lines are all edges in the shortest paths. The points in black edges are core nodes. The circles filled same color belong to the same community, so there are six communities. The light gray nodes are classes that core nodes cannot reach them in supposed search depth. Perhaps these classes function may be adjusted in the feature

By weak link, we can categorize the network into several communities. First we delete all weak links and the links that is not in the shortest paths from the network. Then from each point of every weak link, we use depth-first search algorithm to get all nodes that the max depth is c* D. Because the average diameter of network is 6, and the core nodes are center in network, so the max search depth is half of D, c =0.5. Figure 2 shows the result with a=3, b=2. The whole network is split into six communities with cluster coefficient: 0.12, 0.16, 0.31, 0.30, 0.16, 0.19, 0.16, which is almost near the average the cluster coefficient 0.2 [1]. It is illustrated that this algorithm is effective at mining community structure in large software systems.

5 Conclusion

In this paper, the UML metamodel is regarded as a network. We put forward an algorithm to classify all classes into three types, and describe a new algorithm to mining community structure in UML metamodel network. From the software engineer point, every community may represent one subsystem that performs some function. The one of most important research in software engineer is to optimize and refactor subsystems, so the software network may be a more effectively way to repartition subsystems and refactor those subsystems. Moreover, UML is an extendable model, with the help of UML metamodel network, we can forecast which communities must be enhanced or adjusted to fit the extension we hope. And these are our future work.

References

1. S. H. Strogatz: Exploring complex networks. Nature 410, (2001) 268–276
2. R. Albert and A.-L. Barab´asi: Statistical mechanics of complex networks. Rev. Mod. Phys. 74, (2002) 47–97
3. Christopher R. Myers: Software systems as complex networks: Structure, function, and evolvability of software collaboration graphs. Rev. Mod. Phys. 68, (2003) 68–83
4. Linda Heaton: Meta Object Facility (MOF) Specification Version 1.4 [EB/OL]. http://www.omg.org/cgi-bin/doc?formal/2002-04-03, 02-04-03/05-03-15
5. Linda Heaton: Unified Modeling Language Specification Version 1.5 [EB/OL]. http://www.omg.org/cgi-bin/doc?formal/03-03-01, 03-03-01/05-03-15
6. Bran Selic: Unified Modeling Language Specification Version 2.0 [EB/OL]. http://www.omg.org/cgi-bin/doc?ptc/2004-10-02, 04-10-02/05-03-15
7. M. E. J. Newman: Detecting Community Structure in Networks. Eur. Phys. J. B 38, (2004) 321-330.
8. Csermely, P: The strength of weak links: from stress proteins to social networks. Hungarian Science 111, (2004) 1318-1324

An Adaptive Network Intrusion Detection Method Based on PCA and Support Vector Machines *

Xin Xu[1,2] and Xuening Wang[2]

[1] School of Computer, National University of Defense Technology,
410073, Changsha, P.R.China
xuxin_mail@263.net
[2] Institute of Automation, National University of Defense Technology,
410073, Changsha, P.R.China

Abstract. Network intrusion detection is an important technique in computer security. However, the performance of existing intrusion detection systems (IDSs) is unsatisfactory since new attacks are constantly developed and the speed of network traffic volumes increases fast. To improve the performance of IDSs both in accuracy and speed, this paper proposes a novel adaptive intrusion detection method based on principal component analysis (PCA) and support vector machines (SVMs). By making use of PCA, the dimension of network data patterns is reduced significantly. The multi-class SVMs are employed to construct classification models based on training data processed by PCA. Due to the generalization ability of SVMs, the proposed method has good classification performance without tedious parameter tuning. Dimension reduction using PCA may improve accuracy further. The method is also superior to SVMs without PCA in fast training and detection speed. Experimental results on KDD-Cup99 intrusion detection data illustrate the effectiveness of the proposed method.

1 Introduction

With the wide spread of computer networks, security problems in computer systems have become more and more important since various intrusions or viruses may cause significant losses to information assets. To defend attacks of information systems, lots of security techniques and products, such as firewalls, intrusion detection systems (IDSs), etc., have been developed. Among these security techniques, intrusion detection plays a key role because it is a dynamic defense technique, which is different from earlier static defense techniques including firewalls and access control.

In intrusion detection systems, misuse detection and anomaly detection are the two main classes of detection policies. Misuse detection can detect known attacks by constructing a set of features or signatures of attacks while anomaly detection detects novel attacks by modeling normal behaviors. Both misuse and anomaly detections rely on analysis of large amounts of audit data or events. It is a time-consuming and tedious

* Supported by the National Natural Science Foundation of China Under Grants 60303012, 90104001, Specialized Research Fund for the Doctoral Program of Higher Education under Grant 20049998027, and Chinese Post-Doctor Science Foundation under Grant 200403500202.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 696 – 703, 2005.

work. Thus, intrusion detection techniques based on data mining or machine learning [1][2] have attracted much attention in recent years, which are usually called adaptive intrusion detection techniques.

In data-mining-based IDSs, the process of data analysis and behavior modeling can be automatically carried out and there are also two different processing policies, i.e., misuse detection and anomaly detection. In misuse detection, various standard data mining algorithms [2,3], fuzzy logic models [4], and neural networks [5] have been used to classify network intrusions. In anomaly detection, data mining methods based on statistics [6], or clustering techniques [7] are employed to identify attacks as deviation from normal usage.

Despite many advances that have been achieved, existing IDSs still have some difficulties in improving their performance to meet the requirements of detecting increasing attacks in high-speed networks. One difficulty is the problem of detection accuracy. Since misuse IDSs employ signatures of known attacks, it is hard for them to detect deformed attacks, notwithstanding completely new attacks. Although anomaly detection can detect new types of attacks by modeling a model of normal behaviors, the false alarm rates in anomaly-based IDSs are usually high. Another difficulty of IDSs is to detect intrusions in real-time with large amounts of data in high-speed networks.

To overcome the above problems, this paper proposes a novel adaptive network intrusion detection method based on principal component analysis (PCA) [8] and support vector machines (SVMs) [9]. In the proposed method, PCA is used to reduce the feature dimension of network connection records and SVMs are employed to construct intrusion detection model based on the processed training data. Compared to previous IDS methods, the adaptive intrusion detection method not only has good accuracy but also has advantages both in fast training and testing speed, which will be illustrated in the experiment on KDD-Cup99 dataset.

2 Intrusion Detection as a Multi-class Pattern Recognition Problem

Although the intrusion detection method studied in this paper can be applied to general-purpose IDSs, to facilitate discussion, we will only consider network connection data, especially the KDD-Cup99 dataset in the following.

The KDD-Cup99 dataset is based on the 1998 DARPA intrusion detection evaluation program, where an environment was setup to simulate a typical US Air Force LAN and raw tcpdump data were collected. For each TCP/IP connection, 41 quantitative and qualitative features were extracted as a data record. The data records are all labeled with one of the five types, which are

- Normal: Normal connections are generated by simulated daily user behavior such as visiting web pages, downloading files, etc.
- DoS: DoS denotes the denial of service attacks. A denial of service attack causes the computing power or memory of a victim machine too busy or too full to response to legitimate access. Examples of DoS attacks are Apache2, Back, Land, Mail bomb, SYN Flood, Ping of death, Process table, Smurf, Syslogd, Teardrop, Udpstorm.ï
- U2R: U2R means user to root, which is a class of attacks that a hacker begins with the access of a normal user account and then become a super-user by

exploiting vulnerabilities of the system. Examples are Eject, Ffbconfig, Fdformat, and Loadmodule.

- R2L: The R2L attack or remote to local attack is a class of attacks that a remote user gains access of a local account by network communication, which include Sendmail, Xlock, and Xsnoop.
- Probe: A Probe attack scans the network to gather information of computers so that vulnerabilities can be found for further attacks.

To construct an intrusion detection model based on the KDD-Cup99 dataset, each data record is denoted by a 41-dimensional vector with class labels from 1 to 5, where qualitative elements are transformed to discrete values and class labels correspond to the above five types of connections. Then, the construction of intrusion detection model becomes a multi-class pattern recognition problem so that data mining methods can be employed.

3 Adaptive IDS Using PCA and SVM

3.1 Framework of the Adaptive IDS

The framework of the adaptive IDS includes a training process and a testing process, as shown in Fig.1.

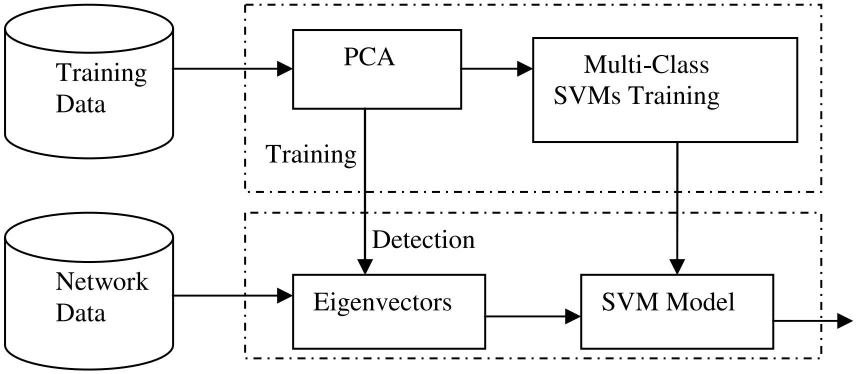


Fig. 1. Framework of the adaptive IDS

The training process consists of two main steps, i.e., the dimension reduction step and the classifier training step. In the dimension reduction step, the PCA algorithm is used to compute the principal components of the data.

3.2 Dimensionality Reduction Using PCA

In both neural network and statistics studies, PCA is one of the most fundamental tools of dimensionality reduction for extracting effective features from high-dimensional vectors of input data. In the following, we will discuss the application of PCA to dimension reduction of network connection data and its combination with SVMs.

As discussed in Section 2, the network data records can be denoted as

$$x_t = [x_{t1}, x_{t2}, ..., x_{tn}]^T \quad (t\text{=}1,2,\ldots, N),\ n\text{=}41 \tag{1}$$

Let

$$\mu = \frac{1}{N}\sum_{t=1}^{N} x_t \tag{2}$$

Then, the covariance matrix of data vectors is

$$C = \frac{1}{N}\sum_{t=1}^{N}(x_t - \mu)(x_t - \mu)^T \tag{3}$$

The principal components are computed by solving the eigenvalue problem of covariance matrix *C*:

$$Cv_i = \lambda_i v_i \tag{4}$$

where λ_i $(i = 1,2,...,n)$ are the eigenvalues and $v_i (i = 1,2,...,n)$ are the corresponding eigenvectors.

To represent network data records with low dimensional vectors, we only need to compute the first *m* eigenvectors which correspond to the *m* largest eigenvalues.

Let

$$\Phi = [v_1, v_2, ..., v_m], \quad \Lambda = \mathrm{diag}[\lambda_1, \lambda_2, ..., \lambda_m] \tag{5}$$

Then we have

$$C\Phi = \Phi\Lambda \tag{6}$$

In PCA, a parameter ν can be introduced to denote the approximation precision of the *m* largest eigenvectors so that the following relation holds.

$$\sum_{i=1}^{m} \lambda_i / \sum_{i=1}^{n} \lambda_i \geq \nu \tag{7}$$

Given a precision parameter ν, we can select the number of eigenvectors based on (6) and (7), and the low-dimensional feature vector of a new input data *x* is determined as follows

$$x_f = \Phi^T x \tag{8}$$

3.3 Hybrid Intrusion Detection Using Multi-class SVMs

Support vector machines (SVMs) are relatively new statistical learning algorithms that provide powerful tools for learning classification or regression models with good generalization ability in sparse, high-dimensional settings. The success of SVMs is due to the statistical learning theory studied by Vapnik, which gives key insights into the structural risk minimization (SRM) principle for improving generalization ability of learning machines [13]. SVM learning can be viewed as an efficient realization of Vapnik's SRM principle and lots of work has been done on revised SVM algorithms with applications in many supervised learning or pattern recognition problems [9].

As discussed previously, adaptive intrusion detection can be viewed as a multi-class pattern classification problem. In the following, we will present the multi-class SVM algorithm for adaptive network intrusion detection.

Suppose the training samples of network audit data are given as

$$\{(\vec{x}_i, y_i)\}, \quad i = 1,2,\ldots,N \qquad y_i \in \{1,2,\ldots,m\} \tag{9}$$

where N is the total number of training samples and m=5 is the number of class labels, which are normal, U2R, DOS, R2L and Probe.

Since SVMs were originally proposed for two-class problems and research work on SVMs mainly focused on binary classification problems, the multi-class pattern recognition problem of intrusion detection can be tackled by decomposing the 5-class problem to several binary problems. In the decomposition, we use the one-against-one strategy commonly applied in the literature. Based on the one-against-one strategy, the construction of multi-class SVM classifiers can be implemented by training 10 two-class SVM classifiers with different training samples.

In two-class SVM learning, a hyperplane is considered to separate two classes of samples. Following is the linear form of a separating hyperplane.

$$(\vec{w} \cdot \vec{x}) + b = 0 \qquad \vec{w} \in R^n, \; b \in R \tag{10}$$

Based on the SRM principle in statistical learning theory, the optimal separating hyperplane can be constructed by solving the following optimization problem

$$\min_{\vec{w},b} \frac{1}{2} \|\vec{w}\|^2 \tag{11}$$

subject to

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1, \quad i = 1,2,\ldots,N \tag{12}$$

In support vector learning, the optimization problem for constructing optimal hyperplane is solved by its Lagrangian dual using Karush-Kuhn-Tucker (KKT) conditions. Furthermore, to reduce the effects of noise and outliers in real data, the following soft margin techniques are usually used, which is to solve the primal optimization problem as

$$\min_{\vec{w},b} \frac{1}{2} \|\vec{w}\|^2 + C \sum_{i=1}^{N} \xi_i \tag{13}$$

subject to

$$y_i(\vec{w} \cdot \vec{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \; i = 1,2,\ldots,N \tag{14}$$

An important element for the success of SVMs is the 'kernel trick', which is to transform the above linear form of support vector learning algorithms to nonlinear ones without explicitly computing the inner products in high-dimensional feature spaces. In the kernel trick, a Mercer kernel function is employed to express the dot products in high-dimensional feature space

$$k(\vec{x}_i, \vec{x}_j) = (\vec{x}_i \cdot \vec{x}_j) \tag{15}$$

Then the dual optimization problem of SVMs for two-class soft margin classifiers is formulated as follows

$$\max_{\alpha} \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} \alpha_i \alpha_j y_i y_j k(\vec{x}_i \cdot \vec{x}_j) \tag{16}$$

subject to

$$0 \le \alpha_i \le C,\ i = 1,2,\ldots,N \text{ and } \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{17}$$

To solve the above quadratic optimization problem, various decomposition-based fast algorithms have been proposed in the literature, such as SMO [10], etc. For details on the algorithmic implementation of SVMs, please refer to [11].

In our multi-class SVM classifier, the decision function of each binary SVM is

$$f_k(\vec{x}) = \operatorname{sgn}(\sum_{i=1}^{N} \alpha_{ki} y_{ki} k(\vec{x}_{ki}, \vec{x}) + b_k) \quad k = 1,2,\ldots,m \tag{18}$$

where $f_k(\vec{x})$ is the decision function of classifier *k* and ($\vec{x}_{ki}, y_{ki}$) (*k*=1,2,…,*m*) are the corresponding training samples.

In the multi-class SVM classification for adaptive intrusion detection, a voting strategy is used: each binary classification is considered to be a voting where votes can be cast for all data points and an input is designated to be in a class with maximum number of votes.

4 Experiments on KDD-Cup99 Data

The proposed method was applied in the KDD-Cup99 intrusion detection dataset to demonstrate its effectiveness in automatic model construction and processing speed enhancement. In the experiments, a subset of KDD-Cup99 data was selected and partitioned to a training data set with 9321 records and a test set with 15705 records. We tested the proposed multi-class SVMs with PCA for dimension reduction, as well as multi-class SVMs using original data dimension. The network data records are normalized to interval [0, 1]. The performance of the classifiers includes training and testing accuracy and the processing speed during training and testing.

The experimental results are shown in Table 1 and 2. In all the experiments, RBF kernel functions are used and the width parameter is chosen as σ =0.1, which was optimized manually. In the implementation of binary SVMs, the LibSVM [12] package was used. Table 1 shows the training and testing accuracy of multi-class SVM using PCA for dimension reduction as well as SVMs without PCA. Note that the accuracy of the proposed SVM+PCA method is fairly good except that the results of class ‘R2L’ are not very satisfactory. The reason may be the amount of U2R data is very small in KDD-Cup99 dataset so that it will cause some information loss when dimension reduction is performed using PCA. However, this problem may be solved by collecting more training data of U2R attacks.

Table 1. Comparisons of training and testing accuracy

Class	SVMs with PCA		SVMs without PCA	
	Training accuracy	Test accuracy	Training accuracy	Test accuracy
Normal	99%	83.9%	100%	74.3%
Dos	99.3%	99.9%	100%	100%
Probe	94.7%	94.1%	99.1%	98.9%
U2R	97.8%	97.8%	100%	100%
R2L	60%	58.3%	100%	100%

For SVMs without PCA, although the training accuracies are slightly better than SVMs using PCA, the test accuracy of class Normal is worse than that of multi-class SVMs using PCA. This demonstrates that dimension reduction using PCA may improve the generalization ability of classifiers. For training data of R2L, the training and testing accuracies of multi-class SVM using full dimension data are very good. As discussed above, the amount of R2L data is very small (only about 20 records) so that classifiers using higher dimension data usually have better performance in accuracy. However, when data amount increases, PCA can be used to reduce data dimension without sacrificing much performance in accuracy and the generalization ability of classifiers using PCA may be improved. Furthermore, as illustrated in the following Table 2, SVMs with PCA will benefit from improved training and testing speed, which is important for high-speed network applications.

Table 2 shows the comparisons of training and testing speed of SVMs with and without PCA. It is clear that the proposed PCA+SVMs classifier is 5 times faster in training and 2 times faster in testing than conventional SVMs without PCA.

Table 2. Speed comparison of PCA+SVMs and conventional SVMs

Classifiers	Training time (s)	Testing time (s)
SVM (Original 41-dimension feature)	151.9	30.4
SVM (Using PCA for feature extraction)	33.3	14.4

5 Conclusion and Future Work

This paper proposes an adaptive network intrusion detection method using SVMs combined with PCA. The aim of the approach is to not only realize automatic construction of intrusion detection models with high accuracy but also make network IDS models to be processed as fast as possible so that the applications in high-speed networks can be feasible. Experimental results on KDD-Cup99 intrusion detection dataset show that the proposed method has comparable accuracy as that of conventional SVMs without PCA and it is much faster in processing speed than conventional SVMs. Future work may include applying and testing the proposed method in real-time intrusion detection for real network data.

References

1. Lippmann R., Cunningham R.: Improving Intrusion Detection Performance Using Keyword Selection and Neural Networks. Computer Networks, 34(4), (2000) 597--603
2. Lee W., Stolfo S. J.: Data Mining Approaches for Intrusion Detection. Proceedings of the 1998 USENIX Security Symposium, (1998)
3. Lee, W., Stolfo, S., and Mok, K.: Adaptive Intrusion Detection: A Data Mining Approach. Artificial Intelligence Review, 14(6), (2000) 533 – 567
4. Luo J., Bridges S. M.: Mining Fuzzy Association Rules and Fuzzy Frequency Episodes for Intrusion Detection. International Journal of Intelligent Systems, (2000) 687-703
5. Cannady, J.: Applying Neural Networks to Misuse Detection. In: Proceedings of the 21st National Information Systems Security Conference. (1998)
6. Mahoney M., Chan P.: Learning Nonstationary Models of Normal Network Traffic for Detecting Novel Attacks. In: Proceedings of 8th International Conference on Knowledge Discovery and Data Mining, (2002) 376-385
7. Shah H., Undercoffer J. and Joshi A.: Fuzzy Clustering for Intrusion Detection. In: Proceedings of the 12th IEEE International Conference on Fuzzy Systems. (2003) 1274-1278
8. Jolliffe I. T.: Principal Component Analysis. Springer. 2nd edition. (2002)
9. Hastie, T. J., Tibshirani, R. J. and Friedman, J. H.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag, 2001
10. Platt J.: Fast Training of Support Vector Machines using Sequential Minimal Optimization. In: B.Scholkopf, C.J.C. Burges, and A.J.Smola, editors, Advances in Kernel Methods—Support Vector Learning, Cambridge, MIT Press. (1999) 185-208
11. Lin C.-J.: Formulations of Support Vector Machines: a Note from an Optimization Point of View . Neural Computation, 13(2), (2001) 307-317
12. Fan R.-E., Chen P.-H., and Lin C.-J.: Working Set Selection using the Second Order Information for Training SVM. Technical report, Department of Computer Science, National Taiwan University, (2005)
13. Vapnik, V. N. Statistical Learning Theory. Wiley. (1998)

Improved Grid Information Service Using the Idea of File-Parted Replication

Jingwei Huang, Qingfeng Fan, Qiongli Wu, and Yanxiang He

School of Computer Science,
State Key Lab of Software Engineering Wuhan University,
Wuhan, Hubei, 430072, China
{lyqingfeng@163.com}

Abstract. The infrastructure of grid information service constituted with highly distributed information providers and aggregate directory is brought forward on the basis of the characteristic of grid information resources in this paper. *The Lightweight Directory Access Protocol* (LDAP), one of the base protocols, is also analyzed in this paper. It is put forward that LDAP is a distributed database. The dynamic updating and replication of LDAP directory tree happens frequently. To solve the problem, it has been proposed that the strategy of *fast spread* and *Cascading spread* can boost the efficiency of grid information service system. Moreover, we use file-parted replication approach to divide the LDAP database file into several blocks that are replicated parallel between LDAP sever points then. In such a way, the system efficiency of parallel processing can be boosted by margin. In addition, based on the idea forenamed, we put forward the technique infrastructure and *upload-controlling algorithm*, both of which are proven to be effective in improving the system efficiency.

1 Introduction

Grid Computing is one of the most popular fields of computing, and information service is the pivot and difficulty of Grid Computing. Foster set forth the infrastructure of grid information service constituted with highly distributed information providers and aggregate directory. He also analyzed the *Lightweight Directory Access Protocol* (LDAP), one of the base protocols. It is put forward that LDAP is a distributed database that has the characteristic of spanning flat and accessing according to the register. The server-to-server communication mode of LDAP defines how to share the LDAP directory and how to update and replicate the information between servers.

The dynamic updating and replication of LDAP directory tree that happen frequently for grid information system is distributed widely, highly fault-tolerant, dynamic and diverse. The characteristic of the grid circumstance has decided the attribute of dynamic and diversity to grid information service system. To increase the efficiency of grid information service system based on and core on the distributed database LDAP, we use the strategies of circle expand and thread expand .We have not only studied how to improve the rate of replicating and refreshing at the degree of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 704 – 711, 2005.

node, but also have studied how to improve the efficiency at the degree that below node. Moreover, we use the viewpoint of file-parted replication to divide the LDAP database file into several blocks that are replicated parallel between LDAP sever points then. In such a way, the system efficiency of parallel processing can be boosted by margin. And based on the idea forenamed, we put forward the technique infrastructure and *upload-controlling algorithm*, both of which are proved to be available in improving the system efficiency.

2 Architecture Overview

The requirements of any Grid based information system are driven by basic properties of the Grid environment. Information sources are necessarily distributed and individual sources are subject to failure. The total number of information providers can be large, and both the types of information sources and the ways in which information is used can be highly varied.

Our Grid information service architecture (Figure 1) comprises two fundamental entities: highly distributed *information providers* and specialized *aggregate directory* services.

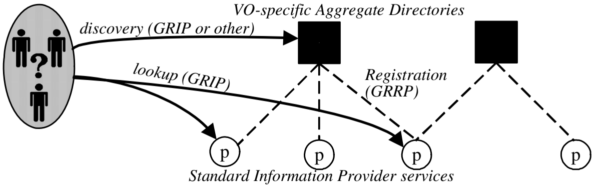

VO-specific Aggregate Directories
discovery (GRIP or other)
lookup (GRIP)
Registration (GRRP)
?
p
p
p
p
Standard Information Provider services

Fig. 1. Architecture overview. Using the Grid Information protocol (GRIP), users can query aggregate directory services to discover relevant entities. Using the Grid Registration Protocol (GRRP), users can make the periodic and continuous update of the information

An information provider is defined as a service that speaks two basic protocols. The Grid Information Protocol (GRIP) is used to access information about entities, while the Grid Registration Protocol (GRRP) is used to notify aggregate directory services of the availability of this information.

We define an aggregate directory as a service that uses GRRP and GRIP to obtain information (from a set of information providers) about a set of entities, and then replies to queries concerning those entities. As we explain below, an aggregate directory can itself adopt GRIP as the protocol by which others query it (and, for that matter, GRRP as the protocol that it uses to notify others of its existence).

The definitions of GRIP and GRRP enable a clean separation of concerns between enquiry and discovery. We observe that this architecture embodies many of the same structural principles as the World-Wide Web, unarguably the largest federated information system. GRIP corresponds to HTTP, and aggregate directories to search engines.

3 The LDAP Information Model

Grid information service performs on the basis of LDAP model that consists of the Directory Information Tree and object. We adopt the standard Lightweight Directory Access Protocol (LDAP) as the protocol for GRIP and GRRP. LDAP defines the data model, query language, and wire protocol.

LDAP directory is also a distributed type-database but not a relation-database. It is a span-platform protocol fitting the condition in which accessing is frequent. LDAP can be distributed and accessed as entities, in which their attribute is alterable.

There are two communication models in LDAP protocol: client-serve and serve-serve communication. Client-serve communication allows the programme of the client to connect the LDAP serves and then to establish, search, modify and delete the data. Serve-serve communication defines the way that many servers share one LDAP directory and refresh, copy the information between servers.

4 Advance the Efficiency Using Dynamic Replicating

The dynamic updating and replication of LDAP directory tree would happen frequently for grid information system is distributed widely, highly fault-tolerant, dynamic and diversiform. Thus it is worthwhile for us to study how to advance the efficiency of the grid information service system. Having studies the strategy of dynamic replicating, and compared many kinds of replicating strategies, we finally discover that the strategy of *fast spread* and *Cascading spread* can boost the system efficiency.

Strategy 1: Cascading Replication

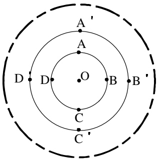
The best analogy for this strategy is a three-tiered fountain. The water originates from the top. When it fills the top ledge it overflows to the next level. When this level also overflows the water reaches down to the lowest part. The data in this strategy flows in a similar way. Once the threshold for a file is exceeded at the root, a replica is created at the next level, but on the path to the best client. Hence the new site for the replica is an ancestor of the best client. Once the number of requests for the file is exceeded at Level 2 it is then replicated at the next lower tier and so on. A very popular file may ultimately be replicated at the client itself. The client that requests a file stores a copy locally. Since these files are large (2 Gigabytes each) and a client has enough space to store only one file at a time, the files get replaced quickly.

The server periodically identifies the popular files and propagates them down the hierarchy. Note that the clients are always located at the leaves of the tree but any node in the hierarchy can be a server. *Strategy1* is illustrated in Figure 2a.

Strategy 2: Fast Spread

In this method a replica of the file is stored at each node along its path to the client. That is to say, when a client requests a file, a copy is stored at each tier on the way. This leads to a faster spread of data. *Strategy 2* is illustrated in Figure 2b.

The experiment results indicate that depending on what is more important in the grid scenario, lower response times or lesser bandwidth consumption, a tradeoff between Cascading and Fast Spread can be made. If the chief aim is to elicit faster responses from the system, Cascading might work better. On the other hand if conserving bandwidth were of top priority, Fast Spread would be a better grid replication strategy.



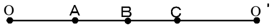


Fig. 2a. Cascading Replication **Fig. 2b.** Fast Spread

5 File-Parted Replication

Having studied how to improve the rate of replicating and refreshing at the degree of node, we will study how to improve the efficiency at the degree below node as follows.

In Files-Parted Replicating we adopt P2P as the way that makes the parallel efficiency. We will describe how to use Files-Parted Replicating to advance the dynamic refreshing efficiency of LDAP distributed directory database.

5.1 The Principle of Files-Parted Replicating

LDAP servers burden all the cost of download when the file of LDAP directory database can be downloaded. With Files-Parted Replicating, when multiple people are downloading the same file at the same time, they upload pieces of the file to each other. This redistributes the cost of upload to down-loaders, (where it is often not even metered), thus making LDAP servers a file with a potentially unlimited number of down-loaders affordable, which is indicated in figure 3.

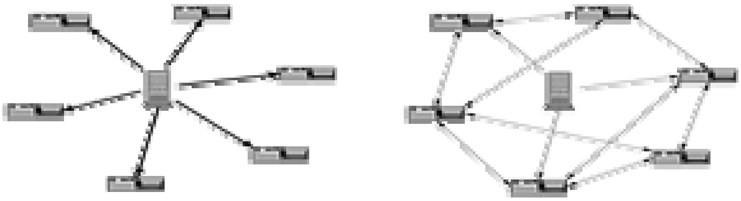


Fig. 3. The Principle of Files-Parted Replicating

LDAP servers make the deployment of Files-Parted Replicating. In the meantime, the down-loaders get the file they want depending on Files-Parted Replicating as efficient as possible.

5.2 Technical Framework

5.2.1 Publishing Content

A static file with the extension torrent is put in the LDAP distributed database on an ordinary web server. The torrent contains information about the file, its length, name, and hashing information, and the url of a tracker. Trackers are responsible for helping LDAP distributed database find each other. They speak a very simple protocol layered on top of HTTP in which a LDAP distributed database sends information about what file it is downloading, what port it is listening on, and similar information, and the tracker responds with a list of contact information for peers which are downloading the same file. LDAP distributed database then uses this information to connect to each other.

5.2.2 Peer Distribution

All logistical problems of file replication are handled in the interactions between LDAP servers. Some information about uploads and downloads rates of LDAP servers is sent to the tracker. The tracker's responsibilities are strictly limited to helping LDAP servers find each other. Trackers are the only way for LDAP servers to find each other, and the standard tracker algorithm is to return a random list of LDAP servers. Random graphs have very good robustness properties. Many LDAP server selection algorithms result in a power law graph, which can get segmented after only a small amount of churn.

In order to keep track of which LDAP server has what, File-Parted Replicating cuts files into pieces of fixed size, typically a quarter megabyte. Each downloader reports to all of its peers which piece it has. To verify data integrity, all the pieces are included in the .frr file. Every LDAP server continuously downloads pieces from all peers, and they can. They of course cannot download from peers they aren't connected to, and sometimes LDAP servers don't have any pieces they want or won't currently let them download.

5.2.3 Pipelining

When transferring data, it is very important to always have several requests pending at once, to avoid a delay between pieces being sent, which is disastrous for transfer rates. File-Parted Replicating facilitates this by breaking pieces further into sub-pieces over the wire, typically sixteen kilobytes in size, and always keeping some number, typically five, request pipelined at once. Every time a sub-piece arrives a new request is sent out. The amount of data to pipeline has been selected as a value, which can reliably saturate most connections.

5.2.4 Piece Selection

Selecting pieces to download in a good order is very important for good performance. A poor piece selection algorithm can result in having all the pieces which are currently on offer or, on the flip side, not having any pieces to upload.

*Strict Priority:*File-Parted replication's first policy for piece selection is that once a single sub-piece has been requested, the remaining sub-pieces from that particular piece are requested before sub-pieces from any other piece. This does a good job of getting complete pieces as quickly as possible.

*Rarest First:*When selecting which piece to start downloading next, peers generally download pieces which the fewest of their own peers have first, a technique we refer to as 'rarest first'. This technique does a good job of making sure that peers have pieces which all of their peers want, so uploading can be done when wanted. It also makes sure that pieces that are more common are left for later, so the likelihood that a peer that currently is offering upload will later not has anything of interest is reduced.

*Random First Piece :*An exception to rarest first is when downloading starts. At that time, the peer has nothing to upload, so it's important to get a complete piece as quickly as possible. Rare pieces are generally only present on one peer, so they would be downloaded slower than pieces which are present on multiple peers for which it's possible to download sub-pieces from different places. For this reason, pieces to download are selected at random until the first complete piece is assembled, and then the strategy changes to rarest first.

Endgame Mode: Sometimes a piece will be requested from a peer with very slow transfer rates. This isn't a problem in the middle of a download, but could potentially delay a download's finish. To keep that from happening, once all sub-pieces which a peer doesn't have are actively being requested it sends requests for all sub-pieces to all peers. Cancels are sent for sub-pieces which arrive to keep too much bandwidth from being wasted on redundant sends. In practice not much bandwidth is wasted this way, since the endgame period is very short, and the end of a file is always downloaded quickly.

5.3 Upload-Controlling Algorithm

File-Parted Replicating has no central resource allocation. Each server is responsible for attempting to maximize its own download rate. The server does this by downloading from whoever they can and deciding which serves to upload to via a variant of tit-for-tat. To cooperate, peers upload; and to not cooperate they 'choke' peers. Choking is a temporary refusal to upload; It stops uploading, but downloading can still happen and the connection does not need to be renegotiated when choking stops.

Upload-controlling algorithm is necessary for good performance. A good choking algorithm should utilize all available resources, provide reasonably consistent download rates for everyone, and be, to some extent, resistant to peers only downloading and not uploading.

Pareto Efficiency: Upload-controlling algorithm means replicating the peers to the peers that upload themselves. Thus the replicating between many peers will happen anytime and the connection brings forward better uploading. Thus bring high transfer rate.

Upload-controlling: On a technical level, each server always guarantees a fixed number of other peers (default is four). Decisions as to which peers to upload are based strictly on current download rate. The current implementation essentially uses a rolling 20-second average. To avoid situations in which resources are wasted by rapidly choking and uploading peers, the server recalculate who they want to choke and who want to upload once every ten seconds, and then leave the situation as is until the next ten second period is up.

Optimistic Uploading: The servers which provide the best download rate would suffer from having no method of discovering if currently unused connections are better than those being used. To fix this, at all times a peer has a single 'optimistic upload', which is uploaded regardless of the current download rate from it. Which peer is the optimistic upload is rotated every third reupload period (30 seconds). 30 seconds is enough time for the upload to get to full capacity, the download to reciprocate, and the download to get to full capacity.

Anti-snubbing: Occasionally a peer will be choked by all the other peers from which it has been downloading. In such cases it will usually continue to get poor download rates until the optimistic upload finds better peers. To mitigate this problem, when over a minute goes by without getting a single piece from a particular peer, File-Parted Replicating assumes it is ’snubbed’ by that peer and doesn’t upload to it except as an optimistic upload. This frequently results in more than one concurrent optimistic upload, (an exception to the exactly one optimistic upload rule mentioned above), which causes download rates to recover much more quickly when they falter.

Upload Only: Once a peer has finished downloading, it no longer has useful download rates to decide which peers to upload to. The current implementation then switches to preferring peers which it has better upload rates to, which does a decent job of utilizing all available upload capacity and preferring peers which no one else happens to be uploading to at the moment.

5.4 Real World Experience

The number of complete down-loaders (‘seeders’) and incomplete down-loaders (‘leechers’) of a large deployment of an over 400 megabyte file over time is showed in figure 4.

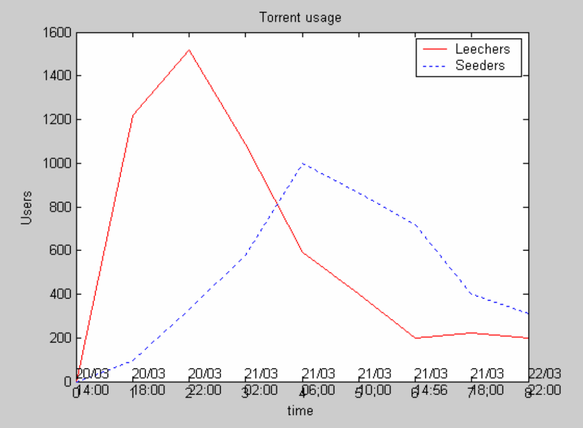


Fig. 4. Seeders and Leechers of a large deployment of a file

In the figure, the “lechers” increases fast after the file become available, but the number exponentially decreases after reaching the peak value. In contrast, the “seeders” increases slower, the peak value appear later, and after that it decrease slowly in which the integral is big. The dramatic increase and decrease of the “leechers” and the stabilization of the "seeders" prove that the dynamic update and replication of the LDAP server carry out and finish rapidly, which improve that the system efficiency can be boosted by margin using the FPR further more.

File-Parted Replicating is based on the download tool Bittorrent that is popular on Internet. BitTorrent not only is already implemented, but is already widely deployed. It routinely serves files hundreds of megabytes in size to hundreds of concurrent down-loaders. The largest known deployments have had over a thousand simultaneous down-loaders.

6 Conclusions and Future Work

The strategy of *fast spread* and *Cascading spread* can boost the efficiency of grid information service system. We use the viewpoint of file-parted replication to divide the LDAP database file into several blocks that are replicated parallels between LDAP sever points. In such a way, the system efficiency of parallel processing can be boosted by margin. In addition, based on the idea forenamed, we put forward the technique infrastructure and *upload-controlling algorithm*, both of which are proven to be available in improving the system efficiency.

We are currently working on how to advance the efficiency of replicating and refreshing at the degree of node. We also plan to explore the same type of thing at the degree that below node, the load-balance of LDAP directory where the replicating will bring the best system effect and so on.

References

[1] Karl Czajkowskiy Steven Fitzgeraldz Ian Fosterx Carl Kesselman . Grid Information Services for Distributed Resource Sharing. *Proc. 10th IEEE International Symposium on High-Performance Distributed Computing* (HPDC-10), IEEE Press, 2002.

[2] I. Foster, C. Kesselman and S. Tuecke. The anatomy of the Grid: Enabling scalable virtual organizations. *Intl. Journal of Supercomputing Applications*, (to appear) 2002. http://www. globus.org/research/papers/anatomy.pdf.

[3] TimothyA.Howes. Lightweight Directory Access Protocol. http://www.kingsmountain. com/ directory/doc/ldap/ldap.html ：

[4] He Yanxiang Fan Qianfeng Zhang Lifei . Design of dynamic replication strategies for a grid computing .*Computer Engineering* 2004.2

[5] Bram Cohen. Incentives Build Robustness in BitTorrent 2003.5. http://www.bittorrent. com/ bittorrentecon.pdf

Dynamic Shape Modeling of Consumers' Daily Load Based on Data Mining[1]

Lianmei Zhang , Shihong Chen, and Qiping Hu

Electrical Engineering College, Wuhan University,
Hubei Province, China, 430072
lloottuuss@163.com

Abstract. The shape characteristic of daily power consumption of consumers can be applied to guide their power consumption behaviors and improve load structures of power system. It is also the basis to obtain the shape characteristic of daily power consumption of a trade and conduct researches in the state estimate of distribution networks etc. Traditional analytical approaches are limited to qualitative analysis with a small coverage only. We propose a model which can perform in-depth analysis of customer power consumption behaviors by data mining through similar sequence analysis to overcome the drawbacks of traditional approaches. The model uses real-time sampling of the energy data of consumers to form the shape characteristic curves. The application and testing of the model under an instance is analyzed in this paper.

1 Introduction

China has entered into another round of serious power crises since 2003. Under this situation, through implementing the economic means of "time dependant energy pricing", the crisis can be partially relieved by consumers' self adjusting power consumption behaviors. In the meantime, via analyzing the consumer's power consumption behaviors, power companies can instruct the consumers to proper ways of using electricity. This is also an effective approach to relieve the crisis of power shortage. Power supply companies can also adjust their policies according to the results of customer consumption characteristics.

At present, two approaches are generally adopted to analyze consumers' power consumption behaviors.

1. Statistical analysis method that directly accumulates consumers' power consumption. Although this method can obtain accurate quantities, its period is too long and only carries out contemporary comparison and elicits qualitative conclusion of a long time-span.

2. Detailed analysis of the load components of large electricity consumers to learn their power consumption characteristics. The drawback of this approach is that the coverage can be too limited because only a few typical large electricity consumers can

[1] This work was supported by the National 863 Project, Republic of China, under Contract 2002AA111010 , 2003AA001032.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 712 – 719, 2005.

be selected for analysis. The method only elicits qualitative conclusion and it is hard to gain evidence to prove whether the conclusion can be popularized. Meanwhile, as the time-span of this method is long, it is very difficult to conduct an analysis of this kind on a monthly basis.

The analysis of consumers' power consumption behaviors on the basis of simple statistical method can no longer meet the needs of the existing consumers and electric power industry. Therefore, new technologies are in increasing needs so as to obtain more detailed and accurate information on power consumption behaviors of consumers.

In recent years, data mining has been applied in the electric power industry, which mainly concentrates on the following three respects:

1. Load forecast [1][2].
2. Analysis of the state and faults of a power system [3].
3. Business analysis of electricity consumers. The behaviors of electricity consumers differ greatly from those of the consumers of other trades. Hence, there have been only a small number of research work on applying data mining to this area, especially on how to fully explore the results of mining [4][5].

The objective of this paper is to analyze consumers' power consumption behaviors and to obtain the dynamic model of consumers' daily power-consumption.

2 Modeling of the Dynamic Shape Curves of Consumers' Daily Load

In distribution systems, it is appropriate to choose consumers with voltage levels of 6KV and 10KV for research. This paper takes the mass data acquired from distribution transformers of 6KV and 10KV as the basis for analysis. We also ascertain the time interval of data acquisition according to the interval of the time acquired by distribution transformers. The data can be acquired even every 15 minutes.

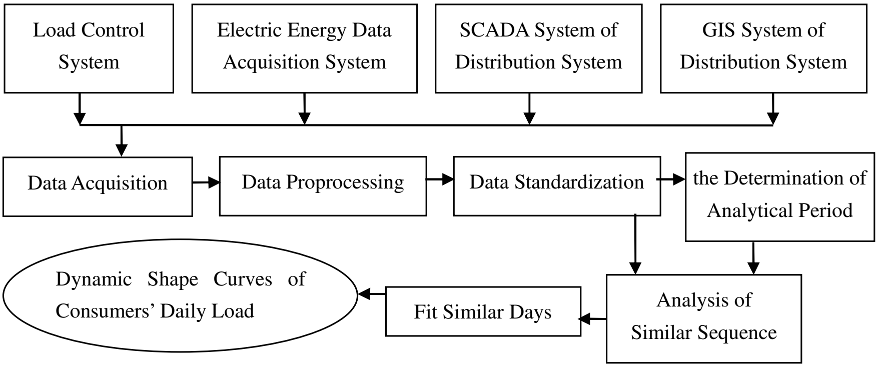

Load Control System
Electric Energy Data Acquisition System
SCADA System of Distribution System
GIS System of Distribution System
Data Acquisition
Data Proprocessing
Data Standardization
the Determination of Analytical Period
Dynamic Shape Curves of Consumers' Daily Load
Fit Similar Days
Analysis of Similar Sequence

Fig. 1. Dynamic Shape Model of Consumer Load

When the time interval of data acquisition is 15 minutes, active power and reactive power can be used for customer consumption characteristic analysis.

The characteristics of daily power consumption include shape characteristics and quantity characteristics. The shape characteristic refers to the shape of the consumers' power consumption changing over time during the sampling period in a day. It mainly reflects the phenomena of peaks, flats, and valleys etc. The quantity characteristic refers to the actual power consumption. The analysis of shape characteristic is as significant as that of quantity characteristic.

This paper focuses on the analysis of the shape characteristic of daily power consumption. Accordingly, we propose the following dynamic shape model of consumers' load (see Figure 1).

2.1 Data Acquisition

According to the above-mentioned approaches, the needed data can be extracted from relevant automation systems such as Load Control System, SCADA System of Distribution System, Geographic Information System of Distribution System, and Electric Energy Data Acquisition System. These data are then saved into the Data Warehouse – see Fig 1.

2.2 Data Preprocessing

For the raw data in the Data Warehouse, apart from the preprocessing measures such as null value clearing and clearing of days without power supply. The most important one is to correct the abnormal data.

Due to the malfunctions of the measurement devices and the communication problems occurring in the process of transmitting the data to the Control Center, the measured data may be false or lost, causing the inaccuracy of the data. Consequently it is necessary to correct such abnormal data.

We divide a day into n time intervals, t_i indicating the No. i time interval $i(1 \le i \le n)$, kWh_i (or $kVARh_i$) indicats the active (or reactive) electric energy consumed at the No. i time interval. If the present value of kWh_i (or $kVARh_i$) is inaccurate, use two methods (altogether 3 procedures) to correct kWh_i (or $kVARh_i$), and the current interval values of the active energy $[kWh_{i\min}, kWh_{i\max}]$ (or $[kVARh_{i\min}, kVARh_{i\max}]$) can be acquired[2].

Procedure 1:
Using historical data

$$kWh_{i,today} = \frac{kWh_{i-1,today}}{kWh_{i-1,day\_before}} \times kWh_{i,day\_before} \quad . \tag{1}$$

where *day_before* is the day before weekday or the weekend / holiday cycle.

[2] In this paper, the methods used for the active energy is similar to the methods for the inactive energy.

Procedure 2:
If there are reliable data points both before and after the data to be corrected, we adopt Newton interpolation method to correct the abnormal data. According to consumers' power consumption situation during each sampling period, it can be seen that the power consumptions in a big range are independent of one another. On account of the high reliability of modern communication technology, it can be considered that the normal data are also highly reliable. This paper employs the third-order Newton interpolation method and selects only two reliable points respectively from before and after the point to be corrected (altogether 4 points) to correct the abnormal data. This both ensures the effect of approaching the true value and lowers the interpolation costs.

Procedure 3:
After correcting the abnormal data, we use interval value $[kWh_{i\min}, kWh_{i\max}]$ (or $[kVARh_{i\min}, kVARh_{i\max}]$)to indicate the amended kWh_i (or $kVARh_i$).

$$kWh_i = [kWh_{i\min}, kWh_{i\max}] = [\max(f(x), kWh_{i,today}), \min(f(x), kWh_{i,today})] \, . \tag{2}$$

Taking the data from 21:00 to 22:00 on January 13, 2003 of a hotel as an example, the raw data are: 84， 84， -4700， 96， 86, where –4700 is abnormal and after amendment the interval value of it is $[91.667,93]$.

2.3 Data Standardization

As the periods of time of different customers are different, the actual active and reactive energy are not the same. To reflect the shape characteristic of daily electrical degrees, it's meaningless to compare between the actual active (reactive) energy directly, and the transition is a necessity.

The per unit value method is employed in the transition. Take the maximum active (reactive) energy of each consumer in the sampling period every day as the fiducial value. The per unit value will be acquired through dividing the active (reactive) energy in each sampling period by the corresponding fiducial value. The per unit values acquired range from 0 to 1.

2.4 Algorithm of Similar Sequences Containing Interval Data

We use the interval number $f = [a, b]$ to represent interval values. The distance between two interval values is described by SS . The SS of interval values f_1 and f_2, denoted as $ss(f_1, f_2)$.

Let $f_1 = [a_1, b_1]$ and $f_2 = [a_2, b_2]$ be two interval numbers; $\bar{f_1} = (a_1 + b_1)/2$ and $\bar{f_2} = (a_2 + b_2)/2$ are respectively centers of intervals f_1 and f_2 .The interval SS between f_1 and f_2 is defined as the following:

$$ss(f_1, f_2) = dist(\bar{f_1}, \bar{f_2}) + \frac{|f_1| + |f_2|}{4} \, . \tag{3}$$

Given two sequences $X=(x_1,x_2,\cdots x_n)$ and $Y=(y_1,y_2,\cdots y_n)$, $x_i=\lfloor \underline{x}_i,\bar{x}_i \rfloor$ and $y_i=\lfloor \underline{y}_i,\bar{y}_i \rfloor$ are interval numbers .

X and Y have similarity if and only if ,given three numbers δ,ε,l , there exist subsequences $X_s=(x_{i1},x_{i2},\cdots x_{il})$ and $Y_s=(y_{i1},y_{i2},\cdots y_{il})$ that are subsequences in X and Y ,respectively , and they satisfy the following conditions[6]:

(1) for any $1\le k\le l-1$

(2) for any $1\le k\le l$, $|i_k-j_k|\le\delta$

(3) for any $1\le k\le l$

(4) $ss(\lfloor \underline{x_{ik}},\overline{x_{ik}} \rfloor, \lfloor \underline{y_{ik}},\overline{y_{ik}} \rfloor)\le\varepsilon$

Moreover, let $\gamma=l/n$ be the coefficient representing the ratio of length of the subsequence.

In the above definition, Condition 1 means that the elements of a subsequence are not required to be consecutive, but their relative order has to conform with the order in the original sequence.

Condition 2 imposes a constraint on the maximum shift of the two sequences along the time axis.

Condition 4 requires that the maximum SS of the two elements being compared should not be greater than a threshold.

Given ε , suppose H is the set of all possible γ values of sequences X and Y , that is

$$H=\{\gamma_i \ |X,Y are(\gamma,\varepsilon,\delta)-similarity\} \tag{4}$$

Then the similarity degree $Sim_{\varepsilon}(X,Y)$ is defined as the following:

$$Sim_{\varepsilon}(X,Y)=\max(\gamma_i), for\forall\gamma_i\in H \tag{5}$$

2.5 The Determination of Analytical Period

Different consumers vary a lot in their power consumption behaviors. Even for the same consumer, its power consumption behaviors may vary in different days. Analysis of data by adopting the above-mentioned similar sequences method indicates that it is reasonable to analyze consumers' power consumption behaviors in months.

When they were analyzed in months, we should make sure whether the shape characteristics of daily power consumption during one month are all similar. If $\varepsilon=0.1,\delta=10,\gamma=90/96$. See Table 1 for the result of similar sequence analysis for one consumer during one month.

It illustrates that among the 31 days of January 2003, this consumer has similar shape characteristics of daily power consumption in 27 days, which account for 87% of the total number of days in January. Considering the power consumption on Saturdays and Sundays may be different from that of working days, we can regard that when the similar days account for 65% of the total number of days in a month, the shape characteristics of daily power consumption are similar in this month.

Table 1. The Result of Similar Sequence analysis for One Consumer During One Month

2003-1-1	2003-1-2	2003-1-3	2003-1-4	2003-1-6	2003-1-7
2003-1-8	2003-1-9	2003-1-10	2003-1-11	2003-1-13	2003-1-14
2003-1-15	2003-1-16	2003-1-17	2003-1-18	2003-1-20	2003-1-21
2003-1-22	2003-1-23	2003-1-24	2003-1-25	2003-1-27	2003-1-28
2003-1-29	2003-1-30	2003-1-31			

Through similarity analysis of 1080 households per month during 3 years, we have found that 92% of them have similar shape characteristics of daily power consumption in every month.

2.6 Shape Characteristic Curve of Daily Power Consumption

We can obtain the shape characteristics of daily power consumption of this consumer in similar days in this month, namely the shape characteristic curve of its daily power consumption. The algorithm is described as follows:

1. Rank the consumers and get the number of consumers N_m ;

2. Assume $m=1$, when $m \le N_m$ Do;//Indicating taking the data of consumer No. m .

3. Rank the data of consumer No. m according to time; get the number of months N_n of consumer No. m .

4.Assume $n=1$, when $n \le N_n$ Do; // Indicating taking the data of month No. n of consumer No. m .

5. Search similar sequences of the data of month No. n of consumer No. m , and get all similar days.

6. Rank the similar days of month No. n month of consumer No. m and get the number of similar days N_i .

7. Assume $N_j = K$; // Indicating that one day can be divided into K time intervals.

8. Assume $j=1$,when $j \le N_j$ Do;//Indicating taking the data of time interval No. j .

9. Assume $i=1$,when $i \le N_i$ Do;// Indicating taking the data of similar day No. i .

10. $kWh_{m,n,i,j}$; or $kVARh_{m,n,i,j}$; indicates the per unit value of quantity of active energy (or reactive energy) consumed at time interval No. j of day No. i of month No. n for consumer No. m .

11. Fit similar days; // the per unit value of quantity of active energy $kWh_{m,n,j}$ (or per unit value of quantity of reactive energy $kVARh_{m,n,j}$) consumed at time interval No. j of month No. n for consumer No. m is:

$$kWh_{m,n,j} = \frac{1}{N_i}\sum_{i=1}^{N_i} kWh_{m,n,j,i}\ ,\ kVARh_{m,n,j} = \frac{1}{N_i}\sum_{i=1}^{N_i} kVARh_{m,n,j,i} \tag{6}$$

12. Termination.

3 Case Studies

The model of shape characteristics of power consumption in similar days can be served as the basis of obtaining model of the shape characteristics of daily power consumption for a trade and calculating the probability flow, which may also be used directly as the guidance for power supply and power consumption.

For instance, for the obtained shape characteristics of power consumption in similar days, power supply companies could compare them with the shape characteristics of daily power consumption of power system. During the months with power supply shortages, according to the shape characteristics of daily power consumption in similar days, the companies could give guidance on consumers' power consumption behaviors by combining their respective production characteristics.

The shape of power consumption in similar days in Jan.2003 of a construction company is shown in Figure 2.

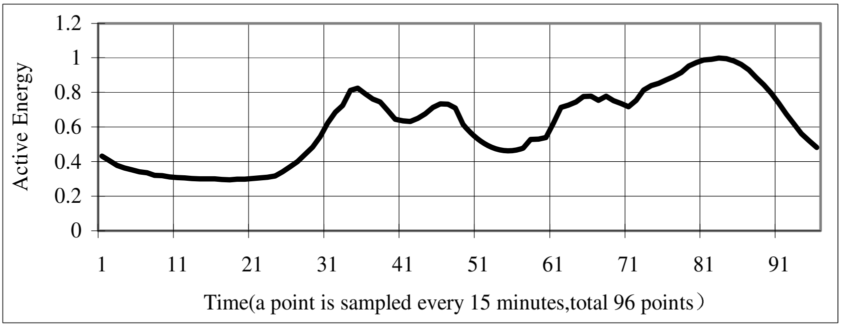


Fig. 2. Shape of power consumption in similar days in Jan.2003 of one construction company

As is shown in Figure 2, the peak period of power consumption of this company is from 18:00 to 23:00, which happens to be the same with that of the system. Through conducting a thorough investigation, we have found that this company could implement the production mode of double shifts that separates valley periods from average periods so as to avoid the peak period of power consumption. In view of the suggestion put forward by the power company, this company has changed its production time and further improved its shape characteristics of daily power consumption.

4 Conclusion

In this paper, dynamic shape characteristic curves of daily power consumption of consumers can be obtained by data mining through similar sequences analysis. The model uses the real-time sampling of the energy data of consumers to form the shape characteristic curves and makes the curves changing slowly along month but not year. The approach can be used to all electricity consumers whose load data can be read automatically.

References

1. Mori, H.; Kosemura, N.; Kondo, T.; Numa, K.: Data mining for short-term load forecasting. Power Engineering Society Winter Meeting, 2002. IEEE ,Volume: 1 ,27-31 Jan. 2002 vol.1. (2002) 623–624
2. Schalk W. Heunis,, and Ron Herman: A Thermal Loading Guide for Residential Distribution Transformers Based on Time-Variant Current Load. IEEE TRANSACTIONS ON POWER SYSTEMS, VOL. 19, NO. 3, AUGUST 2004 (2004) 1294–1298
3. Tso, S.K.; Lin, J.K.; Ho, H.K.; Mak, C.M.; Yung, K.M.; Ho, Y.K.: Data mining for detection of sensitive buses and influential buses in a power system subjected to disturbances. Power Systems, IEEE Transactions on ,Volume: 19 ,Issue: 1 ,Feb. 2004 (2004) 563–568
4. Chang, R.F.; Lu, C.N.: Load profile assignment of low voltage customers for power retail market applications. Generation, Transmission and Distribution, IEE Proceedings- ,Volume: 150 ,Issue: 3 ,13 May 2003 (2003) 263 – 267
5. Gulski, E.; Quak, B.; Wester, F.J.; de Vries, F.; Mayoral, M.B.: Application of data mining techniques for power cable diagnosis. Properties and Applications of Dielectric Materials, 2003. Proceedings of the 7th International Conference on ,Volume: 3 ,1-5 June 2003 vol.3 (2003) 986–989
6. Stephen Shaoyi Liao, Tony Heng Tang, and Wei-Yi Liu: Finding Relevant Sequences in Time Series Containing Crisp, Interval, and Fuzzy Interval Data. IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS—PART B: CYBERNETICS, VOL. 34, NO. 5, OCTOBER 2004 (2004) 2071–2079

A Study on the Mechanism of Virtual SAN-Based Spatial Data Storage with Double-Thoroughfare in Grid*

Jinsong Gao[1, 2], Wen Zhang[1], and Zequn Guan[1]

[1] School of Remote Sensing and Information Engineering, Wuhan University,
430079 Wuhan, P.R. China
[2] Department of Information Management, Huazhong Normal University,
430079 Wuhan, P.R. China
jsgao@public.wh.hb.cn

Abstract. Combining the advantages of the network and virtual SAN technology, the paper focuses on the characteristics of spatial data storage, by proposing a virtual SAN-based architecture of grid GIS data storage. Meanwhile, its corresponding experimental system has been designed to verify this proposal. In order for the storing facilities, FC and iSCSI data based double-thoroughfare have been designed in the proposed system to fulfill the efficient storage and retrieval of huge amount of spatial data. This solution possesses certain practical and applicable values from realizing and serving a technological foundation for exposable access and open interoperability of spatial data.

1 Introduction

Huge amount of spatial data and storage costs are the two major difficulties that GIS technologies have been facing. Adopting economical and practical storage solutions, utilizing existing storage space effectively, and reducing storage cost rationally are the problems needed to be solved urgently in all worldwide data centers [1]. As the distributed storage technology has been used in GIS, the inter-applications among isomerous and remote GIS technologies have been gradually increased. Therefore, it is the need of the storage system that can process an intellectual spatial information management; in other words, it is able to cover the heterogeneities raised from different storage systems, to remove the spatial information-isolated islands and knowledgeable islands, and finally to implement inter-platform shares and interoperability of spatial information [2].

The appearance of grid technology would greatly fulfill and even promote the further development of GIS. Simultaneously along the development of grid computation, the research of grid GIS has been extensively studied world widely [3, 4]. At the same time, the storage and management of spatial data under a grid environment have become ones of the important considerations in all GIS-related

* Supported by Natural Science Foundation of Hubei Province(No.2004ABA013) and Innovative Foundation of Wuhan University(2003-2005).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 720 – 727, 2005.

fields. As a common agreement among many experts in grid computation field, SAN technology is a feasible and available way to solve the storage problem in a network environment [5]. In this paper, a different sensible method is proposed to build a virtual SAN-based grid GIS data storage system.

2 Virtualized Technique of SAN

There are two methods in virtualization of SAN: out-of-band and in-of-band [6, 7]. The first virtualized SAN is the focal discussion of this paper. Figure 1 shows that in the architecture of an out-of-band virtualized SAN, the storage configuration and control of information are provided by the virtualizing engine out of data gateway, which connects and manages the storage network. The paths of data and controlled information are separable. The stored spatial data are transferred through the data channel. Being compared with the in-of-band virtualized SAN, it reduces the delay from a network, enhances the usability of the bandwidth to raise the performance of the whole system, and also avoids the single-point failure and bottleneck in the system. Although its initial cost may be relatively higher from the investing perspective in a short-period, the out-of-band virtualized SAN possesses the expansibility of a good system to add new devices on-line without the need of paying extra-costs on hardware or software so that the continuity of the system upgrade can be maintained with a low investment cost [8]. Despite that it is required to install an agent software in the server, in fact, the data security is strengthened with the agent installation. At one side, the system can be protected by breaking the data channel at once during being attacked, and at the other side, the data would not be lost when the system is failed under the agency existence.

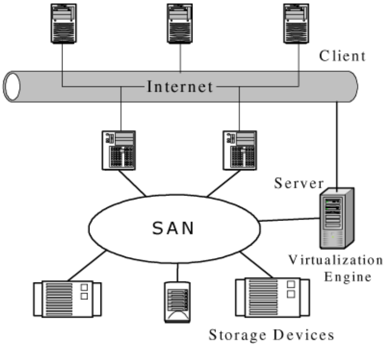


Fig. 1. The architecture of an out-of-band virtualized SAN

Virtual SAN is to implement data sharing between servers and to increase the connectivity between the physical devices; at the same time, the heterogeneous characteristics between the servers can be also concealed. The storage device of virtual SAN is not restricted with the capacity, speed and reliability of any physical devices. It is a very significant point for the implementation of spatial data sharing and interoperability in GIS.

3 Mechanism of Grid Virtual Storage with Double-Thoroughfare

3.1 Design of Structure

The stored data types and levels are different in all kinds of storing equipments; on the whole, it can be divided into products, namely processed data that can be directly provided to users, and semi-manufactured products that are the data required to be processed by data servers before being offered to users. The traditional way is to link storing equipments with servers, and users-needed data must be transmitted via servers. Consequently, the large number of data does not only reduce the efficiency during transmitting process but also probably leads to the situation of data damage or loss. Utilizing the flexible feature of SAN storing structure, we can design an appropriate solution for spatial data storage and applicable condition regarding grid as a platform. The data service can be wrapped up by grid middleware as a unified user interface, and in the aspect of storage, the access to produced data and semi-manufactured produced data are considered separately. Based on the foregoing idea, we have designed the double-thoroughfare structure for spatial data storage. The graphical structure of virtual SAN-based grid GIS data storage system with double-thoroughfare is illustrated in Figure 2. In the architecture, four tiers are built to implement various functions, including user interface, grid GIS functions, spatial data server, and data storage.

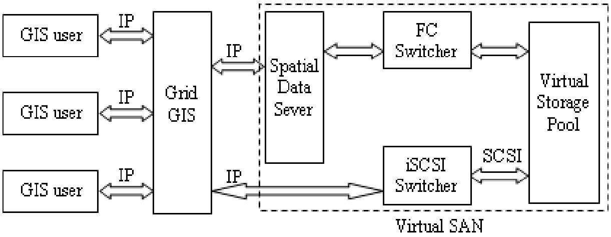


Fig. 2. The structure of virtual SAN-based grid GIS data storage system with double-thoroughfare

(1) User interface: A user interface in grid GIS is provided for data users who can access the spatial data directly without any middle tiers. In other words, it is exposable for users to obtain the stored data from the grid GIS data storage system.

(2) Grid GIS functions: This tier is organized from both grid and GIS components. A uniform interface is provided to users for spatial data shares, analyses and operations. It also provides a powerful resource management as well as spatial analysis function, and is the closest level with users' relationship among all levels.

(3) Spatial data server: Due to the multi-sources and complicacy of spatial data, spatial data server acts a very important role in the grid GIS data storage system, as it is a bridge to communicate users and storage device for the direct control and management of all storage resources. Generally, the spatial data server is structured with many application servers, but its logical mirror is like a virtual super data server.

(4) Data storage: The logical mirror of data storage tier is a seamlessly connective virtual storage pool which is virtualized by either out-of-band or in-of-band virtualized engine. It conceals the heterogeneity of physical devices and is controlled by the server.

As shown in Figure.2, the communication between the spatial data server and each storage device is highly reliable with wide bands, as they are connected by a special fiber channel. The spatial data server is directly connected with the IP network. It receives requests from users and then assigns authorization and mission to storage devices. Finally, authorized storage device is connected with IP network through the iSCSI switcher, which can offer produced data or data storage service to users, and communicates with grid GIS users. If the working states of SAN are divided into five categories, "acceptance of requests", "management of requests", "management of semi-manufactured produced data", "submission of server data" and "submission of stored equipment data", with FC and iSCSI double data thoroughfare, the prior four states can be simultaneously processed with submission of data at a high speed. Meanwhile, the produced data are submitted directly from storing equipments to clients without holding the band of server. Therefore, the conflict for band from user requests to device responses can be reduced, and the high efficiency of the storage system can thus be ensured.

3.2 The Implementation of Grid GIS Data Service

In the grid environment, the grid nodes, organized by each virtual SAN, provide GIS users a basic platform to use spatial data sharing. When users submit their requests through data grid, the grid GIS data storage system would response them through a simple data service workflow shown in Figure 3.

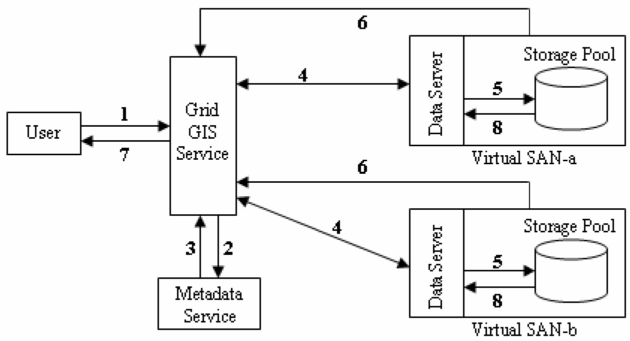


Fig. 3. The realizing workflow of spatial data server

(1) As step-1 in the above workflow, a client sends a data request to grid GIS service when a user has a request of spatial data. This client can be a user host computer, a sub node that is connected with grid GIS.

(2) Grid GIS components send an inquiry request of data access to metadata service, shown as step-2. Then metadata service collects the information of data servers and searches an object server to meet the user's request. The searched result will then be sent back to grid GIS service (step-3).

(3) According to the searched result in step-3, grid GIS service submits the data request to virtual SAN-a and SAN-b in step-4. In this step, grid service is needed to get the identification and authorization of access data server.

(4) In Step-5, data service starts spatial metadata service to search the spatial data stored in virtual SAN according to the data request. After the data block is found, it authorizes the data server to communicate with the user with the availability mechanism, such as availability level and time limitation.

(5) The authorization block is sent to host computer through the data channel in step-6, and then grid GIS service sends it to the user through user interface in step-7.

(6) After the completion of its mission, the storage device sends a report to data server and its authorization is then abolished.

It is a simple workflow of data request service. Users only need to connect the grid GIS in order to submit their requests for service and then obtain the authorization. Through the storage and retrieval of grid GIS, the process of assessing the data of each data centre is exposable for users. A virtual SAN environment has been successfully built and the corresponding workflow has perfectly simulated in this paper.

3.3 Design of Spatial Data Server

Spatial data management is a complicated job, and in this systematic structure, spatial data server is responsible for the management and maintenance of the whole virtual SAN, so the design of data server is particularly important. Spatial data server should finish the combination of local spatial data resources in order to provide a unified and easily utilized interface for grid data service; hence, it would influence the precision and accuracy of data service. Meanwhile, the ability of the data server directly influences the service quality of data-providing. A hierarchical structure of spatial data server is shown in Figure 4.

Grid service tier provides the correlative service of data grid. In the grid service, metadata service mainly provides the related information of other data servers, which may either locate in the same or in different virtual SAN. The metadata is the abstract and general of spatial metadata layer in all data servers. The information, such as managed data range and data layer, among data servers can be acquainted through the metadata server. Based on it, each virtual SAN can together provide data services to users. It is essential in spatial data management that spatial metadata layer describes the spatial data stored in virtual SAN. In spatial data management, metadata can avoid the redundant data storage and build a highly efficient data access mechanism to reduce the data-obtained time. Thus, it must not be lacked in spatial data management. Spatial data management tier provides the essential functions of spatial logical and mathematical processing that can analyze, process and manage the spatial data according to real demands from users. Further storage management tier directly manages the final storage resources, from communicating with the storage device and providing the relative services of storage management, such as data coherence and data restoration. The virtualization module, which is in storage management tier, is coordinated with the applicably programmed virtualized module to implement the virtualization of SAN. In the virtualization course, it provides the storage configuration and controls the information.

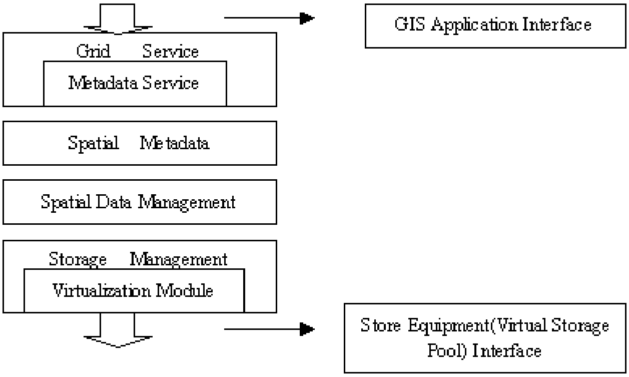
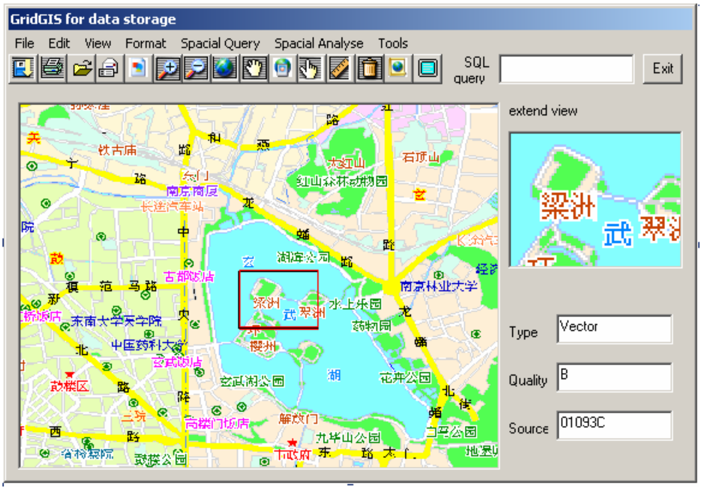


Fig. 4. The hierarchy of spatial data server

4 Experiments

In the system, the spatial data server is a servers' cluster composed of a file server, a database server and a virtualization engine. The servers, all of which are constructed with the virtual SAN, in the cluster connect with the storage equipments through a switch. Based on the above idea, a grid GIS data storage and management system has been developed with B/S model in the heterogeneous LAN. The essential framework of grid GIS is constructed by Globus3.2. Some basic characteristics and functions of grid GIS have been realized by using the technology of Java2 in the system, including the data and host computer registration service to users in the integrated storing and managing environment, centralization of the management of metadata information, storage of spatial data in the unifying form, parallel access of spatial data for multi-user and dynamic conformably monitor of the whole procedure, auto response of the user's request, zoom in/out of vector data in any scales, and other basic data operations in clients. Figure 5 shows an example of reading a vector map in a client browser.

In the trial, a client obtained the data access through the 100MB LAN from the storage system. The capacity of the performance of the server was not reduced distinctly with the increase of accessing users when the data were less than 1MB. The average response time of storage server was linear with the data size. Its peak value was about 33.86ms. The response time increased obviously with the continuous increase of data size more than 1MB. However the upper I/O speed was still maintained. When the data increased to 200MB, the saturated I/O speed of storage server was 76/28Mbps. According to the experiment, the virtual SAN with double thoroughfare mechanism can satisfy the high throughput demand of spatial data access. Shielded the concrete store environment and resource by grid platform and virtual SAN, user can get instant, dynamic spatial data from client host, which can be zoomed, rotated, written and done many other operation conveniently.



Type: Vector

Quality: B

Source: 01093C

Fig. 5. Reading a vector map from client in grid GIS

5 Conclusions

In recent years, the research of grid GIS has become an interest for researchers in many fields. It is a necessity to solve the spatial data storage problem based on a grid environment. Thus, it is, in the real world, necessary of building a grid GIS storage system based on double thoroughfare virtual SAN, which can conceal the physical differences of storage devices and provide for GIS users a uniform data map to implement the exposable access and open interoperability of spatial data.

References

1. Zhou Xing: Research on the Storage & Management of Geographic Data, Science of Surveying and Mapping, 27(4), (2002) 49-51.
2. Zhang Ling, Jiang Dong-xing, Liu Qi-xin, Zhou Lin, Shen Pei-hua: Research and Implementation of Distributed Grid Storage Manage System. Computer Science, 30(6), (2003) 74-77
3. Jin Hai, Ran Longbo, Wang Zhiping, Huang Chen, Chen Yong, Zhou Runsong, Jia Yongjie: Architecture design of global distributed storage system for data grid. High Technology Letters. .9(4), (2003) 1-4
4. Shen Zhanfeng, Luo Jiancheng, Zhou Chenghu, Cai Shaohua, Zheng Jiang, Chen Qiuxiao, Ming Dongping, Sun Qinghui: Architecture design of grid GIS and its applications on image processing based on LAN. Information Sciences, 166(1-4), (2004) 1-17

5. Anon: Network data storage. Civil Engineers Australia. 74(5), (2002) 32
6. Guo Yu-feng, Li Qiong, Liu Guang-ming, Liu Heng-zhu: Research on Storage Virtualization. Application Research of Computers. 21(2), (2004) 56-57, 60
7. Xie Chang-sheng, Gao Wei: Study and Implementation of SAN Virtualization. Application Research of Computers. 20(8), (2003) 130-132
8. Michael Simonyi: Storage area networks and data management. Auerbach Publications© CRC Press LLC. (2002)

A BP Neural Network Predictor Model for Desulfurizing Molten Iron

Zhijun Rong, Binbin Dan, and Jiangang Yi

Department of Industrial Engineering,
Wuhan University of Science and Technology,
430081 Wuhan China
rongzhijun@263.net

Abstract. Desulfurization of molten iron is one of the stages of steel production process. A back-propagation (BP) artificial neural network (ANN) model is developed to predict the operation parameters for desulfurization process in this paper. The primary objective of the BP neural network predictor model is to assign the operation parameters on the basis of intelligent algorithm instead of the experience of operators. This paper presents a mathematical model and development methodology for predicting the three main operation parameters and optimizing the consumption of desulfurizer. Furthermore, a software package is developed based on this BP ANN predictor model. Finally, the feasibility of using neural networks to model the complex relationship between the parameters is been investigated.

1 Introduction

Increasingly stringent quality requirements have imposed the demand for steels with very low levels of impurities such as phosphorus, sulfur, hydrogen, nitrogen, and oxygen, and of nonmetallic inclusions such as MnS, SiO_2, and Al_2O_3. Sulfur is a hazardous ingredient for most types of steels. It has impact on quality and mechanical properties of steel and causes risk of damaging the converter and crack the slab. It is necessary to desulfurize molten iron at different stages of the production process because the hot metal is usually required to contain less than 0.02% sulfur even for normal plain steel. There are different methods of desulfurizing molten iron in the production line, the main methods being the following: desulfurization of molten iron in the ladle with the use of a submersible lance; the ISID method; the Slide Gate method; the KR method; injection of magnesium through a lance with a vaporizer [1]. The KR method that is also called the mechanical stirring method has been successfully used in China for several years now. When desulfurization is done using a stir bar, it is possible to introduce the desulfurizer deep into the molten iron. This leads to fluid interphase contact. The torque and the rotational speed of stir bar are high so that desulfurizer can undergo chemical reaction completely. There are some important operation parameters including the amount of desulfurizer, stirring speed and stirring time by using the KR method. The optimum parameters can ensure high quality of

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 728 – 735, 2005.

final steel product, long service life of manufacturing device and low production cost. However, the operation parameters are assigned according to the experience of operators in many steel plants that implies a high degree of randomicity in the production process. Some of the recent research findings of the current authors in this direction have been highly encouraging [1-2,7]. Artificial Neural Networks are revolutionary computing paradigms that are especially useful to address problems where solutions are not clearly formulated or where the relationships between inputs and outputs are not known. The development of ANN provides new strategy for knowledge acquisition and process control in metallurgy. Compared with traditional models, ANN model achieves fault tolerance capability, self-learning ability, adaptivity and nonlinear mapping through the weighted summation process itself, which is an intrinsic characteristic of it. ANN modeling techniques in metallurgy can be effectively used to develop models to analyze and predict the operation parameters in the production process though the process is complex and hard to be quantized for the boundary condition is not definite. Some research work has used various ANN modeling techniques in metallurgy including ANN prediction models for mechanical properties of materials [3-4] and process of steel making [5-6].

A Neural network predictor model for KR method of desulfurizing molten iron is developed in this paper. The remainder of this paper is organized as follows. In Section 2, the BP neural network model is presented. In Section 3, methods used for developing the model are introduced. In Section 4, run results of the model are illustrated. Finally, in Section 5, the work is summarized and some future directions are highlighted.

2 Proposed BP Neural Network Model

The process of desulfurizing molten iron is shown in Fig 1. As the molten iron from the blast furnace is prepared in the ladle, the stir bar is directly inserted into molten iron into which certain amount of desulfurizer is fed. Desulfurizer reacts with sulfur dissolved in molten iron to form solid reaction product on the molten iron surface. The solid reaction product, being part of slag, will be removed by machine. Reducing the consumption of desulfurizer is very important to save the production cost. The consumption of desulfurizer has been associated with size, temperature and composition of molten iron. It is difficult to get the direct mathematical relation between these parameters. The predictor model is established to analyze the relations between parameters. Size, temperature and composition of molten iron are inputs of the model while amount of desulfurizer, stirring speed and stirring time are outputs of it. This BP neural network model can compute and assign the operation parameters intelligently. The operation parameters are no longer assigned according to experience of operators. This mathematical model can evolve because data gathered from the job site can be input to it directly and used to be training samples. The network is not just given reinforcement for how it is doing on a task. Information about errors is also filtered back through the system and is used to adjust the connection weights between the layers, thus improving performance. BP ANN has been widely used in engineering field and its algorithm is simpler to implement, so BP neural network modeling techniques are used in this research work.

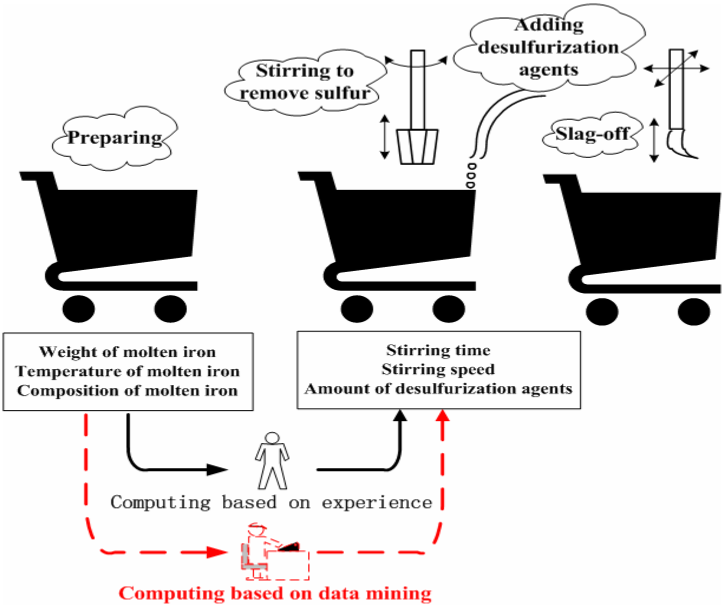

Preparing
Stirring to remove sulfur
Adding desulfurization agents
Slag-off
Weight of molten iron
Temperature of molten iron
Composition of molten iron
Stirring time
Stirring speed
Amount of desulfurization agents
Computing based on experience
Computing based on data mining

Fig. 1. Process of desuflurizing molten iron

The neural networks are trained using actual operation records. BP training adapts a gradient descent approach of adjusting the ANN weights. During training, an ANN is presented with the data of thousands of time (called cycle). After each cycle, the error between the ANN outputs and the actual outputs are propagated backward to adjust the weights in a manner that is mathematically guaranteed to converge to the training technique [8]. The number of output neurons and input neurons can be designed according to the actual requirement. The number of output/input neurons is same as the number of inputs/outputs to the physical model. To reduce the complexity and learning time, the small system should be introduced. Since the three-layer structure model can be used to simulate the random system that has n inputs and m outputs, only one hidden layer is used to simplify the network structure in this paper. Choosing the number of hidden neurons can be complex. In this paper, we choose it according to equation:

$$n_1 = a + \sqrt{n + m} \tag{1}$$

where n_1 is the number of hidden neurons, n is the number of input neurons, m is the number of output neurons, a is constant number between 1 and 10. The number of hidden neurons can be changed to a suitable value during the training. The past production data are used as the inputs to this model. They are required to be normalized.

3 Implementation

A software package has been developed on the basis of this BP neural network predictor model. This software can be set up in the working field to assist the actual production. The software architecture is shown in Fig.2. User inputs the original parameters (size, temperature and composition of molten iron) through the user interface. All the original parameters will be stored in the database for the model training. The supporting database is designed to improve the self -learning ability of the model and manage the original data to simulate the experience of operators in steel plant. Besides being used as data source for the ANN system, this database can be manipulated by the system users connecting the human intelligence with the computer intelligence. Computer drives the program to connect data source and operates data by SQL (structured query language). After being prepared, the data will be analyzed and processed by computing algorithm. The supervised network will train and adjust weights for the predictor model so that the optimized outputs including the amount of desulfurizer, stirring speed and stirring time can be achieved. This software program developed on Windows98/Windows2000/WindowsXP operation system is used to build the model structure. It is produced as a library of procedures written in the VC++ language and is equipped with an interface based on virtual instrument technology.

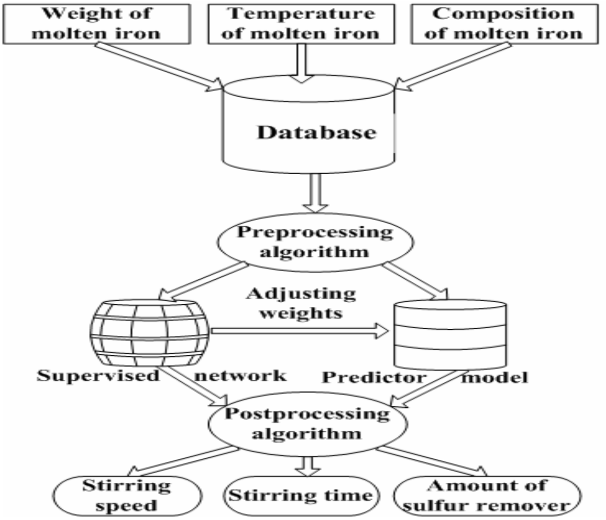
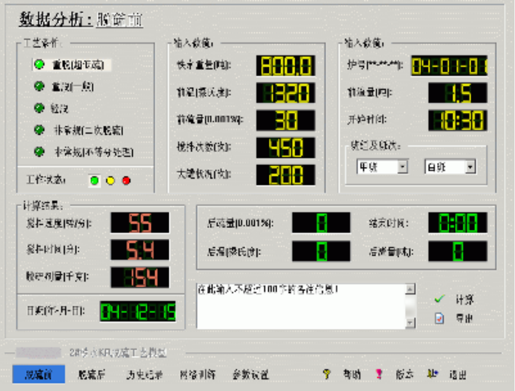


Fig. 2. Predictor model software architecture

The computing algorithm applied is BP neural network algorithm. The choice of activation function can change the behavior of the ANN network considerably. We adopt the sigmoid activation function as the activation function. The model has six input neurons and three output neurons. The number of hidden neurons can be achieved by training the model. The model has a three-layer architecture with only one hidden layer. The maximum training cycle is 10000 and the expectation accuracy is 1e-005.The training results will be transferred to TXT format file. Users input the required information about molten iron including size, sulfur content and temperature through user interface as Fig3 shown. The additional parameters about production device such as operation cycles of stir bar and pig iron ladle are also needed. To obtain the detailed production record, operation time, amount of slag and heat number are also provided. Normal user can also print and check the production record. Thereafter, different production processes including partial desulfuration, complete desulfuration, normal desulfuration and secondary desulfuration are provided for users to choose. When clicking the "computing " button, the predicted values including amount of desulfurizer, stirring time and stirring speed will be shown to the left lower blanks. They can be used to analyze the desulfurization process. After computing, all the data will be stored in database and used to train the model.



数据分析：脱硫前

工艺条件：
- ● 重脱(超低硫)
- ● 重深(一般)
- ● 轻深
- ● 非常规(二次脱硫)
- ● 非常规(不等分处理)

工作状态： ● ● ●

输入数值：

铁水重量(吨):	800.0
前温(摄氏度):	1320
前硫量(0.001%):	30
搅拌次数(次):	450
大罐状况(次):	200

输入数值：

炉号(**-**-**):	04-01-0
前渣量(吨):	1.5
开炉时间:	10:30

班组及班次： 甲班 ▾ 白班 ▾

计算结果：

搅拌速度(转/分):	55
搅拌时间(分):	5.4
脱硫剂量(千克):	154
日期(年-月-日):	04-12-15

后硫量(0.001%):	0	结束时间:	0:00
后温(摄氏度):	0	后渣量(吨):	0

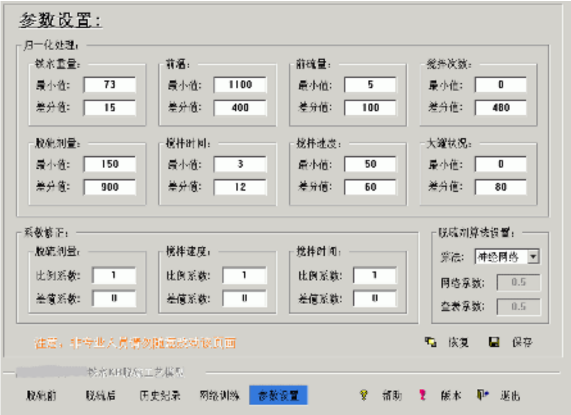
在此输入不超过100字的备注信息！

✓ 计算
导出

[redacted] 2#号KR脱硫工艺模型

Fig. 3. User interface

Authorized users can set parameters for neural network model and train the model as Fig 4,5 shown. The minimum value and difference value of parameters are set for being normalized. If the predicted values are not satisfactory, the proportional



参数设置：

归一化处理：

	最小值:	差分值:
铁水重量:	73	15
前温:	1100	400
前硫量:	5	100
搅拌次数:	0	480
脱硫剂量:	150	900
搅拌时间:	3	12
搅拌速度:	50	60
大罐状况:	0	80

系数修正：

	比例系数:	差值系数:
脱硫剂量:	1	0
搅拌速度:	1	0
搅拌时间:	1	0

脱硫剂算法设置：

算法： 神经网络 ▾

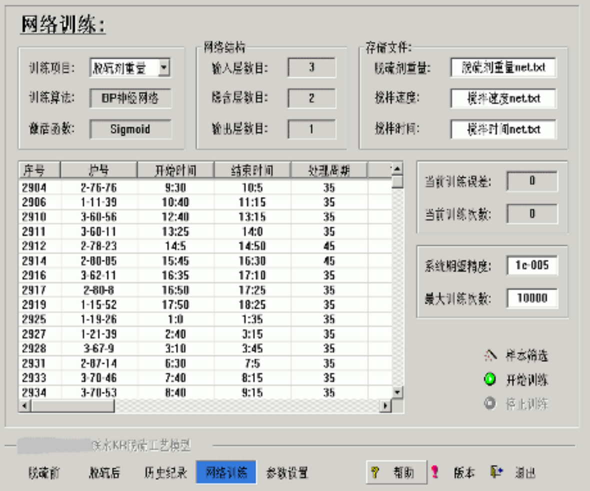
网络系数： 0.5

经验系数： 0.5

注意：非专业人员请勿随意改动该页面

[redacted] 铁水KR脱硫工艺模型

Fig. 4. Setting parameters for neural network model



网络训练：

训练项目：脱硫剂重量

训练算法：BP神经网络

激活函数：Sigmoid

网络结构

输入层数目：3

隐含层数目：2

输出层数目：1

存储文件：

脱硫剂重量：脱硫剂重量net.txt

搅拌速度：搅拌速度net.txt

搅拌时间：搅拌时间net.txt

序号	炉号	开始时间	结束时间	处理周期
2904	2-76-76	9:30	10:5	35
2906	1-11-39	10:40	11:15	35
2910	3-60-56	12:40	13:15	35
2911	3-60-11	13:25	14:0	35
2912	2-78-23	14:5	14:50	45
2914	2-00-05	15:45	16:30	45
2916	3-62-11	16:35	17:10	35
2917	2-80-8	16:50	17:25	35
2919	1-15-52	17:50	18:25	35
2925	1-19-26	1:0	1:35	35
2927	1-21-39	2:40	3:15	35
2928	3-67-9	3:10	3:45	35
2931	2-87-14	6:30	7:5	35
2933	3-70-46	7:40	8:15	35
2934	3-70-53	8:40	9:15	35

当前训练误差：0

当前训练次数：0

系统期望精度：1e-005

最大训练次数：10000

Fig. 5. Training the neural network

coefficient and difference coefficient will be modified to improve prediction accuracy. As Fig 5 shown, the model is trained using error back-propagation and minimizing the root mean square error between the teaching sequence of parameters and the predicted parameters sequence.

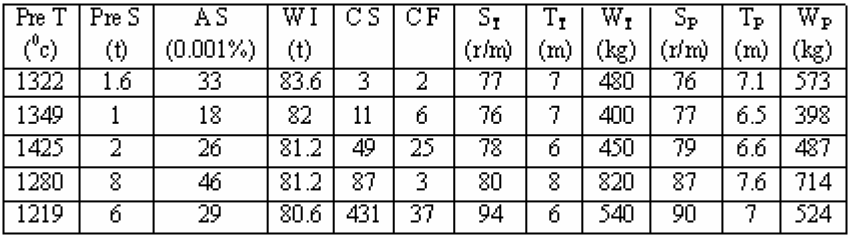
4 Practical Application and Run Results

This BP ANN predictor software has been applied to a steel plant in which the three key control parameters were computed and assigned on the basis of the experiences of operators during the desulfurizing process before. These parameters are so subjective that low efficiency and high cost can be obtained in steel production. Now the control parameters are computed by neural network model, so the loss caused by human factor will be reduced. The past operation parameters were chosen to train the model. If the output errors of the model exceed the permitted range, the operation record can be used as inputs to train the system by adjusting the weights. Therefore, the model can evolve and the accuracy will be improved. Once optimum weights are reached, the weights and biased values encode the network's state of knowledge. ANN training model input/output and ANN predicted output in Table 1. are essential for evaluation of the model.

Pre T: temperature of molten iron before desulfurization (actual operation data used as input)
Pre S: amount of slag before desulfurization (actual operation data used as input)
A S: content of sulfur in molten iron (actual operation data used as input)
WI: size of molten iron (actual operation data used as input)
CS: operation cycles of stir bar (actual operation data used as input)
CF: operation cycles of the pig-iron ladle (actual operation data used as input)

S_T: rotational speed of stir bar (actual operation data used as output)
T_T: time of stirring (actual operation data used as output)
W_T: amount of desulfurizer (actual operation parameters used as output)
S_P: ANN predicted rotational speed of stir bar
T_P: ANN predicted time of stirring
W_P: ANN predicted amount of desulfurizer

Table 1. ANN training Model input/output and ANN predicted output



Pre T (^{0}c)	Pre S (t)	A S (0.001%)	W I (t)	C S	C F	S_T (r/m)	T_T (m)	W_T (kg)	S_P (r/m)	T_P (m)	W_P (kg)
1322	1.6	33	83.6	3	2	77	7	480	76	7.1	573
1349	1	18	82	11	6	76	7	400	77	6.5	398
1425	2	26	81.2	49	25	78	6	450	79	6.6	487
1280	8	46	81.2	87	3	80	8	820	87	7.6	714
1219	6	29	80.6	431	37	94	6	540	90	7	524

The neural network used for the presented model demonstrates good agreement between the actual and predicted value. Especially the predicted value of stirring speed and time is close to the actual operation value. The error between the predicted amount of desulfurizer and the actual amount of desulfurizer is 10%. The sample data was chosen at random. We can find that the values of W_P are almost all less than those of W_T except the first one. It shows that actual desulfurizer level is higher than theoretical level. Though the first actual desulfurizer level is lower than theoretical, the operation records proved it should be increased. According to the investigation in this steel plant, these actual operation values are acquired on the basis of the operators experience even personal tastes and cannot be controlled accurately in the desulfurization process. The desulfurizer level may be different though production task remains same. Since this BP neural networks predictor model is used in this plant, the desulfurizer level is under efficient control. Furthermore, the desulfurization process can be optimized. The preceding discussion shows two main advantages of the model: one is to assist the operation by providing the operation parameters including amount of desulfurizer, time and speed of stirring; the other is to assist the research on relation between input parameters and output parameters of this predictor model. It has been observed that three inputs have strong relation with the amount of desulfurizer when the inputs are changed respectively. These three inputs are: size of molten iron, content of sulfur in molten iron and the temperature of molten iron. Desulfurization is also found essentially unaffected by operation cycles of stir bar and ladle.

5 Conclusions

The research work described in this paper demonstrates that development of neural network techniques for the prediction of operation parameters in hot metal desulfurization is now within the realms of possibility. This predictor model based on ANN demonstrates its validity with the application results. The results obtained from this predictor model can be suitably applied to desulfurization of molten iron. The operation parameters are no longer estimated subjectively but computed based on mathematical model. It has been observed that the nonlinear relation between the parameters can be analyzed through this model. Furthermore, based on the prediction model, it can be stated that the optimum consumption of desulfurizer can be obtained. It is important that future research effort should be directed toward fully clarifying the interrelation between desulfurization parameters and achieving knowledge to improve production performance.

References

1. Yugov, P.I., Romberg, A.L.: Improving the quality of pig iron and steel. Metallurgist. Volume 47(2003) 62 – 65.
2. Yugov, P.I., Romberg, A.L. , Yang , D.: Desulfurization of pig iron and steel. Metallurgist. Vol.44(2000) 11–12.
3. Haque, M. E. , Sudhakar, K.V. : ANN back-propagation prediction model for fracture toughness in microalloy steel. Int. J. Fatigue. 24 (2002)1003-1010.
4. Haque M.E., Sudhakar K.V . : ANN based prediction model for fatigue crack growth in DP steel. Int J Fatigue Fract Eng Mater Struct 2001; 24(1): 63-8.
5. Wu, M. , Nakano, M. , She, J. : A model-based expert control strategy using neural network for the coal blending process in an iron and steel plant. Expert system with applications 16 (1999) 271-181.
6. Schlanga,Martin. , Langb,B. , Poppeb,T. , Runklerb,T. , Weinzierlc,K. : Current and future development in neural computation in steel processing. Control engineering practice 9(2001) 975-986.
7. Dyudkin, D. A. , Grinberg,S. E. , Marintsev, S. N. : Mechanism of the desulfurization of pig iron by granulated magnesium. Metallurgist. Vol. 45(2001).
8. D, Rumelhart., G, Hinton., R, Williams. : Parallel distributed processing. Cambridge (MA): MIT Press, 1986.

A Flexible Report Architecture Based on Association Rules Mining

Qiping Hu

International School of Software, Wuhan University, 430072 P. R. China
huqp@whu.edu.cn

Abstract. This paper proposes flexible report architecture based on association rules data mining. A three-layer architecture is proposed namely, origin-data layer, data-processing layer, and format layer. These three layers are linked by a data variant tree in a power information management system. Users can modify report format as well as data whenever needed. In the origin-data layer data warehouse is used to provide data from multiple databases. In the data-processing layer, on-line analytical processing (OLAP) and association rules are used to enhance the template-making for reports. A smart solution to the problem of fixed report templates is provided and information in a power information management system can be shared. In some sense it can be an all-purpose tool to generate reports with great flexibility.

1 Introduction

The technology of power information management systems is developing rapidly. Its information may reflect the data of production and running, equipment and check standard changes frequently, and the report from multiple information systems should change accordingly in the format and content to meet user requirements. At present, a majority of reports in power information management systems are mainly generated by hand with report tools such as Excel, or derived from a single system. All these reports have some common disadvantages. First, people have to spend much time and energy on them, and the work of generating reports is too onerous. Second, although some management software of power system, for example MIS, has offered some report templates, and these templates are generally limited to some few patterns when the development of software is finished. These patterns can't reflect the changing demands of users, which leads to a short useful life and much maintenance work. Third, the data in a report can only come from an information system, and that limits the function of report. Fourth, the data of an associative analytical data can not be embedded in a report. A general user-oriented flexible report system built on data mining technology and associative analysis is introduced in this paper. The report system is analyzed based on three-layer model. The data and format of the report are dynamic and alterable [1]. This solution is for power supply data warehouse and associative analysis of those data. So the problem of conflicts between the fixed format and changing demands from users is solved. At the same time the coherence and integration of power system's data is realized.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 736 – 743, 2005.

2 Principle of Flexible Report

2.1 Design Idea

The report architecture is described as a three-layer model (Fig.1), i.e. origin-data layer, data-processing layer and format layer.

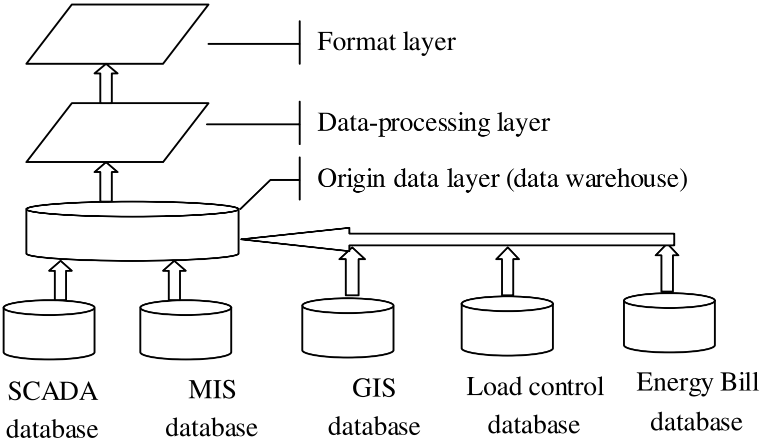


Fig. 1. The Three Layers of Flexible Report

In Fig.1, the origin-data layer is a data warehouse, and the arrow indicates the flow of data. The format of report is separated from its data, which makes the data variable, the format fictile and the processing of data modularized. By constructing the data variant tree (illustrated in chapter 2.2.4) of power information system, the three layers are linked together. The report maker designs the form title, headline of the report and so on in the format layer. All the data of a report are set as variants and its property is defined in the data-processing layer. Data is got from origin-data layer with the variant's information via the data variant tree. After the designing work of report, we save its format and data-processing model. So, if the format layer is viewed as background, the whole report can be regarded as one data layer which is plastered to the background layer perfectly [2]. When running, the format and data-process model of the report can change accordingly to the demands of users. At the same time current data, the results of associative analysis, character, chart or analytical curse can be displayed in the report. The structure of the flexible report is shown as Fig.2.

If a report is analyzed with the object-oriented opinion, a report can be viewed as a formatted text. Generally speaking, its contents are composed of format and data. The object of data can be defined as

```
REPORT=<REPORTNAME, REPORTMODE, DATASOURCE>
```

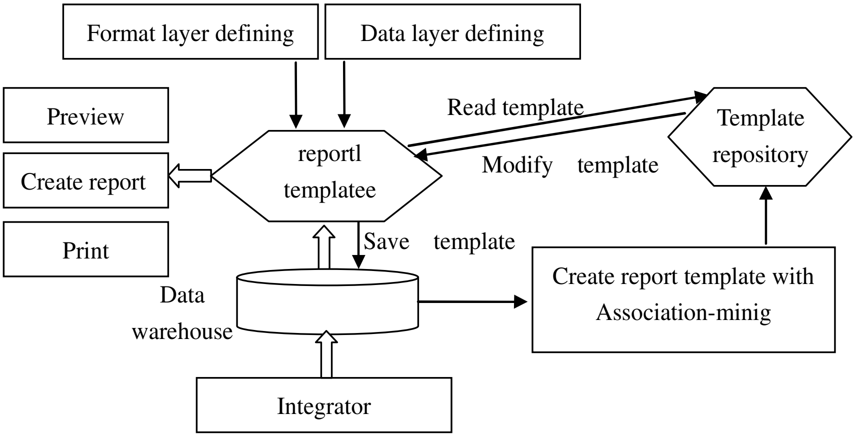


Fig. 2. The structure of flexible report

The REPORTNAME represents the name of report. The REPORTMODE represents the property of format. All kinds of text, frame or picture are its sub-object. The DATASOURCE represents the data source. It is a muster of data items. If permitting users to modify the property of REPORTMOD and DATASOURCE when needed in use, the appearance and data of report can be actively changed, and it is the theory basis on which we describe report with three-layer model.

2.2 The Realization of the Flexible Report

2.2.1 Origin-Data Layer

The origin-data layer corresponds to the data warehouse of a power system. In the electric power system there are several databases. SCADA databases store the real time data. MIS System manages the documental data of equipments, experimental data, and business processes. GIS System manages the facilities relative to their geographic locations. The volume of GIS can be over 10GB. Load Control database which contains the large customer information and their load consumption. The data of a middle scale power supply company may be over 600,000 records a day. Energy database stores the electricity charges data and all customer information of the power supply company. Any customer can retrieves original data for the last three months or six months.

The data are only provided for the respective system. The data of one system cannot be applied by another system and cannot be shared by others. These systems were built for different usages and in different periods. The systems mentioned above can not share the data with the others. There are data the redundancy among the systems. We may take the most efforts to integrate them. The report method described in this paper can integrate many systems as a subsystem of a united system. We apply the data warehouse in associating, consistent and extracting from the database mentioned above.

A data warehouse is an integrated repository that stores information which may originate from multiple, possibly heterogeneous operational or legacy data sources.

There are two approaches to creating the warehouse data, namely, bottom up approach and top-down approach, respectively. The bottom-up approach is used in data warehousing because user queries can be answered immediately and data analysis can be done efficiently since data will always be available in the warehouse. Hence, this approach is feasible and improves the performance of the system. In our project, we use an approach that is hybrid approach by integrator which combines aspects of the bottom-up and top-down approaches. In this approach, some data is stored in a warehouse, and other data can be obtained from the primary sources on demand. When the information and the structure of database are not clear to users, we use translate-table. The translate-table contains the informational data about the creation, management, and usage of the data warehouse. It serves as a bridge between the users of the warehouse and the data contained in it.

2.2.2 Format Layer

The format layer is the background of a report. It determines the report's appearance. If we start our work from drawing tab line, it is a long time and hard task for us. We have to find other simple means.

We recur to the ActiveX technology to solve the format problem. ActiveX is an application technology based on the COM. The purpose of ActiveX is to encapsulate the program segments probably used frequently in programming and provide general interface. The programmer can use the ActiveX Control conveniently in different development platform and realize software's reuse. This advantage makes it easy to deploy them on the web and finally the flexible report becomes feasible based on the Browser/Server mode. Taking the worksheet of that ActiveX control as the background layer, we can meet most of the format demands from users by fully using of the powerful functions in which the Formula One offers for table edit.

2.2.3 Data-Processing Layer

The data processing layer is the kernel of flexible report. It is also the most difficult part in the three layers. The warehouse data is accessed by OLAP server to present the same in a multidimensional way to the front end tools for analysis and informational purposes. Basically, OLAP server interprets client queries and converts them into complex SQL queries required to access the warehouse data. It might also access the data from the primary sources if the client's queries need operational data. Finally, the OLAP server passes the multidimensional views of data to the front end tools, and these tools format the data according to the client's requirements. The report is implemented based on OLAP [3].

In fact getting the data we need from the database is the procedure that the database engine executes the SQL sentence. The definition of variant's property is actually to ascertain the data field in the table of database and the query condition. After analyzing most reports of power system, we found that the changes of report in a same series only occurs at one or two data fields' query conditions. For instance, the monthly report of active power in a substation only changes the value of month when searching data from the database. So in the flexible report we make all the data in the data-processing layer variable, and classify them to three kinds of variants. The first kind is public variants. Its query condition can be changed whenever needed and will affect all of the data got from the database. All the public variants' query conditions constitute the alterable query condition--VARSQL. The second kind is self-defined variants. Each self-defined variant corresponds to the data eventually shown in the cell of report. Their query conditions are ascertained in the data-processing layer and finally the fixed query condition-- FIXEDSQL is formed. The third kind is relation variant. The relation variant represents the relations between different tables in the database when selecting data from multi-tables. Its query condition is RELATION. When we start creating report, each variant's eventual query condition self-defined is:

```
DATAXSQL=VARSQL+FIEXEDSQL+RELATION
```

Then the flexible system will generate corresponding SQL sentence according to DATAXSQL and with the SQL sentence the database engine will search for all the data needed in the database. In the flexible report system all of the SQL sentences are written in the standard SQL language.

In the data-processing layer, we should solve the problem of combining and processing information. The data in the report is not only from one field of the table but the result of processing with many fields. It is necessary to let users define the processing mode easily and save it in the data-processing template. In the flexible report system, many data processing formulas and statistic functions are provided. With them users can easily define the variants' processing mode with the help of data variant tree.

2.2.4 Create Report Template with Association-Mining

Because making report template is hard work and people often want to know which data are associative and what data should be displayed in a report. We apply the association approaches from data warehouse discipline. The templates defined by users are stored in database. And for every template we associate it with descriptive information. We mine the associative information, and the system can accordingly to the mining results produces templates and then display the results [4].

The problem of mining association rules in a table can be defined as follows: Let $I = \{i_1,.., i_n\}$ be a set of literals called items. A set $X \subseteq I$ is called an *itemset.* An itemset X with k elements is called a *k-itemset***.** Let R be a table with transactions t involving elements that are subsets of I. A transaction t *supports* an itemset X if $X \subseteq t$. The *support* of an itemset X is the ratio between the number of transactions of R that support the itemset X and the total number of transactions of R. An itemset X is called a *frequent itemset* if its support value is greater than or equal to the minimum support specified by the user. An *association rul*e is an expression of the form $X \to Y$, where X and Y are itemsets; the rule's *support* is the ratio between the number of transactions of R that contain X and Y and the total number of transactions of R. *Confidence* is the ratio between the number of transactions containing X and Y and the number of transactions containing X. A well-known example of an association rule involving market basket data is: "70% of the purchases that contain diapers also contain beer and 4% of all purchases contain these two items". In this example, 70% is the rule's *confidence* and 4% is the rule's *support.* The mining of association rules consists of finding association rules that satisfy the restrictions of minimum *support* and *confidence* specified by the user. These rules are called *strong rules.* The problem described above is also known as the problem of mining Boolean association rules because it involves mining categorical data.

We can give *confidence and support* levels to mine the saved templates in data warehouse and create templates to display data. Of course, you can modify the templates if you are not satisfied with the mining results or their format.

2.2.5 Data Variant Tree

The data variant tree is the ligament of those three layers of the flexible report. It's also the bridge through which users communicate with the database[5]. As we all know, because of the difference of different database designers and naming criterions, users don't know the specific meanings of every field without the data dictionary. So in the flexible report system the mapping tables are created. In the mapping table corresponding relations are saved and mapping tables act as a translator between the uses and database. The mapping tables include field-mapping table and table-mapping table. The field-mapping table and table-mapping table can be defined as:

```
FIELDMAPPING= {FCHNAME, FENNAME, TBNAME, DBNAME}
TABLEMAPPING={TBCHNAME, TBENNAME, DBNAME}
```

Here, FCHNAME is Chinese meanings of the field, FENNAME is English name of the field, and DBNAME is English name of database in which the table exists. TBCHNAME is Chinese meanings of the table, TBENNAME is English name of the table, and DBNAME is English name of database in which the table exists. On initializing, we assort different tables according to their DBNAME and add them to database node they belong to. At the same time we assort different fields according to their TBNAME and DBNAME, and add them to corresponding table node they belong to.

2.2.6 Saving and Modifying the Report Template

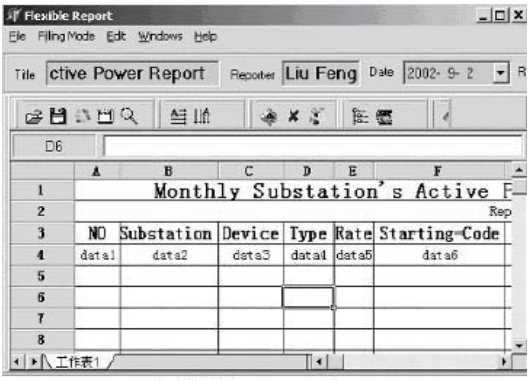
The function of report template is to alleviate the load of workers. A report template can be created and saved if reports are needed to be made repeatedly and these reports have almost the same format and content. Actually a report template can be regarded as a small custom-made program suited to a series of similar reports. In the format-layer the "save" function of Formula One is used to save the template's format information. As to the data-processing layer, its information is saved with linked list and file of record type. In the linked list, each node means a variant of this layer.

In the saving procedure of the report template, the linked list is converted to record type files. Reversely the record type files are converted into linked list when the template is opened. To add a variant to the report template or delete a variant from the report template we only need to deal with a node accordingly in the linked list. Users can modify the template on the foundation of the early one at any moment. So the shortcomings of fixed report template are overcome and reports become flexible.

3 An Application Case Study

We take the substation's active power monthly report as an example to illustrate the procedure of making a report. Firstly, start the application program of flexible report to make a report template. In the user interface, input the title name, date, text and other essential information. At the same time adjust the format to the satisfied effect..

Secondly, open the data variant's property dialog and select the source table. Then define the public data variants and their query condition. At this step the relation variant can be defined if needed. Thirdly, define the self-defined variants. In the property page we can input the variant's name, displaying position, processing mode and fixed query condition. These steps can be repeated if other self-defined variants need to be added. Finally, save the report as a template when all the designing work is finished. Fig.3 shows the process of making report template.



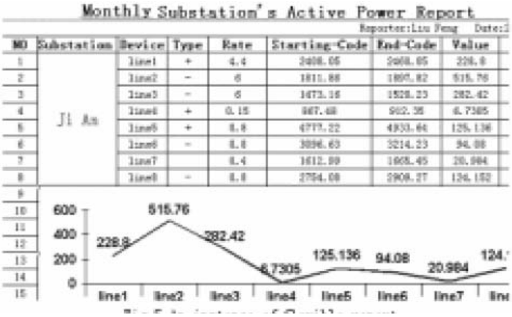
Title: ctive Power Report Reporter: Liu Feng Date: 2002- 9- 2

D6

	A	B	C	D	E	F
1	Monthly Substation's Active F					
2						Rep
3	NO	Substation	Device	Type	Rate	Starting-Code
4	data1	data2	data3	data4	data5	data6
5						
6						
7						
8						

Fig. 3. Making Report Template

The 'data1','data2' and so on are the self-defined data variants. Then we click the button of Generating Report, thus the flexible report system will be connected the database, all the data needed will be searched for and displayed in the corresponding position of the report at last. A finished report is shown in Fig.4. The data and analytical curse can be shown in one figure simultaneously.


Monthly Substation's Active Power Report

Reporter:Liu Feng Date:

NO	Substation	Device	Type	Rate	Starting-Code	End-Code	Value
1	Ji An	line1	+	4.4	2408.05	2460.05	228.8
2		line2	-	6	1811.88	1897.84	515.76
3		line3	-	6	1473.16	1520.23	282.42
4		line4	+	0.15	867.48	912.35	6.7305
5		line5	+	0.8	4777.22	4933.64	125.136
6		line6	-	0.8	3096.63	3214.23	94.08
7		line7		0.4	1612.99	1665.45	20.984
8		line8	-	0.8	2754.08	2909.27	124.152

Fig. 4. An instance of flexible report

4 Conclusions

The flexible report system discussed in this paper has been applied with a very good running state in Power Supply Bureau of Ji-An, Jiang-Xi province, China. It matches most of the demands from users and significantly saves time and resources. The idea of separating report's format from data makes the report flexible and fictile. The integrated data source constructed by data variant tree breaks the enclosure of different subsystems and achieves the goal of sharing information. We can also extend the function of flexible report to drawing analysis curse, supporting decision-making and so on. In fact the flexible report system can be used as an integrated query tool in the power system. Moreover, it can also be applied in the management system of education, medical treatment, corporation and other fields. In conclusion the flexible report system is of a great application foreground.

References

1. Yonggeng Zhou and Youman Deng: Design and application of a platform independent spreadsheet tool for power system, Power system technology, 26 (5) (2002) 57– 61
2. Marcello Bertoli and Andrew Stranieri: Forecasting on Complex Datasets with Association Rules**.** Lecture Notes in Artificial Intelligence, Vol. 3213. Springer-Verlag, Berlin (2004) 1171–1180
3. Ayman Ammoura, Osmar Zaïane, and Randy Goebel: Towards a Novel OLAP Interface for Distributed Data Warehouse. Lecture Notes in Computer Science, Vol. 2114, Springer-Verlag, Berlin (2001) 174 –185
4. Wang, S.L., et al.: A Try for Handling Uncertainties in Spatial Data Mining. Lecture Notes in Artificial Intelligence, Vol. 3215. Springer, Berlin (2004), 513–520
5. Aaron Ceglar, John Roddick, Paul Calder, Chris Rainsford: Visualizing hierarchical associations**.** Springer-Verlag Berlin. Knowledge and Information Systems (2004)

Privacy Preserving Naive Bayes Classification*

Peng Zhang, Yunhai Tong, Shiwei Tang, and Dongqing Yang

School of EECS, Peking University, Beijing, 100871, China
National Lab on Machine Perception, Peking University, Beijing, 100871, China
{zhangpeng, yhtong, tsw, ydq}@db.pku.edu.cn

Abstract. Privacy preserving data mining is to discover accurate patterns without precise access to the original data. In this paper, we combine the two strategies of data transform and data hiding to propose a new randomization method, Randomized Response with Partial Hiding (RRPH), for distorting the original data. Then, an effective naive Bayes classifier is presented to predict the class labels for unknown samples according to the distorted data by RRPH. Shown in the analytical and experimental results, our method can obtain significant improvements in terms of privacy, accuracy, and applicability.

1 Introduction

As the explosion in information technologies progresses, it has become more and more convenient to collect, manage, and analyze unprecedented amounts of data. Data mining has led to significant improvements in many applications, but privacy issues are also brought to us simultaneously. For example, a bank could classify new customers by their risks according to the old customer information. However, it is inescapable to uncover the customers' data so as to breach their personal privacies by the ordinary mining approaches. Therefore, how to solve the privacy preserving problem during the mining process has become one of the most important topics in data mining [1].

It is not the technology of data mining itself but the methods adopted that possibly breach the privacy. Data mining has an essential property that the patterns from large amounts of data usually depend on the aggregate and statistical data but not the individual data records. Then, can we discover accurate patterns without precise access to the original information in individual data records? The proposed approaches in privacy preserving data mining can be broadly classified into the following two categories: distortion based approach and partition based approach.

The central idea of the distortion based approach is that we first distort the values of the original data. Then, we need some reconstruction technique to discover the underlying patterns of the original data. A scheme using random distortion and distribution reconstruction was presented to implement a decision tree classifier in

* This work is supported by the National Natural Science Foundation of China under Grant No.60403041.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 744 – 752, 2005.

[2]. Based on that, an effective EM algorithm for distribution reconstruction was proposed [3]. However, they can only deal with numerical data, and not be applied to binary or categorical data. Then, a method to build decision tree classifier from the disguised data using Randomized Response Techniques (RRT) was presented in [4], but all the attributes in a record must be transformed by a same operation so that it is easy to distinguish whether a record in the disguised data set is true or false. The random parameter is also constrained to be not close to 0.5. Furthermore, the distorted data in [2, 3, 4] are all directly transformed from the original data, which may lead to degrade the privacy preserving level.

Besides the above approaches using data transform, a privacy preserving data mining scheme by limiting the sample size was proposed [5]. Then, a methodology for hiding knowledge in database was presented and applied to classification and association mining algorithms [6]. Although a part of sensitive information in [5, 6] is protected well, the supplied data for mining are all original and precise so that the privacy preserving level for the whole data set is not enough satisfactory.

The partition based approach is to use the Secure Multiparty Computation (SMC) techniques. Each data mining participant just has a part of the original data, and then the patterns are discovered by a set of secure protocols for distributed computation. Privacy preserving decision tree building methods were presented by SMC in [7, 8]. Then, privacy preserving naive Bayes classifiers for horizontally [9] and vertically [10] partitioned data were respectively proposed. But the approaches are meaningful only in the context of distributed database. All the data suppliers must participate in the mining process. One party fault may lead to error results, even mining failure, and the performance is not desirable when the party count becomes large.

In order to improve the accuracy in privacy preservation, we combine the two strategies of data transform and data hiding to present a new randomization method, RRPH, for distorting the original data. Then, a naive Bayes classifier on the distorted data set is implemented. The analytical and experimental results show that our method can obtain significant improvements in terms of privacy, accuracy, and applicability.

The remainder of this paper is organized as follows: In section 2, we define the problem and present a framework. RRPH method is described in section 3. Section 4 implements the privacy preserving naive Bayes classifier. Then, our classification method is analyzed in section 5. The experimental results are shown in section 6. Finally, in section 7, we summarize the conclusions and outline future work.

2 Framework

In the given data set D, each sample is represented by an n-dimensional feature vector, $X=(x_1, x_2, ..., x_n)$, where $x_i \in \{0, 1\}$, i=1, 2, ..., n, respectively describes whether it has the attribute A_i. If the sample X has the attribute A_i, x_i=1; otherwise, x_i=0. All samples in the data set D is supposed to divide into m classes, labeled as C_1, C_2, ..., C_m.

Given an unknown sample, Z (having no class label), the classifier will predict its class label according to the data set D. The privacy preserving classification problem is to predict the class label for Z without precise access to the original data set D.

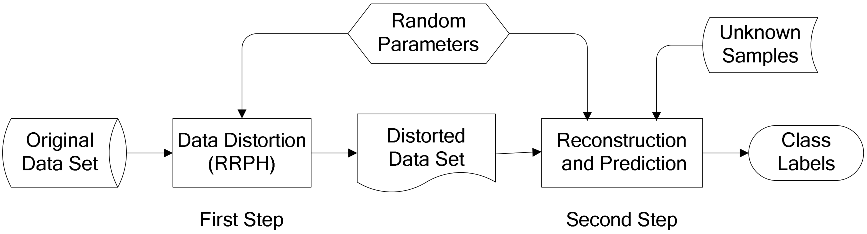


Fig. 1. Framework

In order to solve the problem, we present a scheme including two steps. Fig. 1 shows the framework. In the first step, we propose a new randomization method, RRPH, to distort the original data set. In the second step, we reconstruct the original distribution, and predict the class labels for unknown samples.

3 Randomized Response with Partial Hiding (RRPH)

The essential idea of RRT is that the original data are first distorted by the data supplier according to the random parameters, and then provided to the data user. Although, the detailed original data are distorted, the statistical and aggregate information can still be precisely estimated while there are large amounts of data. Classification depends on the aggregate information of a data set, but not the individual data records, so RRT can be well used for classification.

However, RRT used in the existing privacy preserving data mining approaches are all based on Warner's model. Not only are all the distorted data directly transformed from the original data so as to degrade the privacy preserving level, but also the random parameter is constrained to be not close to 0.5.

The strategy of data hiding can overcome the above disadvantage, but bring another one that the supplied data are all exactly the original data. How about integrating the above two strategies for data distortion to improve the accuracy of mining with better privacy preservation?

Just as this idea, we present a new randomization method, RRPH. The description of RRPH method is as follows:

Given a set of random parameters, $0 \le p_1, p_2, p_3 \le 1$, such that $p_1+p_2+p_3=1$. For each $x \in \{0,1\}$, let $r_1=x$, $r_2=1$, $r_3=0$, the random function $r(x)$ returns the value r_j with probability p_j (j=1, 2, 3). A sample in the data set D can be considered to be a vector $X=(x_1, x_2, ..., x_n)$, such that $x_i \in \{0,1\}$. We generate the distorted vector $Y=(y_1, y_2, ..., y_n)$ by computing $Y=R(X)$, where $y_i=r(x_i)$. That is, y_i takes a value x_i with probability p_1, 1 with probability p_2, and 0 with probability p_3.

In this way, each sample X in the original data set D becomes a distorted sample Y by the random function $Y=R(X)$. Since the form of the distorted sample Y is similar to an original sample, it can be added into the distorted data set D' as a disguised sample.

It seems that only the strategy of data transform is adopted to distort the original data. However, when we select r_2 or r_3 to perform the randomized response with the probability p_2 or p_3, the original samples are actually hidden, which is embodiment of the data hiding strategy.

In principle, it is possible to use different sets of random parameters for distorting different attributes. For simplicity, we will assume here that a single set of random parameters, p_1, p_2, p_3, for all the attributes.

The original data set D has become a distorted data set D' by RRPH method. Then, the process to reconstruct and estimate the distribution is described as follows:

Given an attribute A, let π be the proportion of samples having the attribute A in the original data set D, and λ be the proportion of samples having the attribute A in the distorted data set D'. Suppose a sample X becomes a distorted sample Y by RRPH method. Let X_A and Y_A respectively represent whether X and Y has the attribute A. Then, the values of X_A, Y_A, and their mapping probabilities are shown in Table 1.

Table 1. Probabilities of data mapping by RRPH

No.	X_A	Y_A	Probability
1	0	0	p_1+p_3
2	0	1	p_2
3	1	0	p_3
4	1	1	p_1+p_2

From Table 1, we can obtain $\lambda = \pi(p_1+p_2)+(1-\pi)p_2 = \pi p_1+p_2$. Then, $\pi = (\lambda-p_2)/p_1$.

We first estimate λ, and then π is to be computed by the above equation. Usually, we assume the two parameters p_2=p_3. Then, the proportion of samples having and not having the attribute A in the original data set can be estimated by the same equation.

4 Privacy Preserving Naive Bayes Classification

The naive Bayes classifier will predict Z belongs to the class having the highest posterior probability, conditioned on Z. That is, the naive Bayes classifier assigns an unknown sample Z the class C_i, if and only if $P(C_i|Z) \geq P(C_j|Z)$, $1 \leq j \leq m, j \neq i$.

By Bayes theorem, $P(C_i|Z)=\frac{P(Z \mid C_i)P(C_i)}{P(Z)}$. As $P(Z)$ is constant for all classes, only $P(Z|C_i)P(C_i)$ need be maximized. The class prior probabilities may be estimated by $P(C_i)=s_i/s$, where s_i is the number of training samples of class C_i, and $s=|D|$ is the total number of training samples. In order to reduce computation in evaluating $P(Z|C_i)$, the naive assumption of class conditional independence is made. Given the class label of the sample, that is, there are no dependence relationships among the attributes.

Thus, $P(Z|C_i)=\prod_{k=1}^{n} P(z_k \mid C_i)$. The probabilities $P(z_1|C_i)$, $P(z_2|C_i)$, ..., $P(z_n|C_i)$ can be estimated from the training samples, and $P(z_k|C_i)=s_{ik}/s_i$, where s_{ik} is the number of training samples of class C_i having the value z_k for attribute A_k in the data set D, and s_i is the number of training samples belonging to class C_i in the data set D. Then,

$$P(Z|C_i)P(C_i)=\frac{s_i}{s}\cdot\prod_{k=1}^{n}\frac{s_{ik}}{s_i} \tag{1}$$

Usually, the class label is not private information to be distorted. Therefore, the class labels of the distorted samples are still true information, and the values of $P(C_i)$ can still be exactly computed from the data set by $P(C_i)=s_i/s$. Suppose the values of the attributes are conditionally independent, then $P(Z|C_i)=\prod_{k=1}^{n} P(z_k \mid C_i)$. However, the probabilities $P(z_1|C_i)$, $P(z_2|C_i)$, ..., $P(z_n|C_i)$ cannot be directly estimated from the distorted training samples, but must be computed by reconstructing distribution based on RRPH method.

Let $P'(z_k|C_i)=s'_{ik}/s_i$, where s'_{ik} is the number of training samples of class C_i having the value z_k for attribute A_k in the distorted data set D', and s_i is the number of training samples belonging to class C_i in the distorted data set D'.

According to the last section, $P(z_k|C_i)=\frac{P'(z_k \mid C_i)-p_2}{p_1}=\frac{(s'_{ik}/s_i)-p_2}{p_1}$. Then,

$$P(Z|C_i)P(C_i)=\frac{s_i}{s}\cdot\prod_{k=1}^{n}\frac{(s'_{ik}/s_i)-p_2}{p_1} \tag{2}$$

In this way, when classifying an unknown sample Z, we can use the distorted data set D'. $P(Z|C_i)P(C_i)$ is evaluated by equation (2) for each class C_i. The sample Z is then assigned to the class C_i if and only if $P(Z|C_i)P(C_i) \geq P(Z|C_j)P(C_j)$, $1 \leq j \leq m, j \neq i$.

The major difference between the privacy preserving naive Bayes classifier and the original naive Bayes classifier is how to compute $P(Z|C_i)P(C_i)$, here the equation (1) has been replaced by the equation (2).

5 Analysis

In this section, our Privacy Preserving Naive Bayes (PPNB) classification method is amply analyzed in terms of privacy, accuracy, and applicability.

As its name implies, the motivation and purpose of privacy preserving data mining is to realize knowledge discovery and find potential patterns on the premise of privacy preservation. Therefore, the privacy preserving level is the most important measure to evaluate the approaches. In order to quantitatively compare the approaches better, we define a privacy measure, *Breach*, called privacy breach coefficient.

$$Breach = P_{\text{true}} \cdot P_{\text{identifying true}} + P_{\text{false}} \cdot P_{\text{identifying false}} \cdot P_{\text{reconstruction}} \tag{3}$$

Here, P_{true} and P_{false} are respectively the proportions of true and false data in the distorted data set, such that $P_{\text{true}}+P_{\text{false}}=1$. $P_{\text{identifying true}}$ is the probability for identifying true data from the distorted data set, while $P_{\text{identifying false}}$ is the probability for identifying false data from the distorted data set. $P_{\text{reconstruction}}$ is the probability with which the false data can be reconstructed to the relative original. Once the true data are identified, the privacy is breached. For the false data, privacy breaches need not only identifying the false data but also reconstructing the false data to the original.

Now, we compare the privacy preserving levels of the simple hiding method, DTPD method [4], and our PPNB method by respectively computing their privacy breach coefficients. Suppose the proportions of true original data in the distorted data sets are same for all these three methods. Then,

1. The simple hiding method, providing original data of proportion p. $Breach_1=p\cdot 1=p$.
2. DTPD method, the random parameter is p. $Breach_2= p\cdot p+(1-p)\cdot(1-p)\cdot 1= p^2+(1-p)^2$.
3. PPNB method, $p_1=p$, $p_2=p_3=\frac{1-p_1}{2}$, $Breach_3= p_1\cdot\frac{p_1}{p_1+p_2}=\frac{2p^2}{p+1}$.

Obviously, $Breach_1 > Breach_3$, but the relation between $Breach_2$ and $Breach_3$ depends on the value of the random parameter p.

$$\Delta_1=Breach_2-Breach_3=\frac{2p^3-2p^2-p+1}{p+1}=\frac{(\sqrt{2}p+1)(\sqrt{2}p-1)(p-1)}{p+1}.$$

Thus, when $0<p<\frac{1}{\sqrt{2}}$, $Breach_2 > Breach_3$. Therefore, PPNB method can obtain higher privacy preserving level than the simple hiding method and DTPD method.

It is fundamental for privacy preserving data mining to well preserve the privacy. But the final purpose of data mining is still to discover the useful patterns. Therefore, the results must be acquired by the mining approaches as accurately as possible.

Privacy and accuracy seem to be contradictive. The privacy preservation must lead to the descent of accuracy. In order to increase the accuracy, the privacy preserving level must be declined as the cost. Nevertheless, it is not true! The above analysis shows that PPNB method can obtain higher privacy preserving level compared with DTPD method. Now, we further explain by variance analysis that PPNB method can also make progress in terms of mining accuracy simultaneously, only if the random parameters are appropriately selected. We also assume the proportions of true original data in the distorted data set are same for the two methods. Then,

1. DTPD method, $\hat{\pi}_1=\frac{\lambda_1-(1-p)}{2p-1},(p\neq\frac{1}{2})$, variance $Var(\hat{\pi}_1)=\frac{p(1-p)}{n(2p-1)^2},(p\neq\frac{1}{2})$.
2. PPNB method, $\hat{\pi}_2=\frac{\lambda_2-p_2}{p_1}$, variance $Var(\hat{\pi}_2)=\frac{p_1(1-p_1)\pi}{np_1^2}+\frac{p_2(1-p_2)-2\pi p_1p_2}{np_1^2}$.

   For $p_1=p$, $p_2=p_3=\frac{1-p_1}{2}$, then $Var(\hat{\pi}_2)=\frac{p_2(1-p_2)}{np_1^2}=\frac{(1-p)(1+p)}{4np^2}$.

   Both $\hat{\pi}_1$ and $\hat{\pi}_2$ are enormous likelihood unbiased estimator for π, but

   $$\Delta_2=Var(\hat{\pi}_1)-Var(\hat{\pi}_2)=\frac{1-p}{n}\left[\frac{p}{(2p-1)^2}-\frac{1+p}{4p^2}\right]=\frac{(1-p)(3p-1)}{4np^2(2p-1)^2},(p\neq\frac{1}{2}).$$

   Therefore, when $\frac{1}{3}<p<1$, $Var(\hat{\pi}_1)>Var(\hat{\pi}_2)$.

Combined with the above analysis, we come to the conclusion: when $\frac{1}{3}<p<\frac{1}{\sqrt{2}}$, PPNB method can simultaneously provide higher privacy preserving level and higher mining accuracy than the existing DTPD method. The constraint to select the random parameter, i.e. $p\neq 0.5$, is also avoided.

PPNB method can be applicable to both centralized and distributed database. It is also of outstanding applicability in terms of mining functionality and data type.

In this paper, we present a scheme to solve the problem of privacy preserving classification, but the whole framework is divided into two independent steps. The first step of data distortion can be directly used when we apply the scheme for association rule mining, clustering, and other functionalities. Only the second step need modifying to use a relative mining algorithm for the distorted data by RRPH.

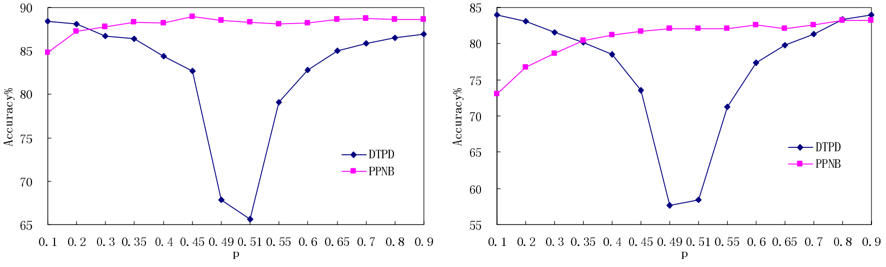
On the other hand, all the existing randomization methods can only deal with numerical data or binary data. None of them can be applied for categorical data. RRPH method can be extended to supply categorical data by adding additional random parameters and relative values of the random function results.

6 Experiments

Our experiments were carried out a synthetic data set and a real data set. The synthetic data set of credit card customers was generated for a bank system. Each customer has sixteen attributes and a class label describing whether he is valuable. We selected a dividing point for each attribute, and transformed them into binary type. 50,000 samples were selected as the training set, and another 5,000 samples as the testing set. The real data set of criminals was from a police system, containing 33,389 records with twelve attributes and a label describing the case type. The prediction task was to determine whether a suspect was related to murder. We used the first 30,000 records as the training samples, and the rest 3,389 records as the testing samples.

We selected different random parameters p_1=p=0.1, 0.2, 0.3, 0.35, 0.4, 0.45, 0.49, 0.51, 0.55, 0.6, 0.65, 0.7, 0.8, 0.9, p_2=p_3=(1-p_1)/2. Class labels of the testing samples were predicted according to the training samples respectively by PPNB method and DTPD method. Then, the prediction results of the two methods were also compared with the original class labels to compute their classification accuracies.

Fig. 2(a) shows the classification accuracies to the synthetic data set by PPNB and DTPD method for each different p values. The classification accuracies to the real data set by the two methods for each different p values are then shown in Fig. 2(b).



(a) The synthetic data set

(b) The real data set

Fig. 2. Classification Accuracy

When p is moving from 0 to 1, the classification accuracy of our PPNB method is continually enhancing, but the privacy preserving level is coming down. However, for DTPD method, when p is near to 0 or 1, the result is quite accurate but the privacy preserving level is fairly low. As p is away from 0 or 1, and approaches to 0.5, the privacy preserving level is increasing, but the accuracy is remarkably descending.

We can see from Fig. 2(a): when p is very small, DTPD method is more accurate than PPNB method; while p is over 0.3, the accuracy of PPNB method is increasing beyond DTPD method. Especially, when p is around 0.5, it can exceed DTPD method much more. The general result shown in Fig. 2(b) is similar to that in Fig. 2(a). The accuracies of the two methods both decline, but the descent is more notable for PPNB method. In Fig. 2(a), PPNB method is more precise as long as $p\geq0.3$; but in Fig. 2(b), only if p is from the interval [0.35, 0.8]. The reason is that naive Bayes classification need the assumption of class conditional independence, with which the synthetic data set is generated, but being inaccurate for the real data set.

However, PPNB method can still be effectively applied so long as we have a reasonable selection for the random parameters. We propose to select p from the interval [0.35, 0.6] to use PPNB method for privacy preserving classification by the trade-off between privacy preservation and mining accuracy.

7 Conclusions and Future Work

We have presented a privacy preserving naive Bayes classification method consisting of two steps: data distortion and classification on the distorted data set. The two strategies of data transform and data hiding are combined to propose RRPH method for data distortion. Then, the naive Bayes classifier is modified to predict the class labels for unknown samples according to the distorted data set. Our method has been analyzed in terms of privacy, accuracy, and applicability. With reasonably selection for the random parameters, it can provide a higher privacy preserving level to the users and retain a higher accuracy in the mining results, simultaneously.

In the future work, we will apply our method to solve incremental classification, and other data mining problems. We will also extend our RRPH method to support data distortion and distribution reconstruction for categorical data.

References

1. V. S. Verykios, E. Bertino, I. N. Fovino, L. P. Provenza, Y. Saygin, and Y. Theodoridis. State-of-the-art in Privacy Preserving Data Mining. In SIGMOD Record, 33(1), 2004.
2. R. Agrawal and R. Srikant. Privacy-Preserving Data Mining. In Proceedings of the ACM SIGMOD Conference on Management of Data, 2000.
3. D. Agrawal, C. C. Aggarwal. On the Design and Quantification of Privacy Preserving Data Mining Algorithms. In Proceedings of the 20th ACM Symposium on PODS, 2001.
4. W. Du and Z. Zhan. Using Randomized Response Techniques for Privacy-Preserving Data Mining. In Proceedings of the 9th ACM SIGKDD International Conference on KDD, 2003.
5. T. Johnsten and V. V. Raghavan. A Methodology for Hiding Knowledge in Databases. In Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining, 2002.

6. C. Clifton. Using Sample Size to Limit Exposure to Data Mining. Journal of Computer Security, 8(4), 2000.
7. W. Du and Z. Zhan. Building Decision Tree Classifier on Private Data. In Proceedings of the IEEE ICDM Workshop on Privacy, Security and Data Mining, 2002.
8. B. Pinkas. Cryptographic Techniques for Privacy-Preserving Data Mining. In SIGKDD Explorations, 4(2), 2002.
9. M. Kantarcoglu and J. Vaidya. Privacy Preserving Naive Bayes Classifier for Horizontally Pertitioned Data. In IEEE ICDM Workshop on Privacy Preserving Data Mining, 2003.
10. J. Vaidya and C. Clifton. Privacy Preserving Naive Bayes Classifier for Vertically Partitioned Data. In Proceedings of the 4th SIAM International Conference on DM, 2004.

A Further Study on Inverse Frequent Set Mining

Xia Chen[1,2] and Maria Orlowska[2]

[1] School of Electronic and Information Engineering,
Tianjin University, Tianjin 300072, P.R. China
[2] School of Information Technology and Electrical Engineering,
University of Queensland, QLD 4072, Australia
{chenxia, maria}@itee.uq.edu.au

Abstract. Frequent itemset mining is a common task in data mining from which association rules are derived. As the frequent itemsets can be considered as a kind of summary of the original databases, recently the inverse frequent set mining problem has received more attention because of its potential threat to the privacy of the original dataset. Since this inverse problem has been proven to be NP-complete, people ask "*Are there reasonably efficient search strategies to find a compatible data set in practice?*" [1]. This paper describes our effort towards finding a feasible solution to address this problem.

Keywords: Inverse Frequent Set Mining, Privacy Preserving Data Sharing, Equivalent Relation.

1 Introduction

Data mining technology has emerged as a means of identifying patterns and trends from large quantities of data. One of the key requirements of a data mining project is access to the relevant data. However, individual privacy concerns limit the willingness of the data custodians to share data. Frequent itemset mining is a common task in data mining from which association rules are derived. As the frequent sets can be considered as a kind of summary of the original database, by recently applying the inverse data mining technique to inference the original database from given frequent itemset collection with the support has received more attention because of its potential threat to privacy.

Inverse frequent set (or "itemset") mining can be described as follows: "Given a collection of frequent itemsets and their support, find a transactional data set such that the new dataset precisely agrees with the supports of the given frequent itemset collection while the supports of other itemsets would be less than the pre-determined threshold". This problem can help us understand how well privacy is preserved in the frequent itemsets and how well the frequent itemsets characterize the original data set.

Mielikainen first proposed this problem in his paper [1]. He proved that finding a binary dataset compatible with a given collection of frequent itemsets or

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 753–760, 2005.

deciding whether there is a dataset compatible with a given frequent set is NP-complete. Meanwhile, Calders in his paper [2] also proposed a similar problem, named “FREQSAT” which is described as “Given some itemset-interval pairs, finding a database such that for every pair the frequency of the itemset falls in the interval”. He claimed that FREQSAT is equivalent to the probabilistic satisfiability problem, and hence NP-complete. Under such circumstances, people asked: “[illegible] ” This paper describes our effort towards finding a feasible solution to address this inverse mining problem.

1.1 Related Works

Another similar study was reported by Ramesh in his paper [3] where he proposed a method to generate market basket data set for bench-marking when the length distributions of frequent and maximal frequent itemset collections are available. Unfortunately, even though the generated synthetic data set preserves the length distributions of frequent patterns, the problem is that the size of transactions generated usually is much larger than that of the original database while the number of items generated is much smaller.

Besides, Calder in his paper [2] also gave a naive “generate-and-test” method to “guess” a database D from given frequent itemsets as well as the number of transactions. The algorithm maintains m counters for given frequent itemsets $I = \{I_1, I_2, ..., I_m\}$, and 1 counter for transaction numbers $|D|$. It requests that every new transaction comes strictly lexicographically after the previous one. For every new transaction (tid, J), it increments the counter $|D|$, and checks for I_i. For all i such that $I_i \in J$, the counter for I_i is incremented. After at most $2^{|I|}$ guesses, the dataset generation stopped.

The above method has some shortcomings. The first problem is its huge computational cost because it blindly tries all possible value assignments, even those devoid of solutions. The second problem is that we should not take for granted that every transaction in the database is unique. As such, we think this algorithm lacks the ability of predicting the exact number of duplicates for each distinguished transaction, and therefore in most cases it might not be able to find a solution.

Under such circumstances, a feasible solution is still expected.

1.2 Paper Layout

The rest of the paper is structured as follows: In section 2, we further define our proposed problem and analyze the potential computational complexity. We then import an equivalent relation into this problem and declare that there must exist a set of equivalent datasets which are compatible with the given frequent itemset collection must exist. In section 3, we focus on demonstrating our proposed dataset generation algorithm. The last section outlines conclusions.

2 Problem Description

2.1 Inverse Frequent Set Mining

We begin with the introduction of notations used in this paper.

Let $I = \{I_1, I_2, ..., I_m\}$ be a set of items. Any $X \subseteq I$ is called an itemset. An itemset consisting of k elements is called a k-Itemset. Let $D = \{T_1, T_2, ..., T_n\}$ be a set of transactions, where each transaction T_i is an itemset. Given X is an itemset, we say a transaction T_i contains X, if $X \subseteq T_i$. The number of transactions stored in a database D is indicated as $|D|$. The support of an itemset $X \subseteq I$ is the cardinality of a set of all transactions containing X, denoted as: $Supp(X) = |T(X)| = |\{T_i | X \subseteq T_i, T_i \in D\}|$. Itemsets that meet a minimum support threshold "σ" are called frequent itemsets, and a collection of frequent itemsets are denoted as: $F(\sigma) = \{X | Supp(X) \geq \sigma, X \subseteq I\}$.

The objective of frequent itemset mining is to find all subsets that contained in at least "σ" fraction of transactions in the database. Conversely, inverse frequent itemset mining is defined as: Given a set of items $I = \{I_1, I_2, ..., I_m\}$, minimal support threshold σ, and a set of frequent itemsets $F = \{f_1, f_2, ...f_n\}$ with fix supports $S = \{Supp(f_1), Supp(f_2), ..., Supp(f_n)\}$, does there exist a database D such that it can satisfy the following constraints: (1) D is over the same set of items I; (2) From D, we can discovery exactly same set of frequent items "F" with the same support "S" under the same minimal support "σ";

2.2 Computational Complexity of the Proposed Problem

The computational complexity of this inverse data mining problem has been thoughtfully analyzed by Mielikainen and Calders .

Mielikainen in his paper [1] proved that deciding whether there is a data set compatible with the given frequent sets is NP-hard and computing the number of data sets compatible with the given frequent sets is #P-hard.

Calders in his paper [2] related the "FREQSAT" problem to the probabilistic satisfiability problem (pSAT) [4] where the items of the database are taken as variables, and the transactions as truth assignments over these items. The frequency of an given itemset $\{i_1, ..., i_k\}$ in the database D is now equal to the probability of the conjunction $\bigwedge_{j=1}^{k} i^j$ occurring in D. A transaction database then represents a probability distribution over the different truth assignments. Since the pSAT has been proven NP-complete, correspondingly the "FREQSAT" problem is NP-complete.

2.3 Equivalent Databases

Obviously, if the given frequent itemeset collection comes from a real database, at least this original database must be one which exactly agrees with the given even though it is computationally hard to "render" it. But is it unique?

Let's consider an example. Table 1 is a sample database with $I = \{A, B, C, D\}$ and $D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$. If we set $Min\_Supp = 2$, then the frequent itemsets we mined from D by using Apriori algorithm are $F = \{B_3, A_5, D_6, C_7,$

Table 1. Sample datbase D

Trans_ID	Items
1	ACD
2	ACD
3	BCD
4	AB
5	AD
6	BC
7	CD
8	CD
9	A
10	C

Table 2. Equivalent Databases of D

Trans_ID	D_1	D_2	D_3
1	ABCD	ABD	ACD
2	ACD	ACD	ACD
3	AD	ACD	AB
4	BC	BC	AD
5	CD	BC	BC
6	CD	CD	BC
7	CD	CD	CD
8	A	CD	CD
9	A	A	CD
10	B	A	A
11	C	-	-

$BC_2, AD_3, AC_2, DC_5, ADC_2\}$ where "X_n" denotes that "X" is an itemset and its support is equal to "n".

However, if F is given, we can find at least three datasets D_1, D_2, D_3 (as showed in Table 2) which satisfy all given constraints.

Definition 1. [illegible] D_1 [illegible] D_2 [illegible] $D_1 \approx D_2$ [illegible]

Apparently, the relation we defined is an equivalent relation as it satisfies reflexive, symmetric and transitive properties. We say $D \approx D_1 \approx D_2 \approx D_3$ because we can get exactly the same frequent itemset collection and support from them. Therefore, we slightly modify this inverse problem to that of finding a database which is equivalent to the original one.

[illegible] Given a set of frequent itemsets with a fixed support $< F, S >$, which is mined from a real dataset D, find a dataset D' such that $D' \approx D$.

The problem of finding a database which is equivalent to the original one has a very practical application in privacy preserving data sharing. It will give data owners an option of tailoring and releasing a database which is not equal, but equivalent to the original one for public sharing. It not only can protect the individual privacy of the original data, but will also keep the high utility of the released data in terms of minable frequent itemsets.

The challenge is: Can we find a feasible approach to find such kind of equivalent database?

3 Proposed Method

3.1 Basic Idea

Conceptually, a transaction database is a two-dimensional matrix where the rows represent individual transactions and the columns represent the items. This matrix can be implemented either horizontally or vertically. Horizontal representation means the database is organized as a set of rows with each row storing an ordered list of items, representing only the items actually purchased in the transaction. On the other hand, vertical representation means the database is organized as a set of granules with each column storing the items in the set of customer transactions.

Previous works in [2] and [3] focus on building the database horizontally with the generation-test framework. Unfortunately, both algorithms do not work well in terms of effectiveness and efficiency as we discussed in related work. In this paper, we propose an incremental vertical database generation algorithm.

We consider a database organized in vertical schema, and then instantiate items sequentially. Meanwhile, we consider the given frequent itemsets constraint to restrict the value assignment for each column. Since we only activate one item within an iteration, the constraint which we need to satisfy is only those rules related to current activated items. The lesser constraint means we can take less effort to find a solution. This way, we can reduce the computational complexity of this problem to some extent. Besides, when we are instantiating a column, we also adopt the “generate and test” schema. In the data generation phase, we try to add the new element into every history transaction and at the same time record both the new subsets it brings and the ones which might already be in the given frequent set. In the test phase, we compare the support of those “old” itemsets with the one in the given set and meanwhile check if the support of those “new” itemsets are still below the given minimal support threshold. If we find any conflict, we simply give up all operations which have been done in the data generation phase and therefore restore their initial state.

3.2 Algorithm

The sketch of this algorithm is given as follows. The input of the algorithm is a set of items I, minimal support threshold σ and frequent itemsets F with the support S. The output is a new database D'.

```
Gen_DB($F, I, \sigma$)
Begin
1. $D' \leftarrow \emptyset$, $R \leftarrow \emptyset$;
2. $I' \leftarrow$Sort items in I by the number of frequent itemsets they involved in descending
   order;
3. For each item $i \in I'$,
   (a) $R \leftarrow R \cup \{i\}$;
   (b) $F_i \leftarrow \emptyset$;
   (c) For each rule $f \in F$,
         i. If $i \in f$ and $f \subseteq R$, Then
            $F_i \leftarrow F_i \cup f$, $F \leftarrow F - f$;
   (d) Sort rules in $F'$ by the length in descending order;
   (e) Gen_Bitmap($i$);
4. Return $D'$;
End.
```

```
Gen_Bitmap($i$)
Begin
  1. $F_s \leftarrow \emptyset$;
  2. For each transactions $d \in D'$,
     (a) Generate:
          i. Set $d \leftarrow d \bigcup i$;
         ii. For all $x \subseteq d$ and $i \in x$,
             If $x \in F_i$, Then $Supp(x) = Supp(x) - 1$
             Else   Adding it into $F_s$;
     (b) Test:
          i. If (for all $x \in F_i$, $Supp(x) > 0$) and
             (for all $y \in F_s$, $Supp(y) < \sigma$) and
             (Agree with all additional constraints), Then Continue;
         ii. If (for all $x \in F_i$, $Supp(x) = 0$) and
             (for all $y \in F_s$, $Supp(y) < \sigma$) and
             (Agree with all additional constraints), Then Return($D'$);
        iii. If (for all $x \in F_i$, $Supp(x) = 0$) and
             (for all $y \in F_s$, $Supp(y) < \sigma$) but
             {Violate any additional constraint}, Then
             Go back to the last transaction d' where $i \in d'$, $d \leftarrow d'$;
     (c) Rollback:
          i. Reset $d \leftarrow d - i$;
         ii. Rollback all operations on $F_i$ and $F_s$;
End.
```

The algorithm is composed of two procedures. In the main procedure "Gen_DB()", we use R to record the items which have been processed so far and F_i to record all frequent itemset $f \in F$ while $f \subseteq R$ as the constraint. In the sub-procedure "Gen_Bitmap()", we use F_s to store all "new" itemsets which are not in F_i. For those itemsets which are already in F_i, we simply modify their support each times. Once all of those supports are equal to "0", the current data assignment becomes acceptable.

Let's illustrate our main idea with an example: "Given frequent itemsets collection $F = \{B_3, A_5, D_6, C_7, BC_2, AD_3, AC_2, DC_5, ADC_2\}$, $I = \{A, B, C, D\}$, and $Min\_Supp = \sigma$".

We first sort items in I in descending order according to the number of frequent itemsets they involve. The purpose of doing this is because the item with more constraints is better to be processed early in order to avoid the potential high computational complexity as the number of items increase. In this case, after this pre-processing, we get $I' = \{C, D, A, B\}$. Then we will instantiate them one by one.

During the first iteration, "C" is instantiated, so $R = \{C\}$, $F_i = \{C_7\}$ and $F_s = \{\}$. According to the algorithm, we add seven "C"s into D'={C,C,C,C,C,C,C}. Then we add a new column "D" into the database and $R = \{C, D\}$, $F_i = \{D_6, DC_5\}$ and $F_s = \{\}$. The item "D" will be sequentially added into each transactions in D' and meanwhile all new itemsets which can be mined from this new transaction, but not in F_i will be automatically added in F_s, and then will be tested whether or not any conflict exists between transactions and constraints. In this case, after we have five "CD", $F_i = \{D_1, DC_0\}$, $F_s = \{\}$. Then the algorithm will not add any other "CD" into D', but will accept a new transaction which has only one "D" such that $F_i = \{D_0, DC_0\}$, $F_s = \{\}$. Finally we get $D' = \{CD, CD, CD, CD, CD, C, C, D\}$ in this iteration.

Now we begin to process "A" and $F_i = \{A_5, AD_3, AC_2, ACD_2\}$, $F_s = \{\}$. Firstly, we generate two "CDA"s, and $F_i = \{A_3, AD_1, AC_0, ACD_0\}$, $F_s = \{\}$. Then, no more "CDA" is allowed. But after adding a "DA" into D', $F_i = \{A_2, AD_0, AC_0, ACD_0\}$, $F_s = \{\}$. Finally, we add another two "A"s into D', such that $F_i = \{A_0, AC_0, ACD_0\}$. So after this iteration, we get $D' = \{CDA, CDA, CD, CD, CD, C, C, DA, A, A\}$. Lastly, we instantiate the column "B", $F_i = \{B_3, BC_2\}$, $F_s = \{\}$. We first add a "CDAB" into D' and $F_i = \{B_2, BC_1\}$, $F_s = \{ABCD_1, ABC_1, ABD_1, BCD_1, AB_1, BD_1\}$. We can not add any one "CDAB" or "CDB" because they both will bring another "BCD" such that $Supp(BCD) > \sigma$. But we can have one "BC" and one "B" such that $F_i = \{B_0, BC_0\}$ while $F_s = \{ABCD_1, ABC_1, ABD_1, BCD_1, AB_1, BD_1\}$ are below the threshold. Therefore, from this iteration, we get D'={CDAB, CDA, CD, CD, CD, CB, C, DA, A, A, B}.

3.3 Tailoring the Data Generation Process

As we discussed above, given a set of frequent itemsets and their supports mined from a real database D, there might exist a set of databases which are different from D, but equivalent to D. From our proposed algorithm, we are able to find one of them. However, can we tailor the data generation process to specifically find a dataset which is either most similar to the original database or the least similar to it? From the aspect of data utility, more similarity means keeping higher data utility. But from the aspect of privacy preserving data sharing, more similarity means that more privacy of the original database might be disclosed.

We define that two databases are similar to each other in terms of database characteristics, more common properties, and more similarities. The main char-

Table 3. Database Characteristics of D, D_1, D_2, D_3

-	D	D_1	D_2	D_3
the Number of Transactions	10	11	10	10
the Maximal Length of Transactions	3	4	3	3
the Minimal Length of Transactions	1	1	1	1
the Length Distribution of Transactions				
4-Length	0	1	0	0
3-Length	3	1	3	2
2-Length	5	5	5	7
1-Length	2	4	2	1
The Number of In-Frequent Itemsets	2	6	3	1

acteristics of a database we are interested in here are: (1) the Number of Transactions; (2)the Maximal and Minimal Length of Transactions; and (3)the Number (Up-bound or Lower-bound) of in-Frequent Itemsets.

We list the main characteristics of datasets D, D_1, D_2, D_3 in Table 3. After comparing the similarity in terms of the database characteristics of these four datasets. We say that D_1 is the least similar one to D and D_3 is the most similar. So if some prior-knowledge about the original database is given in advance, this can be taken as "additional constraints" by the algorithm. Once the algorithm detected any violation on these constraints, it will automatically rollback current value assignment or even go back to the last assignment and try again. In this way, the algorithm can gradually approach the dataset you expected.

4 Conclusions

The work presented in this paper is a further study on the inverse frequent set mining problem. Our study focused on looking for a feasible algorithm to find a compatible data set from a given frequent set collection. We propose an algorithm which can incrementally generate a database compatible with the given set. Furthermore, we studied the special case where if we have enough prior-knowledge about the original database, how can we control the data generation process so as to find the one we expect. This study is just our first step towards solving this problem. More work will be done in the near future, such as refinement of the algorithm, and empirical experiments on real databases. However, our study has shown that in some cases, inverse frequent set mining does have a solution - the databases which are equivalent to the original. By using proper search strategies, people are able to find at least one of these databases. This study can be used to deal with privacy preserving data sharing. It will give data owners another choice in releasing a different version of the original data for public sharing.

References

1. Mielikainen, T.: Inverse frequent set mining. In: IEEE ICDM Workshop on Privacy Preserving Data Mining, Melbourne, Florida, USA, IEEE (2003) 18–23
2. Calders, T.: Computational complexity of itemset frequency satisfiability. In: the 23nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database System, ACM Press (2004)
3. Ramesh, G., Maniatty, W., Zaki, M.: Feasible itemset distribution in data mining: theory and application. In: the 22nd ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems, San Diego, CA (2003) 284–295
4. Lukasiewicz, Thomas: Probabilistic logic programming with conditional constraints. ACM Transactions on Computational Logic (TOCL) **2** (July 2001)

Mathematical Analysis of Classifying Convex Clusters Based on Support Functionals

Xun Liang

Institute of Computer Science and Technology,
Peking University, Beijing 100817, China
Department of Economic Systems and Operations Research,
Stanford University, Stanford, CA 94305, USA
liangxun@icst.pku.edu.cn

Abstract. Classification is one of the core topics in data mining technologies. This paper studies the geometry of classifying convex clusters based on support functionals in the dual spaces. For the convex clusters that are to be classified, a combination of linear discriminant functions could solve the problem. The geometrical depiction of linear discriminant functions and supporting hyperplanes for the convex clusters help to characterize the relations of the convex clusters, and the distances to the convex clusters and complement of convex clusters calibrate the measures between the support functionals and convex clusters. Examples are given.

1 Introduction

In the past years, data mining has surged as a new technology in both academic areas and industrial applications. Many algorithms and theories have been developed [3, 4, 6, 7, 8, 14]. Classification [1, 2, 6, 11] is one of the most well-developed technologies in the data mining field. For the clusters to be classified, linear or nonlinear discriminant functions are used to divide the different clusters. For the case that the clusters (or the hull of the clusters) are convex, a combination of the linear discriminant functions is able to solve the problems [6, 11]. In this paper, we illustrate the classification problem for the convex clusters based on support functionals in the dual spaces.

The geometry of separations for convex clusters was explored in [10] by viewing the minimum distance to a convex cluster K, proving to be the minimum norm duality. It demonstrates that the shortest distance from a point y to the convex cluster K should be equal to the maximal distance from y to the supporting hyperplane H (see Fig.1).

This paper provides an illustration from a different perspective of the problem: Given a convex cluster K in the whole space X, and a point y in K, what is the shortest distance from y to the complement set $X \backslash K$ of the convex cluster K, in term of the supporting hyperplane H (see Fig.2). Then, this paper addresses the separation problem of convex clusters by demonstrating the problem from both inside and outside of the convex clusters, hence relating the both

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 761–768, 2005.

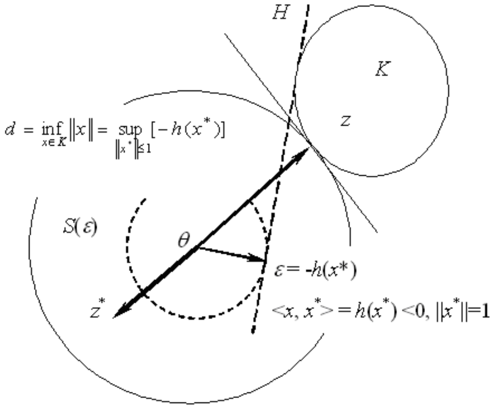

H
K
z
$d = \inf_{x\in K}\|x\| = \sup_{\|x^*\|\le 1}[-h(x^*)]$
$S(\varepsilon)$
θ
$\varepsilon = -h(x^*)$
z^*
$<x, x^*> = h(x^*) < 0, \|x^*\| = 1$

Fig. 1. The distance from the origin θ to a convex cluster K. Theorem 1 can be geometrically explained as follows: $\sup_{\|x^*\|\le 1}[-h(x^*)] = \sup_{\|x^*\|\le 1}[-\sup_{x\in K} < x, x^* >] = \sup_{\|x^*\|\le 1}[\inf_{x\in K} < x, -x^* >] \overset{\text{if } x\|-x^*}{=} \sup_{\|x^*\|\le 1}[\|x\|\,\|-x^*\|] \overset{\text{if } x=z,-x^*=-z^*}{=} \sup_{\|z^*\|\le 1}[\|z\|\,\|-z^*\|] = \|z\|\,\|z^*\|\,|_{\|z^*\|=1} = \|z\| = \inf_{x\in K}\|x\|$. Therefore, we have $d = \inf_{x\in K}\|x\| = \|z\| = \sup_{\|x^*\|\le 1}[-h\,(x^*)]$, and the alignment of z and $-z^*$

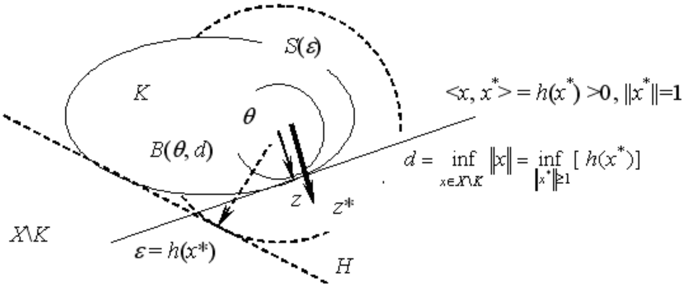

$S(\varepsilon)$
K
$<x, x^*> = h(x^*) > 0, \|x^*\| = 1$
θ
$B(\theta, d)$
$d = \inf_{x\in X\setminus K}\|x\| = \inf_{\|x^*\|\ge 1}[h(x^*)]$
z
z^*
$X\setminus K$
$\varepsilon = h(x^*)$
H

Fig. 2. The distance from the origin θ to set $X\setminus K$. Theorem 2 can be geometrically explained as follows: $\inf_{\|x^*\|\ge 1}[h(x^*)] = \inf_{\|x^*\|\ge 1}[\sup_{x\in K} < x, x^* >] \overset{\text{if } x\|x^*}{=} \inf_{\|x^*\|\ge 1}[\,\|x\|\ \|x^*\|\,] \overset{\text{if } x=z,x*=z^*}{=} \inf_{\|z^*\|\ge 1}[\,\|z\|\ \|z^*\|\,] = \|z\|\ \|z^*\|\,|_{\|z^*\|=1} = \|z\|$. Therefore, we have $d = \inf_{x\in X\setminus K}\|x\| = \|z\| = \inf_{\|x^*\|\ge 1}[\,h\,(x^*)]$, and the alignment of z and z^*

mathematical results. The geometry of classifying the convex clusters is depicted in both cases.

The process of training or learning in the data mining practice is to find the appropriate discriminant functions and support functionals in the dual space.

This paper is organized as follows. Section 2 reviews the minimum distance from a point to a convex cluster. Section 3 develops the problem from the other side, by calculating the minimum distance from a point in the convex cluster to complement of a convex cluster. Examples of classifying convex clusters are given in section 4. Section 5 concludes the paper.

2 Minimum Distance to a Convex Cluster

First, we review some definitions used in this paper.

Definition 1. [illegible] X [illegible] X [illegible] X [illegible] X^*

The norm of an element $f \in X^*$ is $\|f\| = \sup\limits_{\|x\| \leq 1} |f(x)|$.

Definition 2. [illegible] K [illegible] X [illegible] $h(x^*) = \sup\limits_{x \in K} < x,\, x^* >$ [illegible] X^* [illegible] K

Definition 3. [illegible] H [illegible] X [illegible] K [illegible] K [illegible] H [illegible] H [illegible] $\overline{K}$ [illegible] $\overline{K}$ [illegible] K

Definition 4. [illegible] $x* \in X^*$ [illegible] $x \in X$ [illegible] $< x,\, x^* >= \|x\|\; \|x^*\|$ [illegible] $x||x^*$

Alignment is a relation between vectors in two distinct linear spaces: a normed space and its normed dual.

Theorem 1. *(* [illegible] *)* [illegible] X [illegible] K [illegible] h [illegible] $y \in X \backslash K$ [illegible] d [illegible] y [illegible] K [illegible]

$$d = \inf_{x \in K} \|x - y\| = \sup_{\|x^*\| \leq 1} [< y, x^* > - h(x^*)]$$

[illegible] $z \in K$ [illegible] $z^* \in X^*$ [illegible] $z - y|| - z^*$ *(* [illegible] *)*

3 Minimum Distance to Complement of a Convex Cluster

In this section, the problem is studied from the inside of a convex cluster.

Theorem 2. [illegible] X [illegible] K [illegible] h [illegible] $y \in K$ [illegible] d [illegible] y [illegible] $X \backslash K$ [illegible]

$$d = \inf_{x \in X \backslash K} \|x - y\| = \inf_{\|x^*\| \geq 1} [\ h(x^*) - < y, x^* >]$$

[illegible] $z \in X \backslash K$, [illegible] $z^* \in X^*$ [illegible] z^* [illegible] $z - y$

[illegible] First, we prove that case of $y = 0$. We must show $d = \inf_{x \in X \backslash K} \|x\| = \inf_{\|x^*\| \geq 1} [\ h(x^*)]$. In fact, it is easy to show that from the Eidulheit Separation Theorem [10], there exists $x^* \in X^*$ with $\|x^*\| = 1$, such that,

$$h(x^*) = \sup_{x \in K} \langle x,\ x^* \rangle \leq \langle x_n,\ x^* \rangle \leq \|x_n\|\ \|x^*\| \overset{\|x^*\|=1}{=} \|x_n\| \tag{1}$$

Then we have $\inf_{\|x^*\| \geq 1} h(x^*) \leq d$. Let $B(0,\ d)$ be the sphere with radius d and centered at 0, $B(0,\ d) \subset K$. $\forall\ x^* \in X^*$, if $\|x^*\| \geq 1$, we have

$$h(x^*) = \sup_{x \in K} \langle x,\, x^* \rangle \overset{K \supset B(0,\, d)}{\geq} \sup_{x \in B(0,\, d)} \langle x,\, x^* \rangle \overset{\|x^*\| \geq 1}{\geq} \sup_{x \in B(0,\, d)} \|x\| \geq d \tag{2}$$

Hence, $d = \inf_{x \in X \backslash K} \|x\| = \inf_{\|x^*\| \geq 1} h(x^*)$.

In addition, if z achieves the distance d from x to $X \backslash K$, there exists $z^* \in X^*$ with $\|z^*\| = 1$, such that $\langle x,\ z^* \rangle \leq \langle z,\ z^* \rangle, \quad \forall\, x \in K$. Thus

$$h(z^*) = \sup_{x \in K} < x,\ z^* > \ \leq\ < z,\ z^* > \leq\ \|z\|\ \|z^*\| \overset{\|z^*\|=1}{=} \|z\|\ =\ d \tag{3}$$

We already know from the above

$$h(z^*) \geq \inf_{\|z^*\|=1} h(z^*) \geq \inf_{\|z^*\| \geq 1} h(z^*) = d \tag{4}$$

Combining (3) with (4), we have $h\ (z^*) = \langle z,\ \ z^* \rangle =\ \|z\|\ \|z^*\| = d$, and z is aligned with z^*.

Next we extend the result to the case of $y \neq 0$.

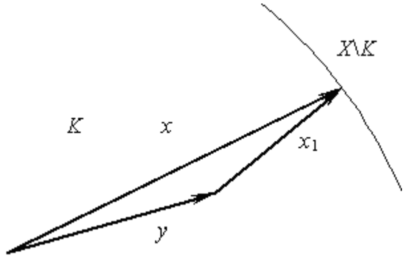


Fig. 3. The distance from the new origin to a convex cluster K

Table 1. Comparison of the two figures. The key difference is row 1. It results in the differences in the signs of $h(x^*)$, as well as the directions of z and z^*

	Theorem 1	Theorem 2
Positions of y and convex cluster K with respect to $h(x^*)$	Different sides	Same side
Signs of $h(x^*)$	Negative	Positive
Directions of z and z^*	Different	Same

Table 2. Comparison between Theorems 1 and 2

In X	Theorem 1 $d = \inf\limits_{x\in K} \|x-y\|$ $= \sup\limits_{\|x^*\|\leq 1} [< y, x^* > -h(x^*)]$, $-z^*$ is aligned with $z - y$	Theorem 2 $d = \inf\limits_{x\in X\backslash K} \|x-y\|$ $= \inf\limits_{\|x^*\|\geq 1} [h(x^*) - < y, x^* >]$, z^* is aligned with $z - y$
In X^*	$d = \inf\limits_{x^*\in K} \|x^* - y^*\|$ $= \sup\limits_{\|x^{**}\|\leq 1} [< y^*, x^{**} > -h(x^{**})]$, -$z^{**}$ is aligned with z^*- y^* (A special case: Theorem 2 on page 121 [10])	$d = \inf\limits_{x*\in X\backslash K} \|x^* - y^*\|$ $= \inf\limits_{\|x^{**}\|\geq 1} [h(x^{**}) - < y^*, x^{**} >]$, z^{**} is aligned with z^* - y^*

Now we move y (originally, $=\theta$) to a new place where $y \neq \theta$, $x = y + x_1$ (see Fig. 3). Hence $x_1 = x - y$. $x \in X\backslash K$. Substituting it into (**??**), we have

$$\begin{aligned}
\inf_{x_1\in X\backslash K} \|x_1\| &= \inf_{x\in X\backslash K} \|x-y\| = d = \inf_{\|x*\|\geq 1} h(x*) \\
&= \inf_{\|x^*\|\geq 1} \left[\sup_{x_1\in K} < x_1,\ x^* > \right] = \inf_{\|x^*\|\geq 1} \left[\sup_{x\in K} < x-y,\ x^* > \right] \\
&= \inf_{\|x*\|\geq 1} \left[\sup_{x\in K} < x,\ x* > - \sup_{x\in K} < y,\ x* > \right] \\
&= \inf_{\|x^*\|\geq 1} \left[\sup_{x\in K} < x,\ x^* > - < y,\ x^* > \right] \\
&= \inf_{\|x^*\|\geq 1} \left[h(x^*) - < y,\ x^* > \right]
\end{aligned}$$

The following table compares the differences of Theorems 1 and 2.

It is not difficult to follow the similar steps in [10] and section 3 of this paper, and obtain the other two mirror theorems in X^*.

4 Examples

In this section, we exemplify the above theorems for some special cases. Even these examples of convex clusters would be trivial and not frequently used in the real world, they are helpful to portray the classification problems of convex clusters.

4.1 If *K* Is a Sphere

If K is a compact sphere with center θ, suppose $y = \theta$, from Theorem 2, we obtain

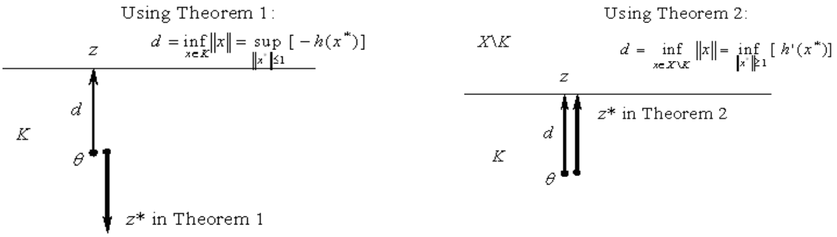
$$d = \inf_{x \in X \setminus K} \|x\| = \inf_{\|x^*\| \geq 1} h(x^*) = \inf_{\|x^*\| \geq 1} \sup_{x \in K} \langle x, x^* \rangle \overset{\text{if } x \| x^*}{=} \inf_{\|x^*\| \geq 1} [\ \|x\|\ \|x^*\|\ ] \overset{\text{if } x = z, x^* = z^*}{=} \|z\|\ \|z^*\|\ |_{\|z^*\| = 1} = \|z\|\ \|z^*\|\ |_{\|z^*\| = 1} = \|z\| (\text{radius}).$$

4.2 If *K* Is a Half-Space

Let K is a half-space. Then K is convex. Using Theorem 1 (see Fig.4 (a)), we have that the distance $d = \inf_{x \in K} \|x\| \quad = \sup_{\|x^*\| \leq 1} [\ -h(x^*)]$.

Since $X \setminus K$ is the complement of K, Theorem 2 can be used (see Fig.4 (b)), we have that the distance $d = \inf_{x \in X \setminus K} \|x\| = \inf_{\|x^*\| \geq 1} [\ h'(x^*)] = \inf_{\|x^*\| \geq 1} [\ -h(x^*)] = \sup_{\|x^*\| \leq 1} [\ -h(x^*)]$.

Consequently, Theorems 1 and 2 show us the same result.


Using Theorem 1:
$d = \inf_{x \in K} \|x\| = \sup_{\|x^*\| \leq 1} [\ -h(x^*)]$
z
d
K
θ
z* in Theorem 1
Using Theorem 2:
X\K
$d = \inf_{x \in X \setminus K} \|x\| = \inf_{\|x^*\| \geq 1} [\ h'(x^*)]$
z
z* in Theorem 2
d
K
θ

(a) (b)

Fig. 4. The distance d from the origin θ to set $X \setminus K$ by Theorems 1 and 2

4.3 If *K* Is a Polyhedral Convex Cluster

If K is a polyhedral convex cluster, we have a further result.

Theorem 3. *([illegible]) [illegible]* 1^*, 2^*, m^* [illegible] $H_j = \{x \in X | < x, x_j* >= \alpha_j\}, \alpha_j \in R, j \in M = \{1, 2, \ldots, m\}$ [illegible] $V = \bigcap_{j \in M} \{x \in X | < x, x_j* >= \alpha_j\} \neq \phi$ [illegible] $y \in V$ [illegible] d [illegible] $X \setminus V$ [illegible] z [illegible] *([illegible])*

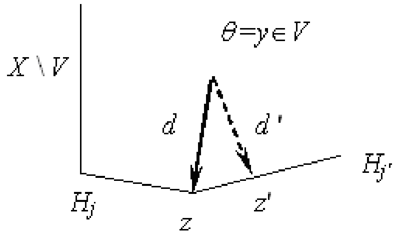


Fig. 5. The global minimum distance point z is perpendicular to the hyperplane, but cannot be at the intersection of hyperplanes

Without loss of generality, we take $y = 0$.

First, we need to prove that if $z \in H_j$,$\|z\| = \inf_{x\in H_j} \|x\|$, then $z\perp \quad H_j$. Since $H_j - z$ is a closed subspace and $-z \notin H_j$, from Projection Theorem, we know that $\forall x_1$,$x_2 \in H_j - z, < -z| \quad x_1 - x_2 >= 0$. Now plus $H_j - z$ and -$z$by z, we have H_j is a closed linear variety and $\theta \notin H_j$, and also $< z|(x_1 + z) - (x_2 + z) >=< z| \quad x_1 - x_2 >= 0, \forall x_1$,$x_2 \in H_j$. Therefore, $z\perp \quad H_j$.

Suppose z and z' achieves the global minimum by distance d on different hyperplanes H_j and $H_{j'}$ respectively. Let $z\perp \quad H_j$ where $z \in H_j \quad \cap \quad H_{j'}$, and $z \neq z'\perp \quad H_{j'}$where $z' \notin H_j, z' \in H_{j'}(j' = 1,\ldots,m; j' \neq j)$. By Projection Theorem used on $H_{j'}$, z' is unique, which is contradicting with the other global minimum distance point $z \in H_j \cap H_{j'}$. As a result, the global minimum distance point cannot be at the intersection of hyperplanes.

5 Conclusion

Theorems 1 and 2 form a group of twins that characterizes the geometry of support functionals from in and out of convex clusters from different standing

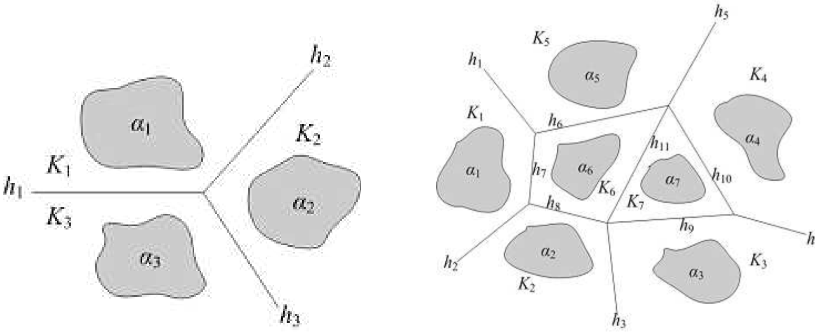


Fig. 6. In some cases, the clusters with arbitrary shapes can be included in convex hulls

points. The support functionals depict the relationship between the supporting hyperplanes and convex clusters.

Many clusters are not convex in nature. However, the hulls of them are convex (see Fig.6). As a result, the problem can be treated as classification problems for convex clusters. The support functions are helpful for us to gain insight of the data.

References

1. Agrawal, R., Ghosh, S., Imielinski, T., Lyer, H., Swami, A.: An interval classifier for database mining applications. In: Proc of VLDB Conf (1992) 560-573
2. Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J.: Classfication and Regression Trees. Wadsworth International Group (1984)
3. Dunham, H. Data Mining—Introductory and Advanced Topics. Prentice Hall (2003)
4. Ester M., Sander K. H, Xu, X.: A density-based algorithm for discovering clusters in large spatial database with noise. In:Proc of 2nd Int Conf on Knowledge Discovery and Data Mining Portland (1996) 226-231
5. Groth, R. :Data Mining—Building Competitive Advances. Prentice Hall (2000)
6. Guo, X., Liang, X.: Simulation study on data mining algorithm SLIQ in large databases. *Chinese Journal of Management Science* 12(2004) 79-82
7. Han, J., Kamber, M.: Data Mining—Concepts and Techniques. Morgan Kaufmann (2001)
8. Hand, D., Mannila, H., Smyth, P.: Principles of Data Mining. The MIT Press (2001)
9. Hiriart-Hurruty, J. B., Lemarechal, K.: Convex Analysis and Minimization Algorithms (1993) 1-127 Springer-Verlag
10. Luenberger, D. G.: Optimization by Vector Space Method. John Wiley & Sons 121 133 (1969) 136-137
11. Mehta, M., Agrawal, R., Rissanen, J.: A fast scalable classifier SLIQ for data mining. In:Proc of Int Conf Extending Database Technology (1996) 18-32
12. Quinlan, J. R.: C4.5: Programs for Machine Learning. Morgan Kaufman (1993)
13. Raileanu, L. E., Stoffel, K.: Theoretical comparison between the cini index and information gain criteria. Annals of Mathematics and Artificial Intelligence 41(1) (2004) 77-93
14. Witten,I. H,Frank,E.: Data Mining Morgan Kaufmann (2000)

Linear Belts Mining from Spatial Database with Mathematical Morphological Operators

Min Wang[1,2], Jiancheng Luo[2], and Chenghu Zhou[2]

[1] Nanjing Normal University, Nanjing, 210097, China
[2] State Key Laboratory of Resources & Environment Information System,
Institute of Geographical Sciences and Natural Resources Research,
Beijing, 100101, China
wangm@lreis.ac.cn

Abstract. In order to mine one typical non-sphere cluster, the linear belts in a spatial database, a mathematical morphological operator based method is proposed in this paper. The method can be divided into two basic steps: firstly, the most suitable re-segmenting scale is found by our clustering algorithm MSCMO which is based on mathematical morphological scale space; secondly, the segmented result at this scale is re-segmented to obtain the final linear belts. This method is a robust mining method to semi-linear clusters and noises, which is validated by the successful extraction of seismic belts.

1 Introduction

Clustering is a primary data mining method for structure or knowledge discovery in spatial databases [1][2]. Since spatial data often cluster as non-spherical (ellipsoid) shape, in which seismic belts are typical examples, spatial clustering algorithms need to be able to identify irregular shapes [3].

The main aim of this paper is to cluster and extract the linear belts because they are important non-sphere clusters. Since natural linear belts (for example, seismic belts) are natural phenomena which can be detected or observed within a certain observing scale range, methods for the mining of such linear clusters should take scale into consideration. To the best of the authors' knowledge, this is not covered by many classical linear belts clustering methods such as Hough transform [4], Fuzzy C-Lines [5], etc.

Scale space theory is a framework for early visual operations developed in the computer vision community for the handling of multi-scale nature of image data [6]. Scale space can be constructed with many morphological filtering operators. In this paper, we realize spatial data clustering with morphological scale space. Similar work can be found in [7], which proposed a clustering algorithm with the closing operator with structuring elements increasing iteratively in size, and used the heuristic method to find the best number of clusters. Di et al. however, didn't describe their algorithm from the viewpoint of scale space, and they didn't give thorough research on how to specify the precision of the raster image and how to remove noises to avoid their disturbing the subsequent morphological operations.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 769 – 776, 2005.

With special reference to Di's work but adopting the scale space point of view, we have proposed a multi-scale clustering method named Multi-scale Clustering Algorithm with Mathematical Morphological Operators (MSCMO) [8]. In this paper, we improve it to mine linear features. The idea is to use MSCMO to get the most suitable scale of re-segmenting the seismic belts, and then obtain the final belts with further morphological processing.

2 Algorithm of MSCMO

We use MSCMO for clustering spatial data with non-sphere shapes. In order to apply the morphological operators to spatial clustering, the vector data set should first be rasterized. Thereafter, a grid covering the whole experimental area with specified precision should be created, and the cells which contain data points should be marked with the value '1', and be marked with the value '0' otherwise. The binary morphological operators take effects only after these operations have been performed.

Through experiments, we find that if noises exist the subsequent creation of scale space will be disturbed seriously. Thus, it is advantageous to perform two operators in sequence: closing then opening with the same 3×3 round structuring element (the smallest template with the property of rotation invariance) to remove all the noises from subsequent morphological operations, but they can be reallocated to some clusters in the end.

For that, we should separate the cells contain noises for more than one cell wide. If two cells containing noises are only with (or within) one cell in distance, they will conglutinated with each other in the operation of closing and survive in the noise removing. We can find the distance threshold between noises and non-noises with the sorted 1-dist graph [9], and set the grid size to less than half of this 1-dist value, which will basically satisfy the need of removing noises.

The algorithm of MSCMO:

```
Input: data set, the precision of grid to be
constructed, the structuring element $B_1$ to remove the
noises (3×3 round structuring element by default)

Output: the clusters

Construct a grid covering the whole testing area by the
specified precision, then convert the vector data space
into raster and create the binary image $S$;

$S = S \bullet B_1 \hbar B_1$, that's, to first close then open, remove
all the noises;

Set current scale $c$=2, with corresponding structuring
element $B_c$;

Count the number of clusters via $COUNT$, loop when
$COUNT$>1{

    $S = S \bullet B_c$
  $c=c+1$
};
Select the number of clusters which has relatively the
longest lifetime as the final clustering, and select
the image which first brings this number as the final
segmentation (clustering) result;
Output the clusters.
```

3 MSCMO in Linear Belts Extraction

We first use MSCMO extract the skeletons of the segmented image at the most suitable scale, get the nodes of the skeletons with hit-or-miss transform and ‘smash’ them to split the skeletons into the arcs, then recombine these arcs into several groups of ‘the longer the better’, ‘the more straight the better’ linear (or near linear) axes, then re-segment the image using the information of nodes, skeletons and axes into several linear (or near linear) belts. We declare that the gotten belts be very close to the true seismic belts which validates the practicability of our method.

3.1 Skeletons Extraction of an Image

Our skeleton extraction method is formulated on the basis of the thinning operator which is based on Hit-or-Miss Transform (HMT) [10].

The eight couples of structuring elements we use to extract the skeletons of an image with thinning are:

$$\begin{bmatrix} 0 & 0 & 0 \\ * & 1 & * \\ 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} * & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & * \end{bmatrix}, \begin{bmatrix} 1 & * & 0 \\ 1 & 1 & 0 \\ 1 & * & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 & * \\ 1 & 1 & 0 \\ * & 0 & 0 \end{bmatrix},$$

$$\begin{bmatrix} 1 & 1 & 1 \\ * & 1 & * \\ 0 & 0 & 0 \end{bmatrix}, \begin{bmatrix} * & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & * \end{bmatrix}, \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ * & * & 1 \end{bmatrix}, \begin{bmatrix} 0 & 0 & * \\ 0 & 1 & 1 \\ * & 1 & 1 \end{bmatrix},$$

where ‘0’ means the missed elements, ‘1’ means the hit elements, and ‘*’ means the positions requiring no action.

3.2 Nodes Search

The nodes can be detected with HMT which needs 16 couples of structuring elements:

$$\begin{bmatrix} * & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ * & 1 & * \end{bmatrix}, \begin{bmatrix} 1 & 0 & * \\ 0 & 1 & 1 \\ * & 1 & * \end{bmatrix}, \begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ * & * & * \end{bmatrix},$$

and the forms of rotating them by 90,180 and 270 degree. All of the nodes can be obtained by:

$$Nodes = \mathop{\hbar}\limits_{i=1}^{16} (S * B_i) \tag{1}$$

In formula 1, *Nodes* represent the node set, and B_i are a couple of structuring elements. '*' is the HMT. The skeletons can then be split into arcs by smashing the nodes.

3.3 Axes Search and Seismic Belts Re-segmenting

The basic idea of this step is to extract the longest arc (*A*) among the arcs, and start from a node of *A*, to get the arc (*B*) which has the smallest corner with *A* in the neighborhood (the corner Φ needs to be smaller than some specified threshold Φ_t, should be specified by the user). Then, we recalculate the angle combining *A* and *B*, and start with the other node of *B*. The above operation is repeated until it meets the end. We then start from the other node of *A*, and repeat the operation till it meets the other end. The above operation will obtain the axis of one seismic belt, and all the arcs searched would be exempted from the subsequent search. The other axes will similarly be extracted by the above procedure by starting with the longest arc which has not been searched. The whole process is to be repeated until no more axes can be extracted.

After all the axes have been obtained, we use the information about the nodes, skeletons and axes to re-segment the image at the most suitable scale to get the seismic belts.

4 Experimental Analyses

Several experiments have been carried out to validate our method. We will give two typical examples in this paper. The data set comes from real-life earthquake data collected in China by the seismic analysis and forecasting center in 1980 and 1989. The object is to mine seismic belts from this data set.

4.1 Experiment 1

In this experiment, we extract 3201 seismic events with magnitude ≥2.2 in the area of [34º-42ºN, 106º-115ºE]. We could easily visualize in Figure 1(a) two nearly parallel seismic belts (in broken lines), which correspond to the north segment of the North-South seismic belt (on the left) and the Shanxi seismic belt (on the right).

The inputs of the experiment are: the grid cells: 150×150; corner threshold Φ_t=60°. The lifetime of the number of clusters along the scale is depicted in Table 1. We can observe that the lifetime of 2 clusters is the longest, while that of 3 clusters is the second longest. By comparing the images at scale 18 which starts the 2 clusters and the scale 14 which starts the 3 clusters, we could find that the connected components in the latter image are actually closer to the true seismic belts. It indicates that 3 is the most suitable number of clusters.

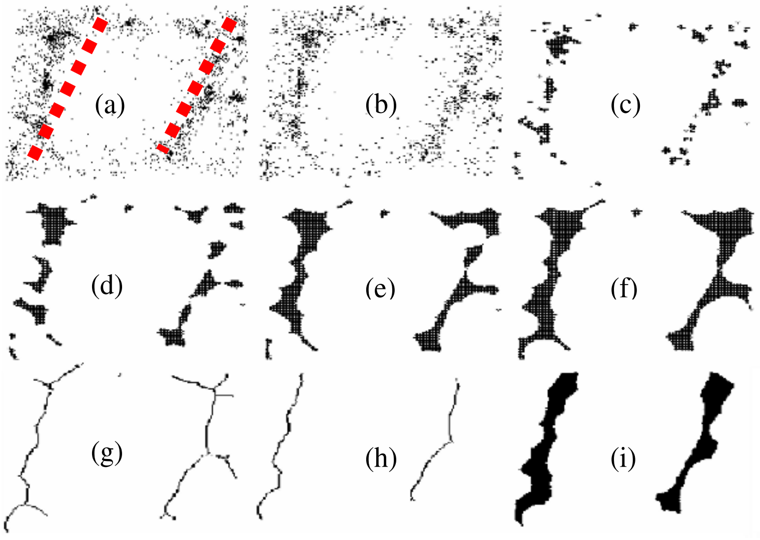


Fig. 1. Mining of seismic belts: (a) original vector-based data set; (b) rasterized image; (c) first scale removed the noises; (d) scale 5; (e) scale 10; (f) scale 13; (g)skeletons; (h)axes of the two longest linear belts; (i) two belts extracted

Table 1. Lifetime of the number of clusters of some scales

Scale	11	12	13	14-17	18-30	31
number	7	6	4	3	2	1

The image of scale 14 is then re-processed with the morphological operations proposed. Figure 1(i) shows the two seismic belts obtained which are very close to the actual seismic belts.

4.2 Experiment 2

In experiment 2, we extract 5294 seismic events with magnitude ≥3.0 in the area of [32º-42ºN, 109º-122º]. There are three main seismic belts which are conglutinant with each other and in which the upper one is in near arch shape. The inputs are: grid cells: 150×150, noises removed with 3×3 round structuring element, corner threshold Φ_t=60°, and trimming threshold=6 cells in length.

We also use MSCMO to extract the most suitable image (see Figure 2). It can be observed that the clustering stabilized at scale 9 with two clusters (we could find in Figure 2 that the segmented image is very different from the actual seismic belts, which must be reprocessed).The image at scale 9 is then used to extract the skeletons, obtain the axes and then extract the linear belts. The three longest linear belts obtained are very close to the actual seismic belts, which indicates the practicality of our method.

As a comparison, we use Fuzzy C-Lines to extract the belts the same data set. We find that Fuzzy C-Lines is very sensitive to noises. So we need to remove all the noises. The inputs of Fuzzy C-Lines are: *m*=2, the number of clusters=4 (considering to the short linear belts in the center of the image), and 100 iterations. The image in the bottom-right corner of Figure 2 presents the central lines of the final clusters, and the points distributed around a central line will belong to it. From this image, we find that the upper seismic belt is split apart which indicates in contrast the advantage of our method: i.e., robust to the 'not very linear' clusters. Besides, a cluster composed by the points with very large interspaces is obtained by Fuzzy C-Lines (see the bottom-right in Figure 2(o)), which is not very reasonable. This shows that our method does a better job on this data set.

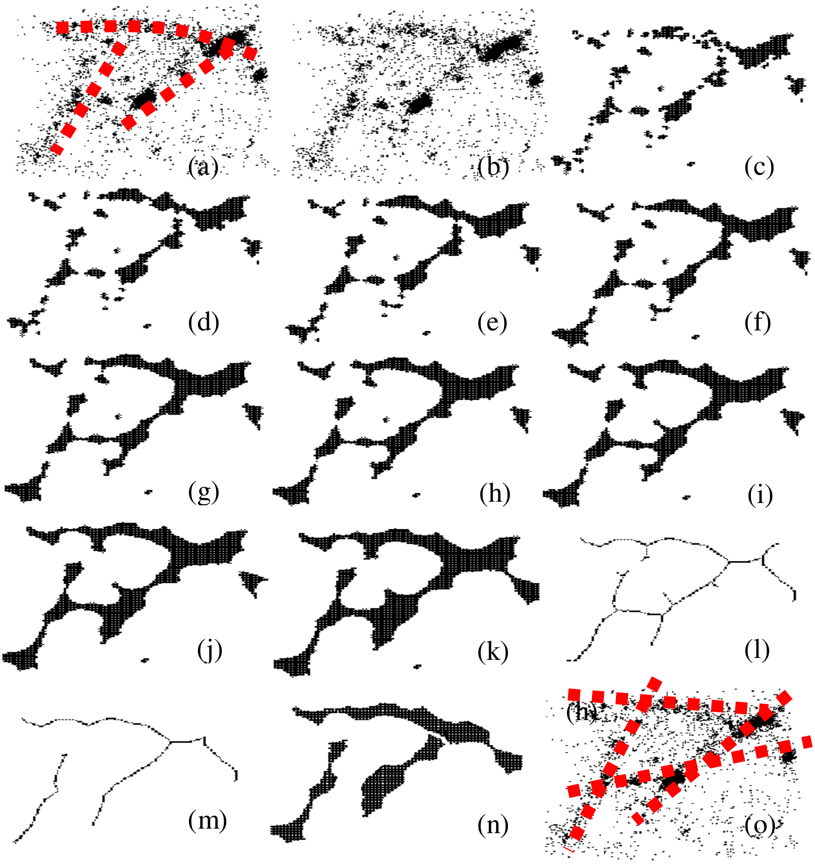


Fig. 2. Experimental area:(a)original data (b)raster image (c)image with noises removed (d)-(j)scale 2-8 (k)the most suitable scale: scale 9 (l)skeletons (m)axes (n)linear belts (o)clustering result of Fuzzy C-Lines

Table 2. Lifetime of the number of clusters of some scales

Scale	5	6	7	8	9-20	21
number	9	8	7	3	2	1

5 Conclusions

In this paper, a multi-scale mining method for linear belts is proposed whose advantages are validated through experiments. The core of this method is the clustering algorithm MSCMO which is based on mathematical morphological scale space. The final linear belts are gotten by the re-segmentation of the image at the most suitable scale obtained by MSCMO. MSCMO is a binary image segmenting method in nature, but through our vector to raster converting method; it can be used in clustering the vector data. After being enhanced, it becomes a good mining method robust to the 'not very linear' features.

In future work, the amalgamation of spatial supervising knowledge in the creation of scale space, the extended application and improvement in mining the other data types of line, polygon and higher algorithm efficiency are what we should pay attention to.

Acknowledgment. This work is supported by Chinese National Natural Science Foundation under grant No.40401039 and Chinese Postdoctoral Foundation.

Reference

[1] Koperski K., Adhikary J., and Han J: Spatial Data Mining: Progress and Challenges Survey Paper. Proc. ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery, Montreal, Canada, (1996)

[2] Ester, M., Kriegel, H.P., Sander, J. and Xu, X: Clustering for Mining in Large Spatial Databases. Special Issue on Data Mining, KI-Journal, 12(1998)18-24

[3] Sander, J., Ester, M., Kriegel H., and Xu, X: Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and Its Applications. Data Mining and Knowledge Discovery, 2(1998)169-194

[4] Asano, T., and Katoh, N: Variants for the Hough transform for line detection. Computational Geometry, 6(1996)231-252

[5] Bezdek, J.C., Coray, C., Gunderson, R. and Watson, J: Detection and characterization of cluster substructure: I. Linear structure: Fuzzy C-lines. SIAM J. Appl. Math.,40(1981): 339-357

[6] Lindeberg, T: Scale-space: A framework for handling image structures at multiple scales. Proc. CERN school of Computering, Egmond aan Zee, The Netherlands, (1996)

[7] Di, K., Li, D.L. and Li, D.Y: A Mathematical Morphology Based Algorithm for Discovering Clusters in Spatial Databases, Journal of Image and Graphics, 3(1998)173-178

[8] Wang Min, Zhou Cheng-hu, Pei Tao, Luo Jian-chen: MSCMO: A Scale Space Clustering Algorithm Based on Mathematical Morphology Operators, Journal of Remote Sensing, 1(2004)45-50

[9] Ester M, Kriegel H P, Sander J, Xu X: A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: Proc. of the Second International Conference on Knowledge Discovery and Data Mining, Portland, Oregon, (1996)324-331

[10] He, B., Ma, T., Wang, Y. and Zhu, H: Digital Image Processing with Visual C++. Beijing, China: People's Posts and Telecommunications Press, (2001)

Spatial Information Multi-grid for Data Mining

Zhenfeng Shao and Deren Li

State Key Laboratory for Information Engineering in Surveying,
Mapping and Remote Sensing, Wuhan University, P.R. China
shaozhenfeng@163.com, dli@wtusm.edu.cn

Abstract. Driven by the issue of geo-spatial data mining under grid computing environment, a new representation method called spatial information multi-grid (SIMG) for depicting spatial data and spatial information is presented in this paper. A strategy of dividing the globe with multi-level spatial grid is proposed. And through further studying on the precision of description on feature of detail, this paper tries to divide the globe with SIMG and constructs the framework of SIMG in China. Based on SIMG this paper tries to realize data mining of different thematic information on the same geographical position, data mining of dynamic spatial-temporal information of the same thematic content, data mining and transforming of different coordinate systems and to make SIMG a fundamental research work for further developing spatial information sharing and data mining.

1 Introduction

Grid is a rising research field in recent years. The purpose of grid technology is to combine all kinds of data resources over the whole Internet to make a huge computer and thus to realize resource sharing and cooperation [3]. Currently multi-scale representation and organization of global geographical information has gotten extensive attention in geographic field and many specialists and researchers have devoted their energies on this topic.

In 1992, Goodchild and Yang presented a hierarchical data structure based on global geographical information system [1]. In 1998, Sahr and White discussed the architecture of discrete global grid system (DGGS) [2]. In 2000, Geoffrey proposed a dividing method, global hierarchical coordinates [4], based on the thought of Goodchild [1]. This method constitutes global multi-scale grid through carrying on octahedral quaternary tri-angular grid (O-QTM) step by step [5].

The method of grid division mentioned above actually belongs to the category of resolving spatial positioning and spatial search mechanism of Geographical Information System (GIS) from a technological view point. How to solve the problem of data management in different coordinate systems is a basic issue we must face up to. For this reason, this paper presents "SIMG of the Earth" based on grid technology. Basic reference, projection technology, different data representation inside the grid, and conversion methods are given special emphases.

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 777 – 784, 2005.

Compared with single computer circumstance, the challenges of grid computing environment to GIS are obvious. Further, with the rapid development of database technology and geographical information system, the applications of data analysis are beyond simply browsing and searching. This paper aims to breakthrough traditional ideas of representing geo-spatial data and information and tries to explore SIMG for data mining on all kinds of applications.

2 Methodology

The core idea of SIMG is to represent the whole globe according to the different Latitude/Longitude Grid of different levels. The geographical position of a grid is identified by the longitude and latitude coordinates of its central point. The basic data items closely related to this grid are recorded (such as longitude and latitude, the coordinates of central point under all kinds of projection systems, etc). The feature (feature of detail) in each grid records the relative position to the central point of current grid, and the coordinate of the detailed features inside a given grid can be converted quickly and flexibly according to the corresponding central point among different projection systems. As a new spatial data organization and description methods, SIMG is looked on as the foundation of organization and management at the node of grid computing environment, so as to facilitate of storing, merging, sharing and utilizing the spatial and non-spatial data. Two key technologies of SIMG are considered helpful for data mining. One is SIMG division, the other is precision analysis of grid dividing in SIMG. Based on the two key technologies, various data mining applications including relationship confirmation between administrative territory and SIMG, information index via grid index, statistics information mining for macroscopical decision-making become more easily.

2.1 A Spatial Grid Division Method of SIMG for Data Mining

The basic idea of grid division strategy of SIMG for data mining can be illustrated through Fig.1.

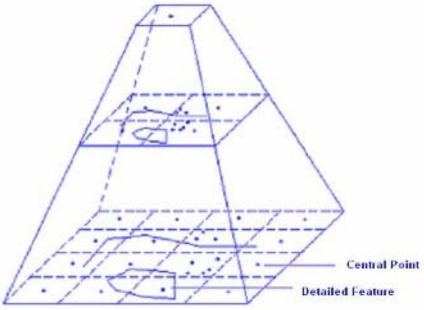


Fig. 1. Grid Division in Spatial Information Multi-Grid (SIMG)

Certain size of longitude and latitude is appointed as the standard of basic grid. Based on this basic level, grid of the next level is divided according to equal longitude and latitude and coded in the way of quad-tree. To keep consistency with current map scale in our country, this paper selects 6°of longitude and 4°of latitude as the division reference of the basic grid.

Fast query technologies based on SIMG is realized through grid index. According to the basic idea of SIMG, a two-level index is appointed to spatial objects. One is grid index among different levels and the other is inter-grid index. Fast conversion between grid and geographical coordinate can be realized simply through grid coding, so data mining can be achieved automatically through the existing spatial digital products based on SIMG. Fig.2 shows the basic grid division result of China with Latitude/Longitude and Fig.3 shows further multi-level subdivision with quad-tree.

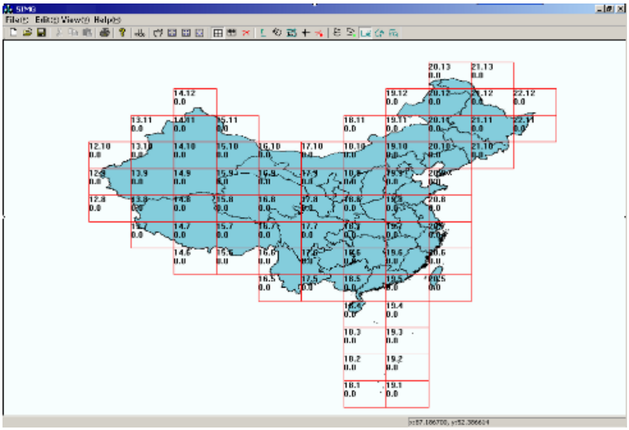


Fig. 2. Basic Grid Division Result of Whole China with Latitude/Longitude

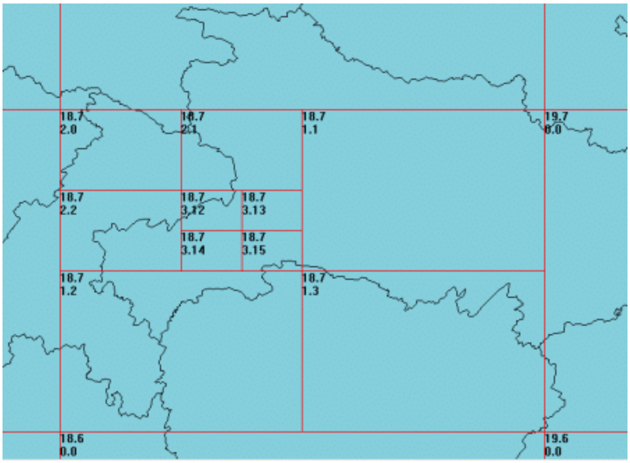


Fig. 3. Multi-level Division with Quad-tree

2.2 Precision Analysis of SIMG for Data Mining

SIMG attempts to design a uniform reference to represent spatial data of different resolution through grids with different size and level. To meet this need, a key factor is to find a quick conversion method with rather high precision. Generally, there are two conversion methods among different coordinate systems. One is rigorous conversion between two different spheroid systems and the other is affine approximate conversion between two planar systems. The former is appropriate for large extent and its precision is higher comparatively. But its computing procedure to acquire the conversion parameters is complicated and time-consuming. The latter is appropriate for small extent and has quick computing speed. As rigorous conversion in SIMG is a very complicated process, rigorous conversion is replaced by affine conversion method to improve the efficiency. If more than three homonymic points can be found in two coordinate systems, then the planar coordinate conversion between different systems can be realized through simple affine conversion. If the rigorous coordinates of n points checked inside of grids at all levels are known, we can select a quicker conversion method through comparison of the precisions of rigorous conversion and affine conversion methods.

In this paper two districts are selected to test precision analysis of our division method. One is from north 38°to 42°and from east 76°to 83°. The other is from north 36°50' to 38°50' and from east 90°50' to 93°50'. In Table 1, μ represents the mean error of checked points; *Max* is the largest error of checked points, the authors select meter as the basic unit; *N* is the number of the checked point whose error is greater than twice of μ; and *Tol* is the tolerance of relevant scale.

Table 1. Precision of Points with Affine Approximate Conversion in SIMG

Grid	*Tol*	Basic Grid				2				… 16				
	/m	Points	μ /m	*Max* /m	*N*	*Points*	μ /m	*Max* /m	*N*	…	*Points*	μ /m	*Max* /m	*N*
1	200	8000	245.2	516.3	1	4000	240.7	507.1	1	…	500	197.0	427.5	0
2	100	4000	118.9	244.1	2	2000	112.8	243.5	3	…				
3	50	4000	57.2	121.8	3	2000	55.7	126.8	2	…				
4	25	4000	29.9	59.9	1	2000	28.4	60.1	1	…				
5	12.5	4000	15.5	35.5	2	2000	14. 6	34.4	3	…				
6	6.3	4000	7.0	14.4	1	2000	7.1	17.6	7	…				
7	3.1	2000	3.9	7.1	8	1000	3.5	7.7	7	…				
8	1.6	2000	1.8	4.0	5	1000	1.7	3.8	4	…				
9	0.8	2000	0.9	1.9	2	1000	0.9	1.9	4	…				
10	0.4	1000	0.4	1.0	3	500	0.4	0.8	4	…				
11	0.2	1000	0.2	0.5	4	500	0.2	0.5	7	…				

In our experiments, coordinate of four corner points and checked points within grids at all levels are generated firstly according to designed division method. And then based on global 30-arc-second digital elevation data offered by the National Geological Investigation Bureau of U.S.A., digital surface model is generated through bi-linear interpolating method. Then the authors acquire the coordinates of corner points and checked points at 1954 Beijing coordinate system, 1980 Xi'an coordinate system, WGS-84 coordinate system respectively by using rigorous coordinates conversion method with the aid of national basic parameters. To take position error into account, three groups of random errors are added to three sets of coordinates, in which the ground resolution of the grid is conditioned. The authors regard 6°×4°as the basic grid and there are 8000 checked points in basic grid and then subdivide the grid step by step. If the grid of different levels is subdivided according to the dividing method shown in Table 1, the real-time coordinate conversion can be realized through affine conversion instead of rigorous conversion method. From Table 1 we can find out that if 16×16 subdivision method is selected, the need of mapping with the scale of 1:10000 can be met when the basic grid is divided once. Experimental results also show that If we subdivide the basic grid 11 times, according to 2×2 subdividing way, only 7 points' coordinates exceed the limit among 500 check points, and the result can meet the mapping with the scale of 1:500 on the whole. This scale is our biggest standard scale at present, so based on this precision we can describe all surface features, linear features and point features with SIMG. A point feature can be represented as a point within a given grid whose position records the relative coordinates to the central point of current grid. A line feature is recorded with several points, which may be in different grids. A surface feature can be divided into several sub-surfaces, and each sub-surface is in one given grid. The grid object table forms the grid database.

2.3 Different Thematic Data Mining Based on SIMG

As a new data expression method, SIMG should face a series of challenges in the course of expressing spatial information and act as a carrier of non-spatial information under grid computing environment, which includes extracting implicit, unknown, uncommon and potentially valuable information or patterns from large database or data warehouse. Using SIMG to store and search data from Grid database is a useful and new method.

Through exploiting grid to representing the information, the central point's coordinates of a grid can be varied among different coordinate systems, the increment of the coordinate is too approximate and can be converted through simple affine conversion in the grid, so can realize data mining of different thematic contents on the same geographical position. If we add different thematic information to the grid, we can mine and analyze the data automatically, then inductively conclude and associate them to find some inner relationship among the data and get some very useful information which is easily ignored by experts otherwise.

Different applications need support of spatial information at different scales. They also have relevant request to the spatial reference coordinate systems and projection types. As the center coordinates of the grids in SIMG are uniform Earth's core coordinates, so we can implement precise coordinate system conversion (as the anchor point) to the grid central point, and then finish the conversion from the object coordinate system to the application coordinate system with affine conversion method.

2.4 Dynamic History Information Mining Based on SIMG

It is well known that data mining not only needs to search and browse spatial data, but also needs to find potential relations within them to further improve the information discovery ability. Data mining needs the operators provide static and dynamic historical data at all levels and forecast data as well. Furthermore because the time of acquiring data for different thematic in the same spatial area may be inconsistent, data can not be simply superposed in analyzing. Based on SIMG, we can form a spatial extrapolative data model of extrapolating the data of different field thematic inside or outside till they can be considered as data of the same time, and then the data mining procedure on modeling and assimilating process can be finished via spatial information multi-grid technique.

3 Application Study

SIMG presents a new kind of spatial data representation method in Grid Computing environment and its mostly effective applications include real-time data mining, data analysis and macroscopical decision-making. Fig 4 illustrates a typical application of conversion between SIMG and administrative territory. In this instance a method of Hubei province territory fitting from grid database based on SIMG is presented.

When the Size of Grid is assigned as the threshold, if the threshold is 10km^2, then the total amount of grid covered by Hubei province is 4391 and the total area is 193,039km^2. The ratio of the total area covered by grid to those covered by the province's administrative territory is 101.5%. The area of the province covered by the grid is188,648km^2. The ratio of the area covered by the grid to the area covered by the total administrative territory is 99.2%. The results are shown in Fig.4 and Table 2.

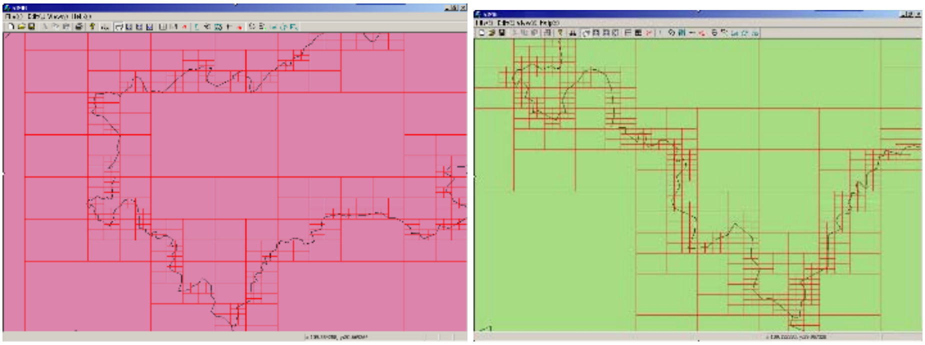


Fig. 4. The figure illustrates (1)Visualization that Assigns the Size of Grid as the Threshold, and (2) Visualization that Assigns the Overlapped Area as the Threshold

If the overlapped area is assigned as the threshold, the results are shown in Table 3. For example, if 1km^2 is selected as the threshold, the total amount of grid covered by Hubei province is 3510 and the total area is 188,650 km^2.

Table 2. Fitting Results that Assign the Size of Grid as the Threshold

Fitting method	Total amount of grid	Amount of grid at different level		Ratio of the total area covered by grid to that covered by the administrative territory	Ratio of the area covered by the grid to the total administrative area
Minimum overlapped area is $10km^2$	4391	2	3		
		3	15	($193,039 km^2$)	($188,648km^2$)
		4	29	101.5186%	99.2090%
		5	91		
		6	188		
		7	496		
		8	3182		
Minimum overlapped area is $1km^2$	9084	2	3	($191,557km^2$)	($189,755km^2$)
		3	15	100.7389%	99.7912%
		4	29		
		5	91		
		6	188		
		7	496		
		8	993		
		9	7289		

Table 3. Fitting Results that Assign the Overlapped Area as the Threshold

Fitting method	Total amount of grid	Amount of grid at different level		Ratio of the total area covered by grid to that covered by the administrative territory	Ratio of the area covered by the grid to the total administrative area
Minimum overlapped area is $1km^2$	3510	2	3	($188,650km^2$)	($188,184km^2$)
		3	15	99.2101%	98.9650%
		4	32		
		5	86		
		6	200		
		7	481		
		8	1088		
		9	1605		
Minimum overlapped area is $0.4km^2$	5410	2	3	($188,637km^2$)	($188,315km^2$)
		3	15	99.2033%	99.0339%
		4	30		
		5	91		
		6	195		
		7	488		
		8	1062		
		9	2388		
		10	1138		

Experimental results illustrate that SIMG can realize mutual conversion of the existing database and grid database, and prove SIMG is a new method for data mining, data representation.

Of course SIMG can be applied to some other kinds of fields of data mining. Based on SIMG, we can extract data from grid database for data analysis via the conversion from SIMG to traditional spatial databases. For example, the census of any region can be finished almost real-time if we have built the grid database of population. Various aspects of information of the same region can be synthesized or integrated for macroscopical decision-making easily based on SIMG. The grid provides service for users to produce, release, search, analyze, process, acquire and use the data in grid database. Moreover, grid environment solves the problem of interoperation among network systems, the data communication on network and so on.

4 Conclusion and Future Work

This paper presents a new method of spatial information representation method for data mining. Key technologies of SIMG such as grid division and index of the detailed feature within grid database are discussed. SIMG for data mining is emphasized. Typical applications of automatic conversion between SIMG and administrative territory are implemented. The results show that SIMG can be extremely helpful for data mining. The study shows that it is a feasible direction to mine data based on the strong Grid technology. Critical issues worthy of further research include: Analyzing model and service model for data mining based on SIMG; Combining SIMG with spatial data warehouse and spatial datum cube for on-line data analysis and data mining. Our related research is on its way.

Acknowledgments

This paper is supported by the Chinese national 863 Project (NO. 2003AA132080).

References

1. Goodchild, M.F., Yang, S: A Hierarchical Data Structure for Global Geographic Information Systems. Computer Graphics, Vision and Image Processing, 54(1): (1992) 31–44
2. Sahr, K., White, D: Discrete global grid systems. Proceedings of the 30th Symposium on the Interface, Computing Science and Statistics 30 (1998) 269–278
3. Ian, F., CarlKesse, L., Steven, T: The Anatomy of the Grid: Enabling Scalable Virtual Organizations. International Journal Supercomputer Applications, 15(3): (2001) 200–222
4. Geoffrey, D: Universal geo-spatial data exchange via global hierarchical coordinates. International Conference on Discrete Global Grids, 3, (2000) 1–15
5. Denis, W: Global Grids from Recursive Diamond Subdivisions of the Surface of an Octahedron or Icosahedron. Environmental Monitoring and Assessment,64(1):(2000) 93–103

A Uniform Framework of 3D Spatial Data Model and Data Mining from the Model

Peng-gen Cheng

Department of Surveying, East China Institute of Technology,
Fuzhou 344000, China
pgcheng@ecit.edu.cn

Abstract. A uniform framework of 3D spatial data model (UF-3DSDM) is proposed in this paper. It is a union of various 3D spatial data models, and any actual 3D spatial data model is its subset. Based on the UF-3DSDM, the stratum modeling method of QPTV is further developed on the real borehole sample data. In the context of 3D QTPV model, a data mining method is given on surface DEM of 3D data.

1 Review of 3D Spatial Data Models

Currently, the core problem existing in 3D GIS are data structure and modeling method of 3D spatial data model. In the past decades, there are many 3D data models and data structures which have been investigated. Molenaar proposed 3D Format Data Structure (3D FDS) [1]. Li Rongxin proposed an integrated 3D GIS system based on many kinds of representation, in which spatial objects are described by various data models, such as CSG, TIN, B-Rep, Octree, and CSG+Octree mixed data model [2]. Pilout M and Chen X have made research on Tetrahedral Network (TEN) model [3]. Shi studied a hybrid data model based on TIN and Octree [4]. Li presented a hybrid data model based on Octree and TEN [5]. Gong presented an object-oriented 3D spatial data model integrated vector and raster data [6]. Zlatanova presented a Simplified Spatial Data Model (SSM) [7]. In recent years, some scholars have investigated 3D data model based on Quasi Tri-Prism Volume (QTPV) [8],[9].

In fact, the research about 3D spatial data model is further than above-mentioned models. According to the basic element of models and method of spatial modeling, 3D spatial data model may be classified as facial model, volumetric model and hybrid model. In the view of data description, it can be divided into vector, raster and integrated vector and raster. According to the above-mentioned categorized systems, we can classify some typical 3D spatial data models, as table 1 shows.

Facial model emphasizes particularly on the surface description of 3D spatial and it is convenient for visualization and data updating, but difficult for spatial analysis. Volumetric model based on the 3D spatial partition and description of real 3D entities, emphasizes on representation both of border and interior of 3D objects, with the disadvantage of taking up too much memory space and being slow in calculation. In fact, since none model can be used to describe all spatial entities exactly, so develop-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 785 – 791, 2005.

ing of hybrid data model or combining several different data structures together have become an important subject in terms of 3D data model investigation. It is impossible to design one kind of data model or data structure to satisfy all applications. Special 3D spatial data models should be designed according to distributed characteristics of the spatial entities in the research field.

Table 1. Classification of 3D spatial data model

	Facial Model	*Volumetric Model*		*Hybrid Model*
		Regular Volume	*Irregular Volume*	
Vector	*TIN, B-Rep, Wire Frame or Linked Slices, Section, Section-TIN, Multi-DEMs (TIN)*	*CSG*	*TEN, QTPV, GeoCellular, Irregular Block, Solid, 3D Voronoi diagram*	*TIN-CSG*
Raster	*Grid, Multi-DEMs (Raster)*	*Voxel,Octree Regular Block*		
Integrated vector and raster		*Needle*		*Octree-TEN, Wire Frame-Block, TIN-Octree*

2 A Uniform Framework of 3D Spatial Data Model

Although various kinds of 3D spatial data models have differences in model object components, data structures describing objects, model application purpose, etc., we can find their generality and relation from various kinds of model object components, thus a uniform framework of 3D spatial data model can be designed formally. The so-called uniform framework of 3D spatial data model is a formalized model, on the basis of considering the existing 3D data model synthetically, adopting object-oriented method and hybrid data model form, and integrating various spatial data models.

In the view of object-oriented, it can be decomposed to four basic objects, point, line, surface and body object, and complex object congregated by the four basic objects. At the same time a high-level super-class, i.e. spatial objects class, can be abstracted from these five objects. Spatial objects class appends a public attribute, which will be inherited by its five subclasses. Point object includes isolated point and point with topological relation. Line object includes line with topological relation, line without topological relation and borderline of a surface. Surface object has regular surface, irregular surface, TIN, regular grid, etc. Body object includes regular volume, tetrahedron, tri-prism volume, column constructed by a series continuous section plane, irregular body closed by irregular border surface, and regular geometry established by CSG, B-Rep, etc. Based on the above-mentioned factors, we can design a Uniform Framework of 3D Spatial Data Model (UF-3DSDM). It shows in Fig. 1.

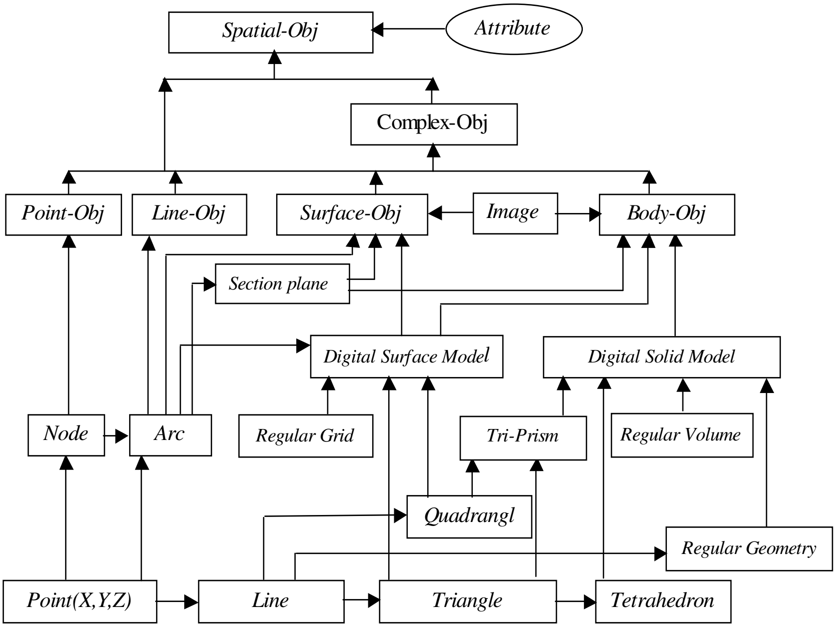


Fig. 1. A uniform framework of 3D spatial data model

UF-3DSDM, only a formalized description of 3D spatial data model framework rather than a concrete model, synthesizes comprehensive characteristics of different models, striving for considering and covering various kinds of spatial data models and their elements, and presents the relation between different spatial data models and elements correctly. In that case, from the point of aggregate, UF-3DSDM is a union of various 3D spatial data models, but a concrete 3D spatial data model in practical application is a subset of the UF-3DSDM. For example, in the TIN model describing a terrain surface, its geometry elements, such as point, line, triangular, digital surface model, surface object and image, are all contained in the uniform framework.

3 Model Instance of UF-3DSDM

According to the UF-3DSDM, different real 3D data models can be extracted. This paper proposed a hybrid data model based on Multi-DEM and QTPV. The so-called multi-DEM model is a stratum model, which sews up the DEM of stratum boundary sequentially from the earth's surface to underground. Obviously, this kind of model is an empty shell, namely a facial model. However, in the geological exploration domain, mineral deposit, considered as non-homogeneous, needs to be expressed by real 3D model. In that case, volumetric model need to be established. In order to integrate the volumetric model with multi-DEM adequately and effectively, it is an ideal choice to adopt the QTPV as the basic volume element. In the hybrid 3D data model based on multi-DEM and QTPV, DEM whose basic element is triangle, is composed of a series of adjoining triangles adjacent to each other. The geometric elements contains in QTPV are vertices, segments (edges, triangular sides), triangles, side quadrangle and tri-prism volume [8]. Therefore vertices, segments, triangles, side quadrangle, tri-prism volume and DEM are regarded as the basic elements of this model. Thus we can design hybrid data model based on multi-DEM and QTPV by using UML. See Fig. 2.

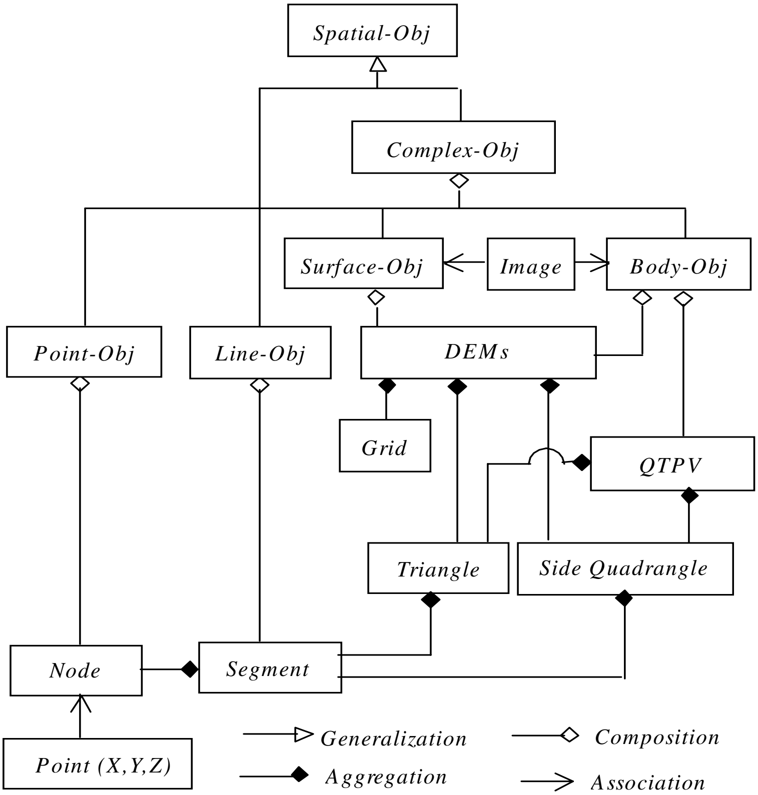


Fig. 2. Hybrid 3D Spatial Model Based on Multi-DEM and QTPV

4 Modeling of QTPV in Stratum

Let the data structure of a borehole curve point consist of 3D coordinates and an adjacent attribute code. The adjacent attribute code, numbered in increasing order from the earth's surface to the subsurface, is the down adjacent geological attribute of a point. The main steps in the modeling are as follows:

(1) Create a triangle (up-triangle) by using the methods of constructing a Delaunay TIN according to the borehole location points on the earth's surface. This triangle is the up-triangle of a QTPV, and the vertices of this triangle correspond to three boreholes.

(2) Expand a new triangle (down-triangle) down along the three boreholes according to the adjacent attribute code of the up-triangle points, as shown in Figure 3. If their codes are the same, the new triangle points are the next points along the boreholes, as Figure 6 (1) shows. If their codes are different, in the borehole with a smaller code (for example a), the new triangle points are the next points along the boreholes. In the borehole with a larger code (for example b or c), the new triangle points will not change, as Figure 6 (2) ~ (4) shows.

(3) Construct a QTPV according to the up-triangle and down-triangle. Then change the down-triangle into the up-triangle.

(4) Repeat steps (2) and (3) until all the up-triangle points are at the bottom of the three boreholes.

(5) Expand the triangle along the triangle side on the earth's surface by using the methods of constructing a Delaunay TIN. Repeat steps (2) ~ (4) and construct all of the QTPVs.

(6) If all of the points are constructed into the triangle, stop the modeling process. Otherwise, go to step (5).

After above processing steps, QPTVs are constructed. The multi-DEMs (i.e. interface) of stratigraphy objects can be picked up by using an algorithm after the QTPVs model is constructed. The algorithm theory is that to find all of the adjacent triangles with the same positive-negative attribute features with different positive side and negative side attributes in one triangle.

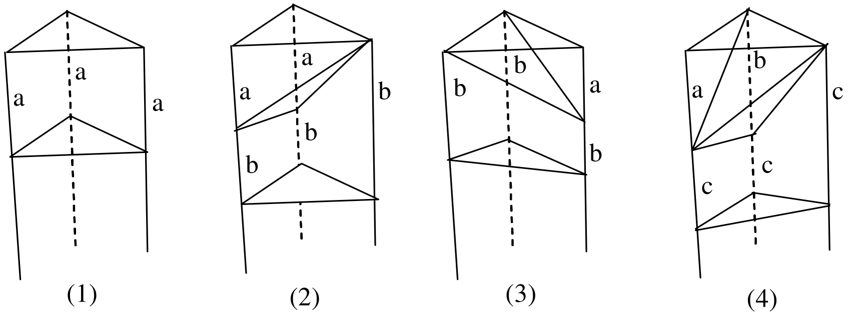


Fig. 3. The down-expansion of QTPV

5 Surface TIN Data Mining from 3D QTPV Model

For getting a hybrid data model from the QTPV model, the stratum interface information, representated by DEM, should be mined from the QTPV model sometimes. Supposing that the five face elments of QTPV have the negative and positive face attribute in each side, then the idea of extracting surface is that searching the ajacent triangles which have the same negative and positive face attribute but the negative attribute is different from the positive attribute in every face. The main steps of mining the surface information are as follows:

(1) Create a stack S and a file DF used for storing all the triangles constructing the surface TIN model.

(2) Select an arbitrary triangle Ts locating on the surface and record its negative and positive face attribute. Attach a search flag to triangle Ts, i.e. $(Ts)_f$= TRUE , then input Ts into DF file and push it into stack S too.

(3) Check the stack S to see whether it is empty? If S is empty then the search work of the surface TIN model is done and all the triangles in DF file are the final results, and stop the search process.

(4) If S has element still, then pop a triangle Tp and find the first adjacent triangle T of Tp.

(5) Check the search flag T_f of T, if T_f =TRUE then go to step (6). Otherwise, that is T_f=FALSE and let T_f =TRUE. Then comparing the negative and positive face attribute of T_f with T_s', if they are the same then push the triangle T into DF file and the same time push it into S, otherwise go to step (6).

(6) Check T_p and to see whether it still has adjacent triangle which has not been processed by step (5)? If it has, then go to step (5), otherwise go to step (3).

After the above steps having been done, the triangles in file DF are the surface TIN model mined from QTPV model.

6 Remarks

In 3D GIS research domain, real 3D data model need to be investigated. A special 3D spatial data model should be designed according to distributed characteristic of the spatial entities in the research field. The proposed UF-3DSDM is only formalized description of 3D spatial data model framework rather than a concrete model. UF-3DSDM is a union of various 3D spatial data models, and any actual 3D spatial data model such as data model based on QTPV, hybrid data model based on Multi-DEM and QTPV, and so on, is a subset of the UF-3DSDM. Worth mentioning that hybrid model based on Multi-DEM and QTPV is suitable for spatial entities modeling in geological exploration engineering field especially. The reason is that it has a capability of modeling stratigraphy interface and mineral deposit at the same time, furthermore, QTPVs data model not only overcome the strict data restriction, i.e. the captured points should be located on a regular 3D grid, but also conquer the disadvantages of TEN, such as huge data volume, complex topological relationship and modeling algorithm complexity.

Data mining among the 3D spatial data model should be researched so as to get rich and deep information, and to transform models each other. In this paper, modeling method of QPTV model of stratum according to the real borehole sample data was proposed firstly, and then surface DEM data mining from 3D QTPV model is put forward. These two methods were realized in our experimental system prototype 3DGeoMV. The result shows that a facial model could be mined from a volumetric model, vice versa. Make research of 3D spatial data mining from different 3D spatial data model is worthy in the future.

Acknowledgement

This research paper is funded by the National Key Fundamental Research Prophase Project of China (2003CCB00600), the Open Research Fund Program of Digital Land Key Laboratory of Jiangxi (DLLJ200501).

References

1. Molanaar, M.: A topology for 3D vector maps. ITC Journal. 1(1992)25-33
2. Li, R.: Date Structure and Application Issues in 3D Geographic Information System. Geomatics. 3(1994)209-244.
3. Pilout, M., Tempfli, K., Molenaar, M.: A tetrahedron-based on 3D vector data model for geoinformation. In: Molennar, M., S. de Hoop, (Eds): Advanced geographic data modeling. Netherlands Geodetic Commission, Publication on Geodesy, Vol.40. Delft, The Netherlands (1994)129-140
4. Shi, W. Z.: A hybrid model for 3D GIS. Geoimformatics. 1 (1996) 400-409
5. Li, D., Li, Q.: A study on hybrid data structure in 3D GIS. Acta Geodatica et Cartographica Sinica, 2(1997) 128-133 (in Chinese)
6. Gong, J., Xia, Z.: An integrated data model in three-dimensional. Journal of Wuhan Technical University of Surveying and Mapping. 1 (1997)7-15 (in Chinese)
7. Zlatanova, S., Rahman, A., and Shi, W., Topological models and frameworks for 3D spatial objects. Computers & Geosciences. 4(2004) 419-428
8. Gong, J., Cheng, P.: Three-dimensional modeling and application in geological exploration engineering. Computers & Geosciences. 4 (2004)391-404
9. Wang, S.L., et al.: A try for handling uncertainties in spatial data mining. Lecture Notes in Artificial Intelligence, Vol. 3215. Springer, Berlin (2004) 513-520

Mining Standard Land Price with Tension Spline Function

Hanning Yuan[1], Wenzhong Shi[2], and Jiabing Sun[1]

[1] School of Remote Sensing Information Engineering,
Wuhan University, 430079 China
[2] Department of Land Surveying and Geo-Informatics,
The Hong Kong Polytechnic University, Hung Hom, Kowloon,
Hong Kong SAR, China
yhn1979yhn@163.com

Abstract. Standard land price is an economical indicator for measuring land value. In this paper, we propose to use the tension spline interpolation function to mine standard land price. First, we extend the definition of standard land price, which is based on land region composed of several neighboring land parcels with the same or similar features. Second, the regional factors that affect the standard land price are classified into the geometric features of point, line and area according to the quantitative rules. Third, a tension spline interpolation function is proposed to mine standard land price, which is determined by the influential factors. Finally, as a case study, the proposed method is applied to mine land prices for Nanning City in China. The case study shows that the proposed method is a practical and satisfactory one.

1 Introduction

Data mining is to extract previously unknown but potentially useful knowledge from the large amount of databases, and it can assist in the governmental, local, organizational and personal decision-making [1]. Land plays a major role in the provision of housing and the social and infrastructural needs of the community. However, it is endangered by such great crises as decreasing plantation, increasing building land, desertification, illegal land transactions in land markets and so on [2]. Standard land price is an economical indicator for measuring land value, and it helps a government to determine whether to exercise its prior power of buying the land parcel at a price much lower or higher than market price. At the same time, land sellers may draw up the base price of a land parcel for bidding, and land buyers can calculate their investment profit before the land transaction according to the standard land price [3]. In order to save and protect land during its disposal, acquistion and management, the government, land sellers and land buyers are all interested in knowing the standard land price, and furthermore, in controlling the price.

Standard land price may be mined via the association between influential factors and price samples. Standard land price is affected by many factors to different degrees [4],[5]. When there are no or insufficient price samples, the standard land price of a

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 792 – 803, 2005.

land region has to be deduced from a functional association between the influential factors and the land price samples. In fact, it is almost impossible to obtain the true functional relationship completely, and an approximate function is had to be selected to approach it.

The least square is applied for measuring residential property values [6]. However, the least square gives an overall prediction function within the whole interval. Furthermore, its predicting function may not pass every sample tuple, which is called a node. Least square is unsuitable for a functional curve with many curvature changes. When the number of sample tuples is very small, the least square estimation cannot preserve the functional smoothness, thus leading to bigger estimated residuals. By contrary, the tension spline interpolation function piecewise approaches the true functional relationship subinterval by subinterval within a given overall interval. Furthermore, it passes all the nodes, which makes the precision of this estimation higher. Moreover, its tension coefficient can adjust the curved shape of the interpolation function. With the various tension coefficients, we may have different results. This allows people to optimize the discovery of land price. The estimated residuals, which are the difference between the predicted value and the true value, are also smaller. Therefore, in this paper, we focus on a discovery of standard land price by using the tension spline interpolation function.

2 Data Preprocessing

The standard land price is defined as the average price of a land region in this paper. Based on the rules for quantifying the factors influencing land price, the sum influential value of a land region can be computed. An average of all the price samples within a land region is an observed value of the standard land price. In other words, each of the land regions has a tuple (influential value, standard land price). Land price samples and the influential factors are usually from the central districts of a city, while there are fewer or no samples of land prices in some places, especial the marginal areas. According to the tuples of land regions with enough price samples, we can derive the relationship between the influential value and the standard land price of a land region. Based on this, an unknown standard land price of the land region with inadequate or no price samples can be calculated with its influential value.

2.1 Standard Land Price

Traditionally, the standard land price is a grade price for the whole city, for example, the land of Nanning City was divided into six grades in 1994. If the standard land price of a land parcel needs to be updated, that for the whole city will have to be updated accordingly. Now, we propose a regional standard land price for a land region. A city is divided into a set of land regions, which are composed of groups of neighboring land parcels with the same or similar features. For example, in Fig. 1, a city is divided into a set of land regions. When the standard land price of a land parcel is out-of-date, we can only update it from the land region to which it belongs. This might be the most efficient way. The standard land price of a land region is mined with the association between influential value and the corresponding price sample.

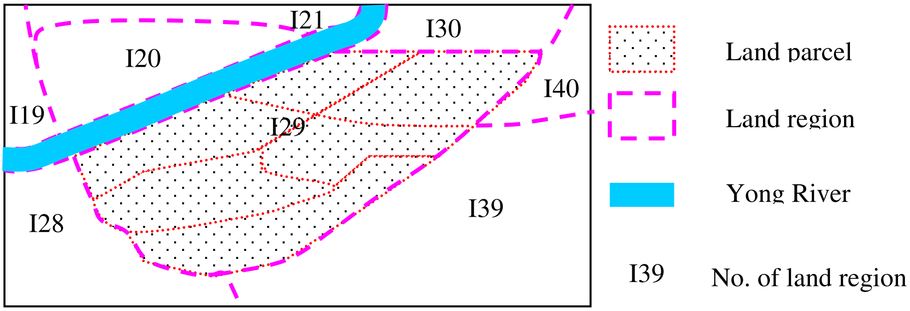


Fig. 1. Land parcel and land region of a city (part)

The standard land price is mined according to the regional factors of these land parcels in terms of the average capacity ratio, average land use density, a fixed number of years, a mining date, etc. Price sample is the latest price of land transaction that has taken place before the discovery. However, in a city, there are always some land regions with inadequate or no price samples because not all land parcels in a land region have enough price samples; some may not even have land price samples, or may not obtain a satisfactory land price sample. We mine the standard land price in one land region with inadequate or no price samples on the basis of the other land regions with rich price samples, since there is a functional relationship $y = f(x)$ between the standard land price y and its influential value x. In a land region with rich price samples, y is the average of the price samples. Each land region has a tuple (x,y). According to the distribution of the tuples in a coordinate system, an approximate relationship function for each region can be determined.

2.2 Influential Factors

Multiple factors affect standard land price to different degrees. However, it is impossible to consider all the factors. Therefore, influential factors are selected and weighted with the integration of the Delphi method and the analysis hierarchy process. The Delphi method is employed to select influential factors and decide their primary weights, then the analysis hierarchy process is used to determine their weights. Their weights formalize a weight vector $W = (w_1, w_2, \ldots, w_m)^T$. Fig. 4 gives such an example result of the influential factors and their weights on industrial land.

In contrast to the individual revising factor, the influential factor is regional. All influential factor information should be complete, impersonal and wide. First, it is collected from each department concerned with standard land price discovery. Second, it is necessary to eliminate obvious errors and check their integrality, dependability and unit coherence, etc., so as to obtain the reasonable influential values according to the quantizing rules. For example, a small change of the standardize capacity ratio may lead to a big fluctuation of land price.

2.3 Computation of Influential Values in a Mining Grid Context

A mining grid is developed in our computerized system. As it is difficult to compute the influential values for each point, the city is divided into a series of grids for discovery as the image rasters in the system (see Fig. 2). The influential value decreases as the relative distance between an influential factor and a grid is increased [7]. Thus it is necessary to compute the relative distance *r*.

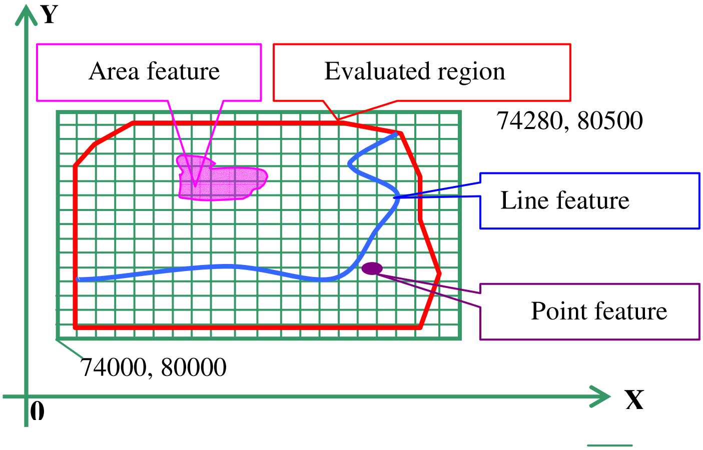


Fig. 2. Computing influential values in the grid context

$$r = \begin{cases} d / R & (d < R) \\ 1 & (d \geq R) \end{cases} \qquad (1)$$

where *d* is a true distance from the factor to a grid point and *R* is the maximum influential radius (distance) of a factor. Note for a point feature, $R = [A / (n \times \pi)]^{½}$, *A* is an urban area, *n* is the amount of factors; for a line feature, $R = A / 2 \times L$, *L* is the length of a line feature.

Primary influential value *k* is used to quantify influential factors. *k* is within the interval [0,100]. 100 denotes the most influential, while 0 denotes no influence. For example, road density has five such grades: biggest, bigger, medium, smaller and smallest. The values of the corresponding primary influential value *k* are 100, 90, 70, 50 and 30 respectively. When an influential feature is quantitative,

$$k_i = 100 \times (a_i / a_{max}) \qquad (2)$$

where k_i is the primary influential value of sample *i*, a_i is its true value and a_{max} is the maximum of a_i. If an influential feature a_i is composed of several atomic parts g_j (j = 1, 2, …, n), then $a_i = (\Sigma g_i) / n$, $k_i = 100 \times [(\Sigma g_i) / n] / max[(\Sigma g_i) / n]$. For example, a middle school is divided into land area (g_1), class number (g_2), student number (g_3) and teacher number (g_4). The influential index of the middle school *i* is $k_i = 100 \times (a_i / a_{max}) = 100 \times [(g_{i1} + g_{i2} + g_{i3} + g_{i4}) / 4] / max\ [(g_{11} + g_{12} + g_{13} + g_{14}) / 4, (g_{21} + g_{22} + g_{23} + g_{24}) / 4, \ldots, (g_{n1} + g_{n2} + g_{n3} + g_{n4}) / 4]$.

The influential factors are classified into the geometric features of point, line and area according to the computation rules. A single influential value is obtained based on the following rules.

- (1) Point features may include, for example, shopping center, depot, dock, hospital, school, post office, park, theater, gymnasium, convenient degree of public transportation, etc. The single influential value f_i of a point feature i affecting a grid point is

$$f_i = k_i \times (1 - r) \tag{3}$$

However, there is a difference to calculate the influential value of a shopping center.

$$f_i = \begin{cases} (k_i - k_j)^{1-r} & (k_i > k_j, r < 1) \\ 0 & (r \geq 1) \end{cases} \tag{4}$$

- (2) Linear features point to road degree, road planning, river and so on. To a grid point, the single influential value f_i of a linear feature i is

$$f_i = \begin{cases} k_i^{(1-r)} & (r < 1) \\ 0 & (r \geq 1) \end{cases} \tag{5}$$

- (3) Area features may include, for example, air pollution, noise pollution, land planning, water supply, power supply, drainage, engineering geology, water flood and log, intensive degree, etc., and their influences may be uniformly distributed in the land mining area. Their single influential values on a grid point can be calculated via equation (6).

$$f_i = 100 \times \frac{x_i - x_{\min}}{x_{\max} - x_{\min}} \tag{6}$$

where x_i is the true value of factor i. $\mathrm{x}_{\max}$ is the maximum of x_i and $\mathrm{x}_{\min}$ is the minimum. In particular, $\mathrm{x}_{\max}$ of the population factor is the optimal density value.

Based on the single influential value, the sum influential value can be computed. For a mined grid, the factors make an influential vector, $F = (f_1, f_2, \ldots, f_m)$, each measure of which has a different weight on the standard land price. The weight vector $W = (w_1, w_2, \ldots, w_m)^T$ has been obtained before. The sum influential value of a grid is then

$$gs = FW / (\Sigma w_i) = (\Sigma f_i w_i) / (\Sigma w_i) \tag{7}$$

where, s is the sum influential value of a mining grid. If $\Sigma \mathrm{w}_\mathrm{i} = 1$, then $gs = \Sigma f_i w_i$. There are many mining grids in a land parcel. The sum of these grids' sum influential values is the land parcel's one $ps = \Sigma gs_i$. A land region includes several land parcels whose sum influential values formalize a vector $PS = (ps_1, ps_2, \ldots, ps_n)$. Their mean is the sum influential value of the land region, namely, $x = (\Sigma ps_i)/n$.

2.4 Price Samples

A price sample is an example of land price in a land transaction. It is the most up-to-date price from land transactions in the land marke. In a land region, the more the price samples exist, the more accurately the standard land price can be calculated. An average of all the price samples within a land region can be called an observed value of standard land price.

As with the collection of influential factor information, price samples should also be complete, impersonal and wide. We first collect the existing land price from each department concerned with land transactions. Land transactions happen when land is transferred, sold, rented, exchanged, mortgaged, denoted or inherited [13], [1]. Land price samples can also be extracted from real estate transactions. But not all land parcels have price samples.

The standard land price of a land region with sufficient land price samples can be computed according to the price samples. Each of the land regions has a tuple (influential value, standard land price). According to the tuples of land regions with enough price samples, we can derive the functional relationship between the influential value and the standard land price of a land region. Based on this, an unknown standard land price of the land region with inadequate or no price samples can be calculated with its influential value, such as land regions in the city boundary. Therefore, the functional relationship needs to be determined on the basis of the influential factors and land price samples.

3 Tension Spline Interpolation

The tension spline interpolation function piecewise approaches the true functional relationship subinterval by subinterval within a given interval. It passes all the sample tuples. Even if there are only three sample tuples, the tension spline interpolation function can still be used with high precision.

The tension spline interpolation function is proposed to mine the standard land price of a land region with inadequate or no price samples. It is known that in a city, there are always some land regions with inadequate or no price samples, because not all land parcels in a region have price samples. However, we can compute sum influential values for a whole city as in the abovementioned section 3. Those standard land prices can then be discovered with the functional relationship $y = f(x)$. In a land region, x is the sum influential value, and y is the average of their price samples. In other words, each land region has a tuple (x,y). According to the distribution of (x_i, y_i) $(i= 0, 1, 2, ..., n)$ in a coordinate system, an approximate function for each region can be determined if we have enough samples. However, the distribution usually shows various curvatures within different subintervals $[x_i, x_{i+1}]$. Thus, we propose the tension spline interpolation function to piecewise approach $y = f(x)$.

Here, we briefly describe the basic principle of the tension spline interpolation function. Suppose there are $n+1$ interpolation nodes: P_0, P_1, …, P_n. The coordinates of node P_i ($i = 0, 1, \dots, n$) are (x_i, y_i), and $x_i < x_{i+1}$($i = 0, 1, 2, \dots, n$-1). That is to say, $y_i = f(x_i)$ is a single valued function (Zhu [14]). In order to compute the approximate function of $y = p(x)$, a tension spline interpolation function is defined. For nodes P_0, P_1, ……, P_n, we call the function $y = p(x)$ the tension spline interpolation function if $p(x)$ satisfies the following conditions:

(1) the spline condition, $p(x)$ is a cubic spline interpolation function;

(2) the interpolation conditions, $p(x_i) = y_i$ $(i = 0, 1, \dots, n)$;

(3) the boundary conditions, $p'(x_j) = y_j'$ $(j = 0, n)$; and

(4) the tension condition, $p''(x) - \sigma^2 p(x)$ is continuous within the interval $[x_0, x_n]$.

Furthermore, it is a linear function on each subinterval $[x_{i,}\ x_{i+1}]$ (i = 0, 1, …, n-1). Here σ is the tension coefficient.

According to the four conditions, we can construct the tension spline interpolation function $p(x)$. It is represented as the piecewise function within each subinterval. For the subinterval $[x_{i,}\ x_{i+1}]$ (i = 0, 1, …, n-1), we have

$$p_i(x) = \frac{p_i''(x_i)sh[\sigma(x - x_{i-1})] + p_i''(x_{i-1})sh[\sigma(x_i - x)]}{\sigma^2 sh(\sigma\ h_i)} + \left[y_{i-1} - \frac{p_i''(x_{i-1})}{\sigma^2}\right]\frac{x_i - x}{h_i} + \left[y_i - \frac{p_i''(x_i)}{\sigma^2}\right]\frac{x - x_{i-1}}{h_i} \tag{8}$$

where $h_i = x_i - x_{i-1}$, $sh(x) = (e^x - e^{-x})/2$, $ch(x) = (e^x + e^{-x})/2$.

The undetermined coefficients, $p''_i\ (x_{i-1})$, $p''_i(x_i)$ can be obtained from a tri-diagonal linear equation group (9). It has a set of solutions because its coefficient matrix is a diagonal dominance and non-singular matrix.

$$\begin{bmatrix} b_0 & c_0 & & & \\ a_1 & b_1 & c_1 & & \\ & \cdots & \cdots & \cdots & \\ & & a_{n-1} & b_{n-1} & c_{n-1} \\ & & & a_n & b_n \end{bmatrix} \begin{bmatrix} p_0''(x_0)/\sigma^2 \\ p_1''(x_1)/\sigma^2 \\ \cdots \\ p_{n-1}''(x_{n-1})/\sigma^2 \\ p_n''(x_n)/\sigma^2 \end{bmatrix} = \begin{bmatrix} d_0 \\ d_1 \\ \cdots \\ d_{n-1} \\ d_n \end{bmatrix} \tag{9}$$

where,

$$a_i = \frac{1}{h_i} - \frac{\sigma}{sh(\sigma h_i)}, b_i = \frac{\sigma ch(\sigma h_i)}{sh(\sigma h_i)} - \frac{1}{h_i} + \frac{\sigma ch(\sigma h_{i+1})}{sh(\sigma h_{i+1})} - \frac{1}{h_{i+1}},$$

$$c_i = \frac{1}{h_{i+1}} - \frac{\sigma}{sh(\sigma h_{i+1})}, d_i = \frac{p_{i+1} - p_i}{h_{i+1}} - \frac{p_i - p_{i-1}}{h_i}, (i = 1,2,..., n-1);$$

$$b_0 = \frac{\sigma ch(\sigma h_1)}{sh(\sigma h_1)} - \frac{1}{h_1}, c_0 = \frac{1}{h_1} - \frac{\sigma}{sh(\sigma h_1)}, d_0 = \frac{p_1 - p_0}{h_1} - p_0', (i = 0);$$

$$a_n = \frac{1}{h_n} - \frac{\sigma}{sh(\sigma h_n)}, b_n = \frac{\sigma ch(\sigma h_n)}{sh(\sigma h_n)} - \frac{1}{h_n}, d_n = p_n' - \frac{p_n - p_{n-1}}{h_n}, (i = n).$$

Hence, we can obtain an overall interpolation function for the whole interval $[x_{0,}\ x_n]$. The tension spline interpolation function has a quadratic derivative within the overall interval $[x_{0,}\ x_n]$. Its coefficient σ can adjust the curved shape of the interpolation function. When σ = 0, the tension spline interpolation function is a general cubic spline interpolation function based on the above defined tension condition. The larger the σ, the stronger the tension of the interpolated curve. It is suitable to mine standard land price using a larger σ for the part of land price figure with larger curvature.

In order to test whether the predicted function is suitable, it is suggested to leave some samples. At the same time, the standard land price of a city is often compared with that of other similar cities so as to ensure that the land price matches the level of the city to be mined.

4 A Case Study

The proposed method was applied in Nanning City, China, and industrial land was taken as an example for the study. In this section, the case study was presented in such subsections as data collection and computation, result and analysis.

4.1 Data Collection and Computation

This city was divided into 46 land regions. We selected the influential factors by the Delphi method, and determined their weights by both the Delphi method and the analytical hierarchy process. Fig. 3 illustrates the result.

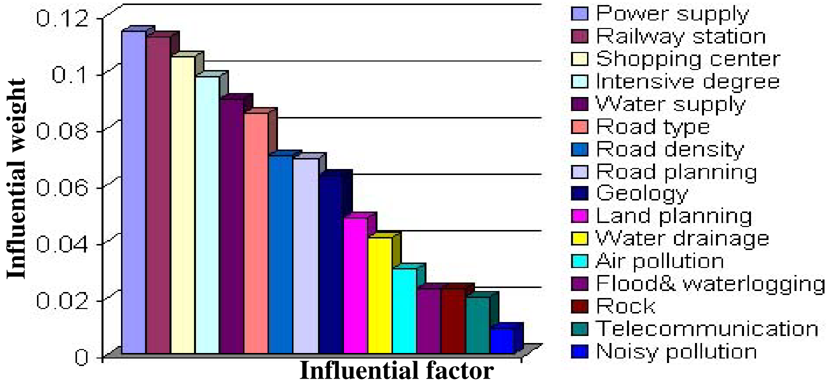


Fig. 3. Figure of influential factors and their weights (∑weights = 1)

Information on all influential factors was collected widely and completely, covering each relevant department. The acquired are various, for example, spatial, social, economics, and business, the amount of which is 10 Giga bytes. We also collected 5027 price samples, most of which were gathered in 39 land regions. First, we cleaned all of the data, checked their integrality, dependability, unit coherence, etc., and then eliminated obvious errors. After 964 price samples had been eliminated, 4063 were left. The city was then divided into a series of mining grids in the land price mining system, and the influential value was computed according to section 3.3.

Putting all 39 tuples (x, y) into the vertical coordinate system, we obtained Fig. 4. As revealed by the figure, the samples show various distributing curvatures within different subintervals. Hence, we selected 32 land regions (Table 2) with richer price samples to piecewise approach the functional relationship with equation(8) on the tension spline interpolation function. The other seven were remained to test its practicality. We computed four times with tension coefficients σ = 0.01, 0.5, 1, 2, 3, and found out that the minimum residual was σ = 2, whose functional curve is shown in Fig. 5.

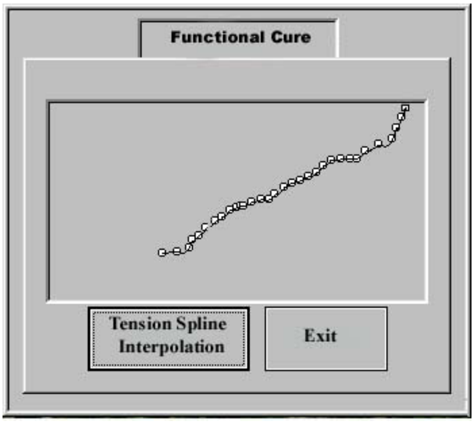


Fig. 4. Functional curve of tension spline interpolation (σ =2)

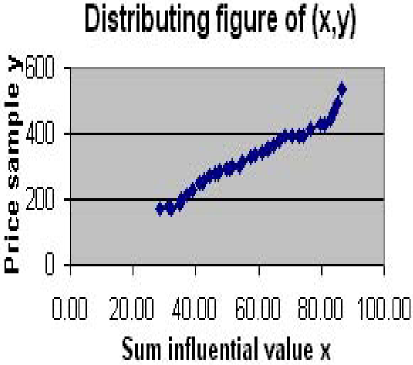


Fig. 5. Distributing figure

By applying the function to test the remaining seven land regions, the tension spline interpolation function with σ = 2 was also practical. Equation (8) could be used to mine the standard land prices of the seven land regions with inadequate or no price samples. The results of both test and discovery are listed in Table 3. *y'* is the mined standard land price in the context of the tension spline interpolation function. The positive value denotes that the mined value is smaller than the observed value from price samples, while a negative value indicates that is bigger.

Table 1. Price samples of industrial land (Unit of land price: *RMB/m*^{2})

No.	*x*	*y*	*No.*	*x*	*y*	*No.*	*x*	*y*	*No.*	*x*	*y*
I01	28.50	172	I17	44.48	274	I27	57.30	328	I37	72.71	394
I03	32.14	174	I18	45.94	278	I28	59.04	338	I38	74.39	395
I05	34.76	184	I19	46.90	282	I30	61.07	344	I39	76.40	412
I07	35.50	203	I20	47.62	283	I31	62.80	352	I40	79.60	428
I08	36.96	211	I21	49.67	291	I34	64.80	362	I43	82.70	443
I11	38.81	232	I23	51.80	298	I26	66.50	379	I44	83.50	467
I13	41.07	247	I24	53.70	297	I35	68.20	391	I45	84.90	494
I15	42.60	256	I25	54.94	311	I36	70.61	393	I46	86.40	536

Table 2. Tested results and mined results (σ =2 Unit of land price: *RMB/m*^{2})

Tested results					mined results		
No.	*x*	*y*	*y'*	*v=y-y'*	*No.*	*x*	*y'*
I02	31.21	176	174.5	1.5	I04	33.80	189.3
I10	37.86	220	218.0	2.0	I06	35.41	197.0
I14	41.97	250	250.1	-0.1	I09	37.30	213.1
I22	50.49	295	296.6	-1.6	I12	40.06	235.5
I33	56.00	314	312.6	1.4	I16	43.93	265.6
I32	63.10	354	353.0	1.0	I29	60.87	347.2
I41	81.00	430	428.4	1.6	I42	85.90	519.1

4.2 Results Analysis

The results were satisfactory. By looking at the whole results, we found that the standard land price for an industrial area changes with its influential factor. I46 was in the city center, and its price was the highest. I45 ~ I01 went from the center to the boundary one by one, and their prices also decreased one by one. I01, whose price was the lowest, was at the city's edge. This obeyed the land price rule. The fact of higher land prices in the center would make industry move to the edge of this city. As a result, this would benefit the citizens' lives in terms of environmental conditions. In order to check whether the standard land price matches Nanning City's level, the result was compared with other cities. Nanning City is the capital of Guangxi Zhuang Autonomous Region, and the most developed of the five capitals of China's autonomous regions. By Chinese standards, it is a medium-sized city. Generally speaking, the contrasting results were that Nanning's standard land price was the highest in Guangxi and in all the capitals of the Chinese autonomous regions. Of all the big cities in China, Nanning's standard land price was average.

This result was compared with the least square in terms of the functional curve (Fig. 7) and test samples (Table 4). After $y = ax^2 + bx + c$, $y = ax + c$ and $y = \mathrm{a} + bc^x$ had been tried, we found that $y = ax + c$ matched the distributing figure of sample (x,y) best. $y = ax + c$ was then selected for the least square. We put both the two functional curves of the tension spline interpolation function (σ =2) and the least square into the same vertical coordinate system. Fig. 6 and Table 4 were thus obtained.

In Table 4, *y'*, *y''* are respectively the mined standard land price in the contexts of the tension spline interpolation function and the least square. A positive value denotes that the mined value is smaller than the observed value from price samples, while a negative value indicates that it is bigger. As Fig. 6 and Table 4 reveal, the least square gives a total statistical function within an interval. It is good where price samples are many, but bad where price samples are few. The tension spline interpolation function piecewise approached the true function subinterval by subinterval. It went through each sample tuple, and its functional curve was almost the true functional curve of the standard land price. Their residual errors also indicat this conclusion, $\delta_{\text{tension spline interpolation function } (\sigma=2)} = 0.5 << \delta_{\text{least square}} = 4.0$. The reasons are that, within a given overall interval, the tension spline interpolation function piecewise approached the true function subinterval by subinterval, and its mined functional curve went through each sample tuple; the least square on the other hand only described a whole statistical rule, and its mined functional curve did not go through each sample.

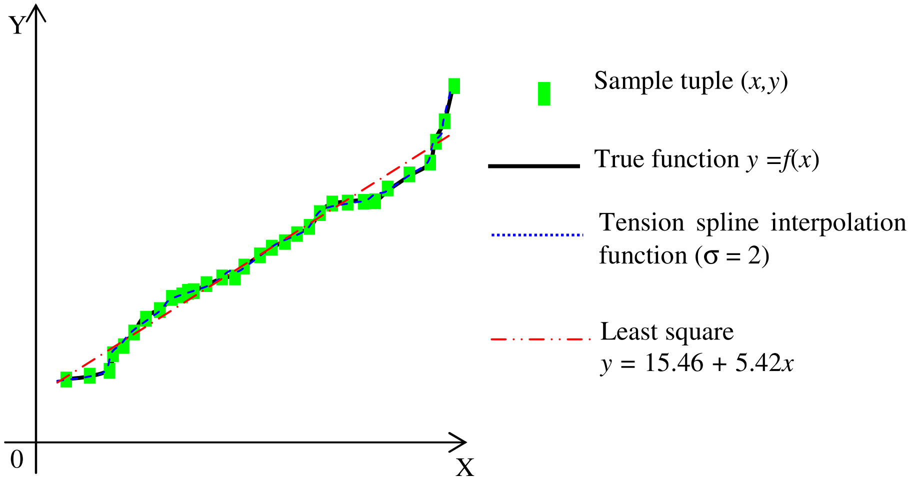


Fig. 6. Mine standard land price of industrial land

Table 3. Comparable test between the tension spline interpolation function (σ =2) and the least square (Unit of land price: *RMB/m*^{2})

Sample tuple (x,y)			Tension spline interpolation function		Least square	
No.	*X*	*y*	*y'*	*v=y-y'*	*Y''*	*v=y-y''*
I02	31.21	176	174.5	1.5	184.6	-8.6
I10	37.86	220	218.0	2.0	220.7	-0.7
I14	41.97	250	250.1	-0.1	243.0	7.0
I22	50.49	295	296.6	-1.6	289.2	5.8
I33	56.00	314	312.6	1.4	319.0	-5.0
I32	63.10	354	353.0	1.0	357.5	-3.5
I41	81.00	430	428.4	1.6	454.5	-24.5
$\delta = [(\sum v^2)^{1/2}] / n$				0.5		4.0

5 Conclusions

Based on the extended definition of standard land price, the tension spline interpolation function was proposed to mine standard land price in the context of a series of grids for mining land price.

The tension spline interpolation function was proposed to mine the standard land price of a land region with relatively fewer or no price samples, which is determined by the influential factors. It piecewise approaches the true functional relationship subinterval by subinterval and has quadratic derivative within the overall interval. The coefficient σ can adjust the curve shape of the interpolation function. The larger the σ, the stronger the tension of the interpolated curve. It is suitable for estimating the standard land price using a larger σ for the part of land price figure with larger curvature.

As a case study, the proposed method was applied to mine the land price for Nanning City. The results were tested by other price samples and compared with the prices of land in other similar cities. Both indicated that the standard land price was very close to its real estate market and matched the city's level. The case study indicated that the proposed method is a feasible one and the tension spline interpolation function is suitable for mining land price.

Acknowledgements

The work described in this paper was supported by the funds from National High Technology R&D Program (863) (Project No.: 2001AA135081), China Postdoctoral Science Foundation (2004035360), Wuhan University (216-276081), and the CRC scheme, Research Grant Council of The Hong Kong SAR (Project No.: 3_ZB40) and The Hong Kong Polytechnic University (Project No.: 1.34.9709).

References

1. Wang, S.L., et al.: A try for handling uncertainties in spatial data mining. Lecture Notes in Artificial Intelligence, Vol. 3215. Springer, Berlin (2004) 513-520
2. Dale, P. F., Mclaughlin, J. D.: Land Administration. Oxford University Press Inc, New York (1999)
3. Zhang, X. Q.: Urban land reform in China. Land Use Policy, 14 (1997) 187-199
4. Colwell, P. F., Munneke H. J.: The structure of urban land prices. Journal of Urban Economics. 41 (1997) 321-336
5. Wang, S.L., et al.: Rough spatial interpretation. Lecture Notes in Artificial Intelligence, Vol. 3066. Springer, Berlin (2004) 435-444
6. Tse, Y.C. R., Love, E.D.P.: Measuring residential property values in Hong Kong. *Property Management*, 18 (2000) 366-374
7. Smersh, G. T., Smith, M. T.: Accessibility changes and urban house price appreciation: a constrained optimization approach to determining distance effects. Journal of Housing Economics 9 (2000)187-196
8. Zhu, C. Q.: Arithmetic and its Application in Surveying & Mapping. Press of Surveying and Mapping, Peking (1997)

Mining Recent Frequent Itemsets in Data Streams by Radioactively Attenuating Strategy*

Lifeng Jia, Zhe Wang, Chunguang Zhou, and Xiujuan Xu

College of Computer Science, Jilin University,
Key Laboratory of Symbol Computation and Knowledge Engineering
of the Ministry of Education, Changchun 130012, China
jia_lifeng@hotmail.com cgzhou@jlu.edu.cn

Abstract. We propose a novel approach for mining recent frequent itemsets. The approach has three key contributions. First, it is a single-scan algorithm which utilizes the special property of suffix-trees to guarantee that all frequent itemsets are mined. During the phase of itemset growth it is unnecessary to traverse the suffix-trees which are the data structure for storing the summary information of data. Second, our algorithm adopts a novel method for itemset growth which includes two special kinds of itemset growth operations to avoid generating any candidate itemset. Third, we devise a new regressive strategy from the attenuating phenomenon of radioelement in nature, and apply it into the algorithm to distinguish the influence of latest transactions from that of obsolete transactions. We conduct detailed experiments to evaluate the algorithm. It confirms that the new method has an excellent scalability and the performance illustrates better quality and efficiency.

1 Introduction

Mining frequent itemsets is an essential data mining operation in many data mining problems such as mining association rules, sequential patterns, closed patterns, and maximal pattern. It has been widely studied and applied since the last decade. The problem of mining frequent itemsets in large databases was first proposed by Agrawal *et al.* [1] in 1993. When it comes to the environment of data streams, mining frequent itemsets becomes a challenging problem, because the information in the streaming data is huge and rapidly changing. Consequently, infrequent items and itemsets can become frequent later on and hence cannot be ignored.

A data stream is a continuous, huge, fast changing, infinite sequence of data elements. The nature of streaming data shapes the algorithm which only requires scanning the whole dataset when it is devised to support aggregation queries. In addition, this kind of algorithms usually owns a data structure far smaller than the size of the whole dataset. The first algorithm to on mining frequent itemsets in data streams using

* This work was supported by the Natural Science Foundation of China (Grant No. 60433020) and the Key Science-Technology Project of the National Education Ministry of China (Grant No. 02090).

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 804 – 811, 2005.

the estimation mechanism is the Lossy Counting proposed by Manku and Motwani [2]. It is a single-pass algorithm based on the well-known Apriori property: if any length k pattern is not frequent in the database, its length (k+1) super-patterns can never be frequent. Han et al. [3] developed a FP-tree-based algorithm, called FP-stream, to mine frequent itemsets at multiple time granularities by a novel titled-time windows technique. Ruoming and Agrawal [4] developed a single-scan algorithm to mine frequent itemsets by viewing the streaming data as an indivisible model, instead of breaking it into batches of transactions. All of the above algorithms capture the situation in which the entire data steam is divided into many batches of transactions. All above algorithms only focus on the frequent itemsets over the entire data stream, and overlook the importance and practicability of mining recent frequent itemsets. To find recent frequent itemsets in the data stream, the algorithm not only needs to scan the dataset once, but also must differentiate the information of recently generated transactions from the useless or invalid information of obsolete transactions. Teng *et al.* [5] proposed a regression-based algorithm, called FTP-DS, to mine recent frequent itemsets in the sliding window. Chang *et al.* [6] developed estDec for recent frequent itemsets in the data stream where each transaction has a weight and it decreases with age. Moreover, Chang *et al.* [7] also proposed a single-scan algorithm for recent frequent itemsets based on the estimation mechanism of Lossy Counting algorithm.

The rest of the paper is organized as follows. In Section 2, we present our algorithm by and large, and then we will introduce the suffix-forest for storing the summary information of streaming data, the lattice for storing frequent itemset in the data stream, and the counter sequence and depth first itemset growth method. The introduction and analysis of radioactively attenuating strategy is established in detail in Section 3. In Section 4, we turn to the experiment evaluation. Eventually, we draw the final conclusion of our algorithm in Section 5.

2 MRFISF Algorithm

MRFISF is newly-devised algorithm based on the estimation mechanism [2] and is also batch-processed. To make it understood, we illustrate each main operation of MRFISF algorithm with a simple example. Note that, it is assumed that all the items in data streams were ordered beforehand.

2.1 MRFISF Algorithm

Input: A data stream D, minimal support threshold θ, and maximal estimated support error threshold ε, decaying factor ϕ.

Output: recent frequent itemsets in lattice.
For each batch of transactions in the data stream D:

1. Construct the suffix-forest to store the summary information.
2. Generate frequent itemsets from the suffix-forest by counter sequence and depth first itemset growth method.
3. According to the estimation mechanism, insert new frequent itemsets into lattice or update the frequencies of old frequent itemsets already in lattice.
4. Pruning the infrequent itemsets by now from the lattice.

5. According to the attenuating points, reduce frequencies of itemsets which need attenuating by decaying factor ϕ and Pruning infrequent itemsets due to the attenuation from the lattice..

2.2 Suffix-Forest and Lattice

To mine frequent itemsets in data streams, MRFISF chooses the suffix-forest to store the summary information of data streams. Now, let us turn to describing the steps of constructing a suffix-forest.

For the problem of mining frequent itemsets, the data element in data streams is the transaction composed of items. First, when a new transaction $t=\{I_1,I_2,\ldots,I_n\}$arrives, it is projected into its suffix-sets:$\{\{I_1,I_2,\ldots I_n\},\{I_2,I_3,\ldots I_n\},\ldots,\{I_{n-1},I_n\},\{I_n\}\}$. Second, to store them, suffix-trees are established according to these suffix-sets which are viewed as many different individuals. Third, all of the identical nodes of each suffix-tree are connected by a node link team (NLT for short). We use "suffix-tree(I_i)"and "node(I_k)" to denote the suffix-tree whose root is item I_i and a node I_k of suffix-tree respectively.

Lemma 1: In the suffix-tree(I_i), its sub-tree whose root is I_k (k>i) must be a sub-tree of suffix-tree(I_k).

Proof: During the construction of suffix-tree(I_i), when constructing a certain branch of sub-tree whose root is item I_k, MRFISF algorithm must deal with a certain suffix-set $\{I_i,\ldots,I_k,\ldots,I_n\}$.Meanwhile, the suffix-set$\{I_k,\ldots,I_n\}$must appear together with this suffix-set$\{I_i,\ldots,I_k,\ldots,I_n\}$. Thus, MRFISF either constructs a new branch of suffix-tree(I_k) to store the suffix-set $\{I_k,\ldots,I_n\}$or inserts it into an already existed branch in suffix-tree(I_k) by updating the frequency of nodes in that branch. Consequently, the lemma 1 is proofed

Based on lemma 1, we can draw a conclusion: if a certain node(I_m) of suffix-tree(I_i) has a corresponding node(I_m) in the suffix-tree(I_k), such a node(I_m) must have an ancestor node(I_k) in the suffix-tree(I_i). In order to recognize the corresponding identical nodes in different suffix-trees, we need to code nodes of suffix-tree. Suppose that the nodes of complete suffix-tree are coded by the depth first. MRFISF algorithm endows nodes of the actual (complete or incomplete) suffix-tree the same serial numbers as those they own in the corresponding complete suffix-tree. To make this coding method clear, let us illustrate it with a simple example, a batch of transactions, {a,c,d}, {a,b,d},{b,d},{b,c,d}. The serial numbers of nodes of suffix-tree(a) of above example are endowed on the basis that the complete itemset includes 5 items {a,b,c,d,e}.

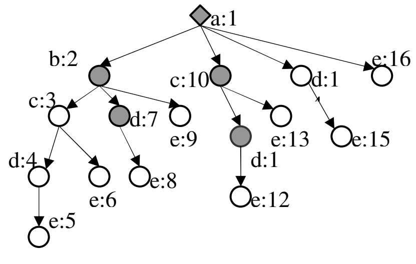


Fig. 1. the complete suffix-tree(a)

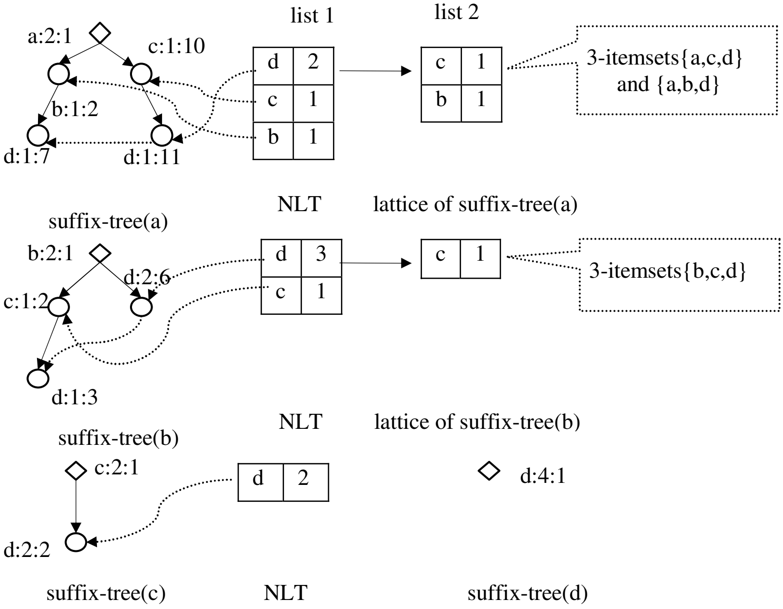


Fig. 2. The suffix-forest and the lattice of frequent itemsets

Fig 1 shows the serial number of each node in the complete suffix-tree(a) by means of depth first. Fig 2 shows the suffix-forest and lattice generated by MRFISF algorithm with above simple example. Note that the instance is so simple that MRFISF do not execute the radioactively attenuating strategy. Every node of the suffix-tree shown in Fig 2 contains three aspects of information: nodename, frequency, and serial number. In addition, the shaded nodes of the complete suffix-tree(a) in Fig 1 are the corresponding nodes of actual suffix-tree(a) in Fig 2. Therefore, same nodes in complete and actual suffix-tree(a) have the identical serial numbers.

2.3 Itemset Growth

According to the method of coding the nodes of suffix-forest, there is a special relationship between the corresponding identical nodes in different suffix-trees: In the suffix-tree(I_i), the difference of the serial number of node(I_n) and that of its ancestor node(I_m) is equal to the serial number of its corresponding node(I_n) in the suffix-tree(I_m). Consequently, we can use this relationship to identify whether a node in a suffix-tree has a corresponding identical node in another suffix-tree. To explain this relationship clearly, let us illustrate it with the suffix-tree(a) in Fig 2. For the node(d) whose serial number is 7, and its ancestor node(b) whose serial number is 2, the difference of serial numbers of the two nodes is (7-2)+1=6. The serial number of its corresponding node(d) in the suffix-tree(b) is 6, which is equal to the difference value. All above phenomena are based on the special relationship.

Lemma 2: In an assumed suffix-tree(I_1) with i kinds of child nodes{I_2 ,...,I_{i+1}}, let $f(I_{i+1})$ and $c(I_{i+1})$ denote the frequency and the serial number of node(I_{i+1}) respectively. For each node(I_{i+1}) of suffix-tree(I_1), if it has a corresponding node(I_{i+1}) in suffix-tree(I_k), the serial number of this node(I_{i+1}) is stored in the serial number set C_k ($1<k<i+1$). Let the total serial number set (TSN-SET for short) $C = C_2 \hbar \ldots \hbar C_i$**,** if $C \neq \varnothing$, the frequency of (i+1)-itemset{I_1,...,I_{i+1}}=$\sum f(I_{i+1})$, such that $c(I_{i+1}) \in C$; Otherwise, the frequency of (i+1)-itemset{I_1,...,I_{i+1}}=0.

Proof: In order to compute the frequency of (i+1)-itemset{I_1 ,..., I_{i+1}}, we need to know all nodes(I_{i+1}) which are the common child nodes of node(I_1),..., node(I_i) in the suffix-tree(I_1). According to the conclusion of lemma 1, if a certain node(I_m) of suffix-tree(I_i) has a corresponding node(I_m) in the suffix-tree(I_k), such a node(I_m) must have an ancestor node(I_k) in the suffix-tree(I_i), and the statement of lemma 2, the set C_k stores serial numbers of nodes(I_{i+1}) which have nodes(I_k) as ancestor nodes and the TSN-SET $C = C_2 \hbar \ldots \hbar C_i$. So, if there is the occurrence of nodes(I_{i+1}) which are the common child nodes of node(I_1),..., node(I_i), the serial numbers of these nodes(I_{i+1}) must appear in the TSN-SET C($C \neq \varnothing$). Subsequently, the frequency of (i+1)-itemset{I_1,..., I_{i+1}} is equal to the sum of frequencies of these nodes(I_{i+1}), $\sum f(I_{i+1})$, such that $c(I_{i+1}) \in C$. Otherwise, if the TSN-SET C is empty($C = \varnothing$), the frequency of (i+1)-itemset{I_1,..., I_{i+1}}must be equal to zero.

Based on the lemma 2, we turn to introducing the new itmeset growth method: counter sequence and depth first itemset growth. However, before detailing it, we first define two core operations as follow:

Definition 1 (Insert Itemset Growth(IIG)): When MRFISF has already computed the frequency of i-itemset{I_1,...,I_m,...,I_k}(k>m>i), through the operation of insert itemset growth, MRFISF will compute the frequency of (i+1)-itemset{I_1,..,I_p,I_m,..,I_k}, such that I_p is newly inserted item(i<p<m<k). For example, the IIG operation makes 3-itemset{c,e,f} grow to 4-itemset{c,d,e,f}

Definition 2 (Replace Itemset Growth(RIG)): When MRFISF has already computed the frequency of i-itemset{I_1,...,I_m,...,I_k}(k>m>i), through the operation of replace itemset growth, MRFISF will compute the frequency of i-itemset{I_1,...,I_p,...,I_k}, such that I_p is newly inserted item to replace the item I_m (i<p<m<k). For example, the RIG operation makes 3-itemset {c,e,f}grow to 3-itemset {c,d,f}.

We stipulate that the priority of IIG operation is higher than that of RIG operation. We also stipulate that the sequence of newly-inserted item by both IIG and RIG operations is according to the counter item sequence. Based on the discussion so far, the counter sequence of inserting items of IIG and RIG operations not only guarantees that all frequent itemsets will be generated by MRFISF algorithm, but also makes sure that all redundant computation will be eliminated during the phase of itemset growth. Now, let us turn to the simple example in Fig 2 again, the c unit in the list 2 of lattice of suffix-tree(a) denotes 3-itemset {a,c,d}, because c unit grows from the d unit which denotes itemset {a,d} by IIG operation. The b unit in the list 2 of lattice of suffix-tree(a) denotes 3-itemset{a,b,d}, because b unit grows from the c unit which denotes itemset {a,c,d} by RIG operation.

3 Radioactively Attenuating Strategy

In nature, the attenuating rule of radioelement is $N = N_0 \cdot 2^{-t/T}$ .In this formula, *N* denotes the number of existing radioelement atoms without attenuating, N_0 denotes the number of radioelement atoms before the attenuation, *t* denotes the attenuating time, and T denotes the half life of the radioelement which is the time the quantitative radioelement atoms need to attenuate to the half number of them.

3.1 Radioactively Attenuating Strategy

Before clearing the details of new strategy, let us introduce two definitions first.

Definition 3 (Lifecycle of a Frequent Itemset): Lifecycle of a frequent itemset is the phase of time which begins when it appears as a frequent one and ends when it is turning to be infrequent.

Definition 4 (The Attenuating Point): The attenuating point of a frequent itemset is the time when $t > 2^p \cdot t_0$ and $t \geq 2^{P+1} \cdot t_0$, such that t is the lifecycle of frequent itemset, p is an integer, and t_0 is a unit of time.

Whenever a certain attenuating point of itemset arrives, the frequency of itemset is reduced by the decaying factor ϕ (ϕ <1). The new regressive strategy guarantees that the older a transaction is, the less its influence is put to the recent frequent itemsets it contains. The new strategy also secures that the attenuating rate of influence of transactions is slowing down with the lapse of time, which is generally consistent with the attenuating rate of radioelement reasoned from the formula above. Consequently, the regressive method is called radioactively attenuating strategy.

3.2 The Analysis of Accuracy

Now, we analyze the accuracy of radioactively attenuating strategy by comparing it with other two regressive strategies adopted by FTP-DS algorithm and escDet algorithm respectively.

The FTP-DS algorithm only focuses on the transactions in the sliding window, and overlooks the influence of transactions outside the sliding window to the recent frequent itemsets. If the size of sliding window is too small, the result will fail to reflect the recent change of a data stream. In other words, the result will be inaccurate. If the size of sliding window is big enough, FTP-DS fails to differentiate the different influence of transactions in the big sliding window. However, the MRFISF algorithm with radioactively attenuating strategy takes all transactions in data streams into account, and gradually reduces the influence of old transactions to the recent itemsets. Obviously, the ability of monitoring the continuous variation of a data stream of radioactively attenuating strategy is better than that of FTP-DS algorithm.

The estDec algorithm mines the recent frequent itemsets in the data stream in which each transaction has a weight and decreases with age. However, there is a situation that some useful transactions might be faded away so fast that lessen the accuracy of result, because the influence of an old transaction is decayed whenever a new transaction arrives. Nevertheless, such an improper situation will not appear in the radioactively attenuating strategy, because the regressive rate of transaction is slowing down as the lapse of time, which guarantees that whenever a transaction is faded away, it already is useless or invalid for the recent frequent itemsets.

4 Experimental Evaluation

We evaluate our algorithm MRFISF by using a synthetic database generated by the IBM Quest Synthetic Data Generator. We use MFISF and MRFISF to denote the novel algorithms for mining frequent itemsets in data streams and for mining recent frequent itemsets by adopting the radioactively attenuating strategy respectively. The minimal support threshold θ is 0.1% and the maximal estimated support error threshold ε is a tenth of θ.

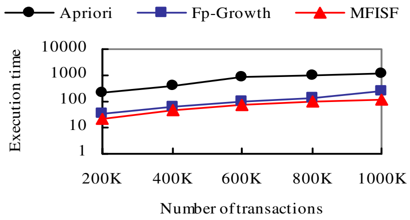


Fig. 3. The performance of algorithms (a)

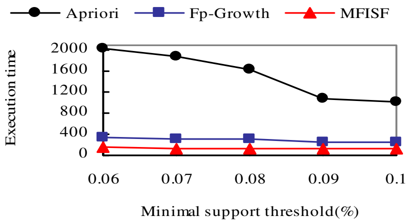


Fig. 4. The performance of algorithms (b)

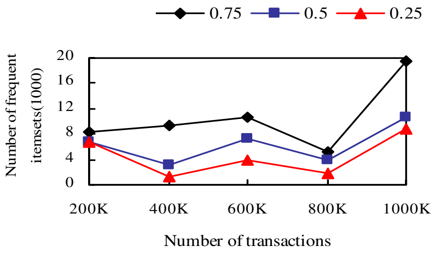


Fig. 5. The number of frequent item sets of different sizes of datasets

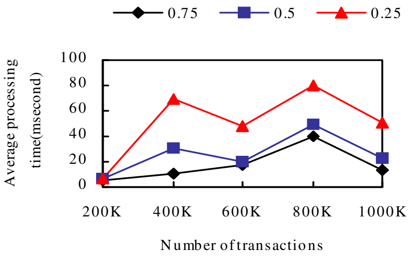


Fig. 6. Average provessing time

First, we compare the performance of Apriori and FP-Growth algorithms which can be downloaded in the web: *http://www.cs.umb.edu/~laur/ARtool/* with that of MFISF algorithm in the environment of T5.I4.D1000K. Fig 3 and Fig 4 show execution times of three different algorithms with the increasing dataset size and with the increasing minimal support thresholds. Since the test dataset is very dense, i.e. the frequency of each item is very high, the execution time of MFISF algorithm do not vary a great extent when θ varies. However, MFISF still outperforms both Apriori and Fp-Growth algorithms greatly. Second, Fig 5 and Fig6 provide the information concerning the number of frequent itemsets mined by the MRFISF algorithm with different decaying factors: 0.75, 0.5, 0.25 and the average processing time with the increasing size of dataset. The evaluation manifests the outstanding ability of MRFISF algorithm to mine recent frequent itemsets.

5 Conclusion

We proposed a new algorithm MRFISF to find out all recent frequent or frequent itemsets over the whole data streams. MRFISF not only takes full advantage of the special property between suffix-trees to avoid traversing every suffix-tree and generating candidate frequent itemsets, but also utilizes a newly-proposed itemset growth method to optimize the algorithm. Moreover, we also bring forward a novel regressive strategy for differentiating the information of recently generated transactions from that of obsolete transactions. Experiment results show that MRFISF algorithm has an excellent ability to mine recent frequent itemsets in data streams.

References

1. R. Agrawal, T. Imielinski, and A. Swami.: Mining Association Rules between Sets of Items in Large Databases. In ACM SIGMOD Conf. Management of Data (1993) 207-216
2. G. S. Manku and R. Motwani.: Approximate Frequency Counts Over Data Streams. In Proceeding of the International Conference on Very Large Data Bases, Hong Kong, China (2002) 346-357
3. C.Giannella, J. Han, and J. Pei, X. Yan and P. S. Yu.: Mining Frequent Patterns in Data Streams at Multiple Time Granularities. Next Generation Data Mining, Chapter 3 (2002) 191-211.
4. Ruoming Jin, Gafan Agrawal. An Algorithm for In-Core Frequent Itemset Mining on Streaming Data, [online] Available: http://www.cse.ohio-state.edu/~agrawal/ (2004)
5. W. G. Teng, M. S. Chen, and P. S. Yu.: A Regression-Based Temporal Pattern Mining Scheme for Data Streams. In Proceeding of the 29th VLDB Conference (2003) 93-104
6. J. Chang and W. Lee.: Finding Recent Frequent Itemsets Adaptively over Online Data Streams. In Proceeding of the ACM International Conference on Knowledge Discovery and Data Mining, Washington, DC (2003) 487-492
7. J. Chang and W. Lee.: A Sliding Window Method for Finding Recently Frequent Itemsets over Online Data Streams. Journal of Information Science and Engineering, (2004) 753-762

User Subjectivity in Change Modeling of Streaming Itemsets

Vasudha Bhatnagar and Sarabjeet Kaur Kochhar

Department of Computer Science,
University of Delhi, Delhi-110 007, India
{vbhatnagar, skochhar}@cs.du.ac.in

Abstract. Online mining of changes from data streams is an important problem in view of growing number of applications such as network flow analysis, e-business, stock market analysis etc. Monitoring of these changes is a challenging task because of the high speed, high volume, only-one-look characteristics of the data streams. User subjectivity in monitoring and modeling of the changes adds to the complexity of the problem.

This paper addresses the problem of i) capturing user subjectivity and ii) change modeling, in applications that monitor frequency behavior of item-sets. We propose a three stage strategy for focusing on item-sets, which are of current interest to the user and introduce metrics that model changes in their frequency (support) behavior.

Keywords: Data streams, Data mining, Monitoring, Change modeling, User subjectivity.

1 Introduction

Need to characterize the data generation process has been the chief motivation for stream monitoring systems. The underlying data generation process in a domain is a function of various parameters, some of which may be unknown. The objective of data mining technology is to discover some of these unknown parameters and interrelations between them. Monitoring the data is the natural course of action in such a scenario and modeling of changes in the discovered trends is an important step in the task of characterization. In applications such as network flow analysis, e-business, stock market analysis etc, monitoring of streaming data needs to be supplemented by focused observation of non-trivial changes in the discovered trends. It is these changes, which characterize the data generation process. This is true of any evolving database and particularly for a data stream.

Monitoring data streams is a challenging task because of limited memory usage and real time requirements[1, 5, 18, 25, 29]. Detection and monitoring of changes in the trends discovered from data streams is another challenge in stream analysis [1, 3, 19, 23, 26]. Applications that require monitoring the discovered trends are being increasingly used in the areas of web mining, network man-

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 812–823, 2005.

agement, market basket data analysis etc.. Monitoring applications are usually run as continuous queries on data streams managed by specialized data stream management systems [1, 23, 27]. Continuous queries are registered with DSMS and they typically maintain some synopses, which are updated after processing each transaction. After an initial gestation period, they yield continuous stream of results at some predefined periodicity. Thus a continuous query monitors the data stream for a fixed set of parameters (K-median and variance [6], moving averages, correlations [31] etc.) and leaves it to user to interpret the variations.

The changing trends in mined knowledge from data streams can be derived from changing data in the streams. The concept has been referred to as [illegible] and deserves attention. For example, in a process control system, observed increase in the average temperature of a boiler may indicate malfunctioning in the process, while rate of change may indicate the severity of the situation. Note that "severity" is a subjective notion with respect to the end-user and time, and therefore must be integrated with the task of knowledge differentiation. This makes monitoring applications more demanding in terms of user involvement.

Our Contribution. In this paper we address the problem of incorporating user subjectivity in modeling changes in the frequencies of streaming item-sets. We propose metrics that characterize the data generating process. The proposal forms the basis of a user centric trend monitoring system that integrates user subjectivity with change modeling and is capable of supporting following functions:

i) Allow user to dynamically focus monitoring on "interesting" item-sets (Section 2).
ii) Detect support changes of different orders in time varying trends, and automatically shifts the monitoring focus(Section 2 and Section 3).
iii) Computes lower order change metrics that facilitate derivation of higher order changes in data generation process (Section 3).

1.1 Related Work

User centric nature of KDD applications has been emphasized in [9] and has been the motivation for CRISP-DM model[16]. A user-centric model for KDD process which inherently supports design of monitoring applications has been proposed in [8, 30].

With the data stream management systems in place [1, 23, 27], the researchers have been drawn towards the problems of online mining of changes in time varying data streams[15, 19] and monitoring the data streams[4, 11, 17, 31]. The importance of monitoring applications has been highlighted in[11]. The real time needs of stream monitoring applications, which are not met by set oriented relational DBMS have also been emphasized in this work. The problem of monitoring large number of time series data streams in an online fashion has been addressed in[31]. The proposal monitors parameters such as moving average, standard deviation and correlations in multiple streams and makes observations over a three level time interval hierarchy.

The work presented in [20] is a significant beginning for quantifying the difference between two data sets in terms of the models they induce. An intuitive, user friendly process of analyzing significant changes and trends in data streams has been presented in [3]. The problem of change point detection in time series data has been investigated in [22], wherein the proposed iterative algorithm fits a model to a time segment and uses a likelihood criterion to determine if the segment contains new change points.

Some important works related to mining of frequent item-sets in streaming data can be found in [10, 14, 21, 28]. The estWin algorithm adaptively monitors the recent changes of frequent item-sets over a specified window in data streams[12]. EstDec algorithm [13] identifies the change of frequent item-sets over online data streams, diminishing the effect of old transactions by a decay function. A dynamic storage structure [illegible] is proposed in [10] to maintain the counts of frequent patterns. The structure is mined over multiple time granularities using time tilted windows and captures frequency histories of all the patterns in the pattern tree.

We observe that most of the above works have addressed the problem of detecting specific changes in the stream for all item(set)s. [illegible]

Organization of the Paper: The paper is organized as follows: We begin by laying down a three stage strategy for capturing user subjectivity in Section 2. Section 3 proposes metrics for modeling change in frequency behavior. Section 4 elaborates the experimental study and Section 5 concludes the paper.

2 User Focus

Focusing, which is a common technique for catering to user subjectivity, also offers online data reduction and faster online computation and forms the theme of this section. We propose a three stage filtering strategy to capture the user focus(interest). The first stage allows the user to define the monitoring space. Second stage accepts a two-level support threshold from the user and categorizes the item-sets as [illegible] or [illegible] item-sets. During the third stage, neighborhoods of the support thresholds are specified for fine grained monitoring.

2.1 Stage I: Monitoring Space

The first stage directly captures the user's current interest by allowing him to dynamically select items of perceived importance or de-select items that do not currently seem interesting for monitoring. Initially the user may choose items based on his requirement, experience or domain knowledge. Subsequently, the feedback about the behavior of item-sets provided by the system may influence user's decision.

Consider a transaction database D. Let there be n items in the database represented by a set of literals $I = \{i_1, \ldots, i_n\}$. Each transaction t consists of a set of items such that $t \in \{Pow(I) - \phi\}$. An item that is selected for monitoring is termed as [illegible]

Definition 1. [illegible] B [illegible] $M^S = \{Pow(B) - \phi\}$ [illegible]

Since set B can be dynamically updated either automatically (Section 2.3) or manually by the user, the set M^S is also time varying. In his work we omit the temporal variations in M^S and instead restrict to the study of frequency behavior of elements of M^S.

Modifications in set B lead to alteration in the monitoring space and may some times lead to high memory requirement. The following theorem provides the basis of the mechanism for controlling the size of B.

Theorem 1. [illegible] M [illegible] t [illegible] k [illegible] B [illegible] $log_2(M)$

[illegible] Let k be the cardinality of B at time t. We know that $|M^S| = O(2^k)$, i.e. $O(2^k)$ item-sets may have to be monitored in the worst case, with memory requirement of $O(2^k)$. Thus $O(2^k) \leq M$ or $k \leq O(log_2(M))$.

2.2 Stage II: Categorize Item-Sets

The second stage establishes the boundaries for categorizing the item-sets filtered from stage-I. The user specifies two support levels (S_l) and (S_h) which divide the set M^S into three partitions[1] (Figure 1). The left tail of the curve represents [illegible] item-sets with support $< S_l$. The middle partition, represents [illegible] item-sets with support between S_l and S_h and the right tail of the curve represents the set of [illegible] item-sets with support $\geq S_h$. Note that this subsumes the traditional model for frequent item-set mining, where $S_l = S_h$.

2.3 Stage III: Automatic Focusing

Transitions across the partitions may provide useful clues regarding changes in the frequency trends of the item-sets and form the basis of change modeling. Some of the transitions are naturally expected as the data generation process progresses e.g. gradual crossing of the item-sets from the left tail to the right tail of the curve (Figure 1). However, unexpected and frequent transitions may be indicative of changing nature of transactional data.

Singletons in the neighborhood of the support boundaries S_l and S_h are candidates for close monitoring because they are potential jumpers to adjacent

[1] The support counts of item-sets can be approximated by a Zipf (log-log) Curve[2].

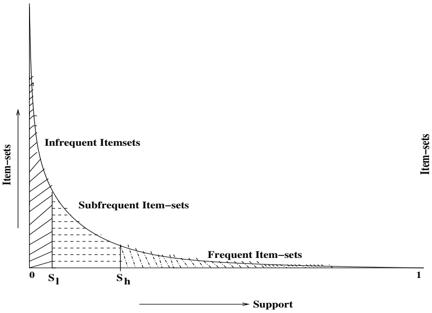


Fig. 1. Partitioning M^S using 2-level support threshold

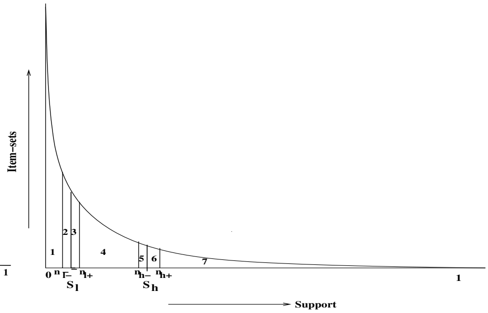


Fig. 2. Stage III of I-filter: Defining neighborhoods

partitions. These items are considered sensitive to changes in the data generation process. To focus on these items, stage III allows the user to specify neighborhoods $N_l = \{N_l^-, N_l^+\}$ and $N_h = \{N_h^-, N_h^+\}$, of the two support thresholds S_l and S_h respectively (Figure 2). Considering the neighborhoods to be the sensitive zone, singletons that move into a neighborhood $(\{N_l^-, N_h^-\})$ are automatically considered for close monitoring (included in B), or those which cross out of a neighborhood $(\{N_l^+, N_h^+\})$ towards the right tail are removed from focus (excluded from B).

User defines the neighborhoods by specifying $\{n_l^-, n_l^+, n_h^-, n_h^+\} \in [0,1]$, which divide M^S into seven partitions characterized by support levels $\{S_l - n_l^-, S_l, S_l + n_l^+, S_h - n_h^-, S_h, S_h + n_h^+\}$.

Stage III empowers the user to perform fine grained monitoring by dynamically defining the neighborhoods. Setting wide neighborhoods allows monitoring of large number of item-sets, though in practice memory and/or processor constraints may influence the width of neighborhood.

3 Change Modeling

Change modeling aims at identifying the changes that can arise in the context of the discovered knowledge[2] and the time interval over which it has been observed. Changes discovered over smaller time intervals are used to derive changes over longer time intervals, giving rise to notion of knowledge differentiation.

3.1 The Approach

Our approach to change modeling is based on comparing discovered trends over discrete intervals of time. This is different from the adaptive approaches[12, 6]

[2] For example, changing clustering schemes for an evolving database, time changing predictive models from a data stream, or changing frequency behaviors of sets of items.

where a window slides over the streaming data, and metrics are computed for recent data, discounting the effect of older data.

If t is the periodicity of mining, W^1 observations on mined results over $t * W^1$ units of time, lead to modeling changes of order one. Subsequently, observing W^{i+1} (essentially the window size) changes of order (i), and consolidating them leads to changes of order $(i+1)$. The size of the observation window for order (i) may encompass one or more windows of order $(i-1)$, as per the user desire.

Intuitively the technique of consolidation must be incremental in nature. We use additivity as the basis of consolidation, while admitting there can be more sophisticated approaches. Thus, [illegible] changes are available after periodic mining by subjecting the mined results to fast analytic processing. [illegible] changes are derived from [illegible] changes and are statements of changes over relatively longer duration of time. They give clues about the relative stability and consistency of the underlying data generation process.

The [illegible] changes portray the frequency change behavior of individual item-sets over finer granularities of time and are preserved as synopses, while the [illegible] changes, derived from the history of [illegible] changes, convey changes over relatively longer time durations. It is important to understand the underlying semantic differences between the changes of different orders, in the context of the type of discovered knowledge and the application domain. It is this area, which is complex and highly prone to user subjectivity. We take an approach for modeling the changes in the frequency behavior of item-sets that integrates both objective and subjective aspects.

3.2 Lower Order Changes

In this subsection we define change metrics of order one and two that characterize individual item-sets.

Persistence Factor. Persistence Factor of an item-set gives an idea of the uniformity of its frequency distribution in the stream with respect to time. It quantifies the [illegible] of an item-set in the partitions of the monitoring space (Figure 2) over a user defined observation window. PF of an item-set captures [illegible] change and is directly derived from the mined support of an item set.

Definition 2. [illegible] x [illegible] s_t^x [illegible] t[illegible], [illegible] $[0,1] \longrightarrow \{1,2,3,4,5,6,7\}$, [illegible]

$$M_x^t = \left\{ \begin{array}{l} 1 \quad\; s_x^t < n_l^- \\ 2 \;\; n_l^- \geq s_x^t < S_l \\ 3 \;\; S_l \geq s_x^t < n_l^+ \\ 4 \;\; n_l^+ \leq s_x^t < n_h^- \\ 5 \;\; n_h^- \geq s_x^t < S_h \\ 6 \;\; S_h \geq s_x^t < n_h^+ \\ 7 \quad\; s_x^t > n_h^+ \end{array} \right\} \tag{1}$$

In the above definition we use mapping to all the partitions defined by the user. However, the user is free to customize the mapping to a desired subset of these partitions.

As the data generating process progresses, item-sets transit from one partition to another and change their memberships.

Definition 3. [illegible] x [illegible] t [illegible] $t-1$

$$T_x^t = \left\{ \begin{matrix} 0 & \text{[illegible]} \ M_x^t = M_x^{t-1} \\ 1 & \text{[illegible]} \end{matrix} \right\} \tag{2}$$

Although individually each transition is both meaningful and interesting, sometimes non significant changes in the support count may cause alerts. It may not be prudent to alert the user for each of these (possibly) large number of temporary changes. We observe the membership of an item-set over a user defined observation window called [illegible], before concluding and communicating its behavior to the user. The size of the [illegible] is a function of the mining periodicity and is defined by the user as the number of observations W^1. Consolidation of membership over the [illegible] gives a measure of persistence of an item set in a partition. We call this measure [illegible] (PF) and define it as follows.

Definition 4. [illegible] PF_x^p [illegible] x [illegible] p [illegible] x [illegible] p [illegible]

Figure 3 shows the PF of an item x for P-window of size 5.

Mining Time	1	2	3	4	5
Partition No.	3	2	3	2	3

Fig. 3. Computation of PF: $PF_x^3 = 3/5$ and $PF_x^2 = 2/5$. For all other partitions PF of item x is 0/5

We use the concept of [illegible] to derive change metrics of order two, which characterize an item-set in linguistic term. The motivation behind this is the realization of the fact that interpretation of changes in PF trends is subjective with respect to the user and domain.

Item-Set State. The [illegible] of an item-set is an expression of the stability of its support over a user defined observation window called the [illegible]. It is derived by consolidation of PF of an item-set in a partition. The [illegible] encompasses one or more [illegible] as per the user desired time granularity. We use [illegible] of the PF of an item-set for a partition as the consolidation measure and then categorize the [illegible] using linguistic terms. We employ user given thresholds for categorization to facilitate the user level abstraction.

Definition 5. [illegible] S_x^p [illegible] x [illegible] p [illegible] $[0,1] \rightarrow \{Miss, Rare, Oscillating, Consistent, Concrete\}$ [illegible]

$$S_x^p = \left\{ \begin{array}{ll} iss & vg(PF_x^p) \leq \theta_{miss} \\ are & \theta_{miss} < vg(PF_x^p) \leq \theta_{rare} \\ ransient & \theta_{rare} < vg(PF_x^p) \leq \theta_{trans} \\ onsistent & \theta_{trans} < vg(PF_x^p) \leq \theta_{cons} \\ oncrete & vg(PF_x^p) > \theta_{cons} \end{array} \right\} \quad (3)$$

[illegible] $\theta_{miss}, \theta_{rare}, \theta_{trans}, \theta_{cons}$ [illegible]

The user can query the state of an item-set in a given partition in a given time interval.

Item-Set Spread. We introduce the notion of [illegible] of an item-set as an indicator of its variation in support count over the [illegible], and hence an intuitive index of its volatility. Higher value of [illegible] indicates higher range of variation in the support of an item-set over the window. Spread of an item-set at a time instant is computed as follows:

Definition 6. [illegible]

$$S_x = max_{i=1}^{W^2}(\sum_{p=1}^{7} \lfloor PF_x^p \rfloor) \quad (4)$$

[illegible] W^2 [illegible]

The $\sum$ term computes the spread in a [illegible]. Spread of an item-set x is the maximum of the spreads over all [illegible]s in the [illegible]. Thus for an item x, $S_x = 2$ (Figure 3), indicating that it was present in only two partitions during the current [illegible] of size same as [illegible].

This definition of spread recognizes even spurious presence of an item-set in a partition. The extent of presence of an item-set in a partition that is important for the user, is integrated in the following definition:

Definition 7. [illegible] $\gamma \in \{0, 1\}$

$$S_x = max_{i=1}^{W^2}(\sum_{p=1}^{7} (PF_x^p \geq \gamma) \; 1 : 0)) \quad (5)$$

[illegible] W^2 [illegible]

3.3 Higher Order Changes

The Higher order changes aim at providing an insight into the long term behavior of the data generation process. They make a statement about its consistency and relative stability. They are derived from the history of lower order changes. Due to lack of space, we just mention two possible change metrics, skipping the detailed discussion.

Transition Rate. $\mathcal{T}_x$ of an item-set x over the observation window $[t1, t2]$ quantifies the stability of data generation process with respect to the item-set x. Lower value of $\mathcal{T}_x$ indicates higher stability of the data generation process.

Consistency Quotient. of the data generation process quantifies its stability. A high Consistency Quotient indicates high stability of the data generation process while a weak one points that the data generation process has been showing changes quite often. It is highly user subjective and can be computed in more than one ways: i) average or weighted transition rate of all the monitored item-sets ii) Proportion of total number of item-sets with "consistent" state with respect to total number of monitored item-sets.

4 Experimental Study

We developed muti-threaded C++ program to implement the proposal. The program was compiled using gcc compiler and executed under Red Hat Linux 7.3 operating system. The hardware environment consisted of 2.3 GHz AMD Athlon XP processor and 256 MB DDR RAM. The program was run in a stand alone environment, with no other user process.

The program consists of two threads. The first thread simulates both I-filter and mining unit (since these program units must run in mutually exclusive mode). The second thread simulates monitoring unit and computes change statistics.

The experimental study was performed using a synthetically generated data-set. A customized data generating program was required to control the frequency behavior of item-sets selected for study. The program allows the user to control the size, cardinality and average transaction length of the data-set. The generated data-set contained 1000 items and 10 million transactions which were continually streamed in for performing the experiments. We chose the set of [illegible], B $=\{I_1,I_2,I_3\}$ giving us the monitoring space of size 2^3. We, however, chose to focus on a subset of M^S, S=$\{I_1,I_2,I_1I_2,I_3\}$, for study. The support for all the item-sets except for the members of set S was random.

The experiments were performed w.r.t. only one partition as it was easy to introduce and demonstrate the changes in a single partition. We chose partition number 6 (Figure 2) for the purpose of demonstration. Following parameters were supplied by the user through a parameter file: $S_l = 0.3$, $S_h = 0.7$, $S_h + n_h^+ = 0.9$, $S_h - n_h^- = 0.6$, $S_l + n_l^+ = 0.4$, $S_l - n_l^- = 0.1$, S-window size = 3 units. [illegible] was calculated with: $\theta_{miss} = 0.2$, $\theta_{rare} = 0.3$, $\theta_{trans} = 0.5$, $\theta_{cons} = 0.7$. The mining periodicity was chosen so as to allow approximately 25000 transactions for PF calculations.

The first experiment was designed to demonstrate detection of lower order changes. For this purpose, we induced pre-configured variations in the frequency of item-sets belonging to S. Figure 4 shows these induced variations and Figures 5 and 6(a) show the computed [illegible] and [illegible] respectively. The results demonstrate that the system is able to faithfully detect the induced changes as follows:

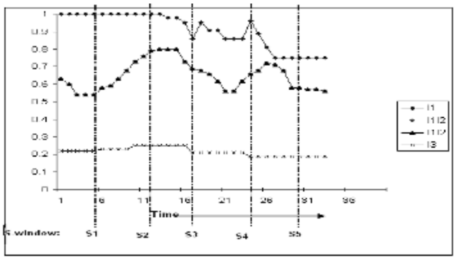
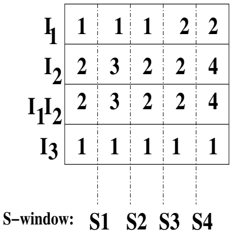
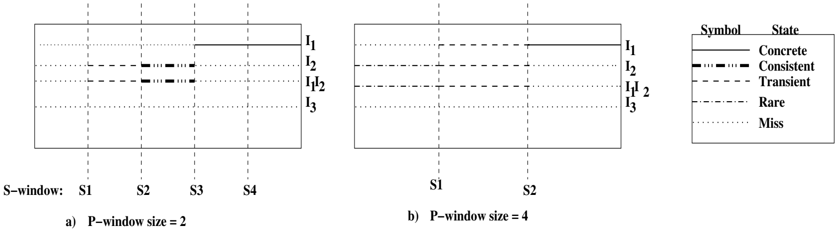


Fig. 4. Support of items I_1, I_2, I_1I_2, I_3



I1	1	1	1	2	2
I2	2	3	2	2	4
I1I2	2	3	2	2	4
I3	1	1	1	1	1
S-window:	S1	S2	S3	S4	

Fig. 5. Spread of items I_1, I_2, I_1I_2, I_3



a) P–window size = 2

b) P–window size = 4

Fig. 6. Effect of P-Window size on state of item-sets

i) The support for I_1 was kept steadily high for the duration of three S-windows and later fluctuated. Figure 5 shows the confinement of I_1 to one partition for first three S-windows and in two partitions thereafter. Figure 6(a) shows I_1 to be totally , -.. for the first three S-windows and . . . , thereafter in the studied partition.

ii)The support of I_2 was kept constantly fluctuating over different S-windows. Figure 5 captures this behaviour of item. It shows I_2 to be present in two partitions for first S-window, in three partitions during second, in two partitions for third and fourth S-windows and in three partitions thereafter. Figure 6(a) shows the behaviour of I_2 with respect to partition 6. I_2 is . -.. from the partition for the first S-window, ,. . . .- . , in the second S-window, . . .-., . , in the 3rd and , -.. thereafter.

iii) The support of I_3 is very low throughout. Figure 6(a) shows that it was totally . -.. from the studied partition and Figure 5 shows that its presence was restricted to one partition throughout.

The second experiment was designed to study the effect of P-window size on the granularity of change consolidation for computing .., , of an item-set. The experiment was performed with two different P-window sizes. As expected, the results show that while the smaller P-window size captures finer changes, the larger P-window size shows consolidated changes. In Figure 6(a), P-Window of size 2 shows that item I_2 was initially , -.. , became ,. . . .- . , and finally . . .-., . , before going totally . -.. again from the studied partition. Figure 6(b) consolidates this behavior of I_2 by depicting that for a P-Window of size 4, I_2 was initially [illegible], became [illegible] and went totally [illegible] thereafter.

5 Conclusion

In this paper we highlighted the importance of user involvement in monitoring and modeling changes in data streams. We proposed metrics of varying orders for capturing changes in the discovered support trends. The lower order changes characterize individual item-sets while the higher order changes characterize the underlying data generation process. An important feature of this work is capturing user subjectivity for modeling changes of all orders.

References

1. D. Abadi, D. Carney, et al. Aurora: A Data Stream Management System. In SIGMOD '03: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, pages 666–666. ACM Press, 2003.
2. L. A. Adamic. Zipf, Power-laws, and Pareto - A ranking tutorial . Information Dynamics Lab, HP Labs, Palo Alto, CA 94304.
3. C. C. Aggarwal. An Intuitive Framework for Understanding Changes in Evolving Data Streams. In Proceedings of the 18th International Conference on Data Engineering (ICDE'02). IEEE Computer Society, 2002.
4. Arvind Arasu, Gurmeet Singh Manku. Approximate Counts and Quantiles over Sliding Windows. In ACM Symposium on PODS, 2004.
5. B. Babcock, S. Babu, M. Datar, et al. Models and Issues in Data Stream Systems. Proceedings of 21st ACM Symposium on PODS, 2002.
6. B. Babcock, S. Babu, et al. Maintaining Variance and K-Medians over Data Stream Windows. Proceedings of 22nd ACM Symposium on PODS, 2003, San Diego, CA.
7. S. Babu and J. Widom. Continuous Queries over Data Streams. Technical Report, Stanford University Database Group, Mar 2001.
8. V. Bhatnagar. Intension Mining: A New Approach to Knowledge Discovery in Databases. PhD thesis, Jamia Millia Islamia, New Delhi, India., 2001.
9. Brachman and Anand. The Process of Knowledge Discovery in Databases. Chap.2 in Advances in Knowledge Dicovery in Databases, AAAI/MIT Press, 1996.
10. C. Giannella, J. Han, J. Pei, X. Yan, and P.S. Yu. Mining Frequent Patterns in Data Streams at Multiple Time Granularities. H. Kargupta, A. Joshi, K. Sivakumar, and Y. Yesha (eds.), Next Generation Data Mining, 2003.
11. D. Carney, U. Centintemel, et al. Monitoring Streams: A New Class of Data Management Applications. In Proceedings of the 28th VLDB Conference, China, 2002.
12. J. H. Chang and W. S. Lee. estWin:Adaptively Monitoring the Recent Change of Frequent Itemsets over Online Data Streams. In Proceedings of the 12th CIKM, pages 536 – 539, New Orleans, LA, USA, 2003.
13. J. H. Chang and W. S. Lee. Finding Recent Frequent Itemsets Adaptively over Online Data Streams. In ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), pages 487 –492, 2003.
14. M. Charikar, K. Chen, and M. Farach-Colton. Finding Frequent Items in Data Streams. Theor. Comput. Sci., 312(1):3–15, 2004.

15. G. Cormode and S. Muthukrishnan. What is new: Finding Significant Differences in Network Data Streams. In IEEE INFOCOM 2004.
16. CRISP-DM Homepage. CRoss Industry Standard Process for Data Mining. http://www.crisp-dm.org.
17. M. Datar, A. Gionis, P. Indyk, et al. Maintaining Stream Statistics over Sliding Windows. In Annual ACM-SIAM SODA, Jan 2002.
18. P. Domingos and G. Hulten. Catching Up with the Data: Research Issues in Mining Data Streams. In ACM SIGMOD Workshop on Research issues in Data Mining and Knowledge Discovery, 2001.
19. G. Dong, J. Han, L.V.S. Lakshmanan and others. Online Mining of Changes from Data Streams: Research Problems and Preliminary Results. In Proceedings of the ACM SIGMOD Workshop on Management and Processing of Data Streams., 2003.
20. V. Ganti, J. Gehrke, R. Ramakrishnan, et al. FOCUS : A Framework for Measuring Differences in Data Characterstics. In Proc. of 18th Symposium on PODS, 1999.
21. Graham Cormode and S. Muthukrishnan. What's Hot and What's Not: Tracking Most Frequent Items Dynamically. In Proceedings of the 22nd ACM SIGMOD-SIGACT-SIGART symposium on PODS, pages 296–306. ACM Press, 2003.
22. V. Guralnik and J. Srivastava. Event Detection from Time Series Data. In Proceedings of the fifth ACM SIGKDD 1999, pages 33 – 42.
23. The STREAM Group. STREAM: The Stanford stream data manager. IEEE Data Engineering Bulletin, 26(1), 2003.
24. J. Han, J. Pei, and Y. Yin. Mining Frequent Patterns without Candidate Generation. In Proceedings of Int'l. Conf. SIGMOD 2000, May 2000.
25. M. R. Henzinger, P. Raghvan, and S. Rajgopalan. Computing on Data Streams. SRC Technical Note 1998 -011, Digital Systems Research Center, Palo Alto, California, May 1998.
26. G. Hulten, L. Spencer, and P. Domingos. Mining Time-Changing Data Streams. In Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, pages 97–106. ACM Press, 2001.
27. J. Chen, D. Dewitt, F. Tian, and Y. Wang. Niagracq: A Scalable Continuous Query System for Internet Databases, pages 379–390, 2000.
28. G. S. Manku and R. Motwani. Approximate Frequency Counts over Data Streams. In Proceedings of the 28th Intl Conf on VLDB, Hong Kong, China, Aug 2002.
29. S. Muthukrishnan. Data streams: Algorithms and Applications. In Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 413–413. Society for Industrial and Applied Mathematics, 2003.
30. S.K.Gupta, V. Bhatnagar, et al. Architecture for Knowledge Discovery and Knowledge Management, Knowledge and Information System Journal, 7(3), pages 310–336, Springer-Verlag London Ltd, 2005.
31. Yunyue Zhu, Dennis Shasha StatStream: Statistical Monitoring of Thousands of Data Streams in Real Time. In International Conference on VLDB, China, 2002.

A Grid-Based Clustering Algorithm for High-Dimensional Data Streams

Yansheng Lu, Yufen Sun, Guiping Xu, and Gang Liu

College of Computer Science & Technology,
Huazhong University of Science & Technology,
Wuhan, 430074, China
yufens@163.com

Abstract. The three main requirements for clustering data streams on-line are one pass over the data, high processing speed, and consuming a small amount of memory. We propose an algorithm that can fulfill these requirements by introducing an incremental grid data structure to summarize the data streams online. In order to deal with high-dimensional problems, the algorithm adopts a simple heuristic method to select a subset of dimensions on which all the operations for clustering are performed. Our performance study with a real network intrusion detection stream data set demonstrates the efficiency and effectiveness of our proposed algorithm.

1 Introduction

The demands of recent applications promote the researches on data streams. By nature, a data stream is an appropriate model when a potentially infinite volume of data is arriving continuously and it is either unnecessary or impractical to store the data in some form of memory [1]. The applications that generate data streams include web applications, network monitoring, telecommunications data management, stock-market analysis, sensor networks, and so on.

Clustering is a widely used technique for data mining, indexing, and classification [11]. The aim of clustering is to partition a data set into subsets (clusters) such that members of the same cluster are similar and members of distinct clusters are dissimilar, where the similarity of two data members is usually defined by a distance function [5]. Traditional clustering algorithms typically process data that are stored in a secondary memory. For an online clustering algorithm for data streams, it cannot require information from slow secondary memory because it must catch up with the continuously arriving data streams [7]. So besides the two basic considerations of a clustering algorithm, i.e. the requirements of running time and memory usage, an online clustering algorithm for data streams should be incremental and single-pass [6]. Moreover, a lot of stream data are high dimensional in nature [4], so an algorithm for data streams should have the ability to deal with high-dimensional data. For an evolving data stream, a clustering algorithm should pay more attention to recent data than outdated history data. By now, only one algorithm called HPStream considers all of these requirements [4], and others just consider some of the requirements [1, 2, 3, 6].

X. Li, S. Wang, and Z.Y. Dong (Eds.): ADMA 2005, LNAI 3584, pp. 824 – 831, 2005.

In this paper, we propose an algorithm called Grid-based Clustering algorithm for High-dimensional Data Streams (GCHDS) that uses a two-level structure to perform online clustering on high-dimensional evolving data streams. Unlike other two-level algorithms for data streams in which the second level needs to be processed off-line [1, 2, 3], both levels of our algorithm can be processed on-line. In our algorithm, the arriving data are summarized into a grid data structure, then the grid data structure is processed to find connected cells as clusters in a selected subspace. Our algorithm can produce high accurate clusters within a very short running time.

In the following parts of the paper, we first discuss related work in section 2. Then an incremental grid data structure that can summarize a data stream on-line is introduced in section 3. The main algorithm GCHDS is given in section 4. The method for selecting dimensions to construct the subspace in which the clustering is performed is also described in this section. Our performance study on a real data set is reported in section 5. We conclude our algorithm in section 6.

2 Related Work

In literature, there have been several clustering algorithms proposed for data streams [1-4, 6]. Unable to perform complex clustering fast enough to catch up with the stream data, most of these algorithms divide the clustering process into an online component which periodically stores detailed summary and an offline component which performs complicated clustering on this summary [1, 2, 3]. The summary is often acquired by a rough clustering process. For algorithm STREAM [1], the k weighted centers for each chunk can be thought as a compressed representation of the chunk, and the final output of this algorithm is obtained by clustering these weighted centers of i chunks. Both the weighted centers and the final output are produced by algorithm LSEARCH, which produces high quality clusters at a cost of long processing time. Wang et al propose a two-tier structure method that produces characteristic points as the summary of a data chunk [2]. An improved CURE algorithm is then used to perform clustering on these characteristic points off-line. The algorithm CluStream maintains online micro-clusters that are defined as a temporal extension of the clustering feature (CF) vector to summarize the stream data [3]. An offline macro-clustering process then uses the summary statistics of the micro-clusters to produce the clustering result.

Algorithm CluStream [3] and algorithm HPStream [4] process evolving data streams by using time windows and a fading function. In our grid-based algorithm, a fading parameter is used to eliminate the influence of history data and outliers.

For high-dimensional clustering tasks, clustering algorithms that define the similarity between data numbers as some kind of distance between them will produce poor clustering results because all pairs of points tend to be almost equidistant from one another in a high-dimensional space [4]. Projected clustering algorithms that perform clustering in subspaces of the original data space are proposed especially to solve this problem [11, 12, 13]. CLIQUE is the first such algorithm that identifies clusters in different subspaces, which are comprised of different combinations of dimensions [12]. By finding dense units in all subspaces, CLIQUE identifies clusters in these subspaces as connected dense units. The clusters produced by this algorithm may overlap largely, which makes the algorithm cannot fulfill many applications' requirements that the data set should be partitioned into disjoint subsets. Moreover, the running time of CLIQUE is exponential in the highest dimensionality of any dense unit. So it cannot be used to process data stream on-line. PROCLUS is another projected clustering algorithm that tries to find clusters in different subspaces [13]. But PROCLUS cannot be used to process data stream on-line because it uses a complex iterative phase to find clusters and their corresponding dimensions. DOC is a Monte Carlo algorithm that repeatedly does random sampling and then evaluates the sample [11]. The best sample is considered as the approximation of an optimal projective cluster. These algorithms for subspace clustering on static data set are not suitable for data streams because they have high computational complexity and they cannot be easily extended to be incremental. To catch up with the speed of data streams, the process used to decide the subspaces in which clusters lie must be simple and effective. HPStream is the first algorithm that is proposed to perform projected clustering on high-dimensional data streams [4]. For each cluster, HPStream selects the dimensions over which the radii of the cluster are small. SURFING is a simple algorithm that tries to find interesting subspaces by checking whether the projections of data points in one subspace distribute uniformly [14]. If the distribution of the projections of data points in a subspace is not uniform, there may be some clusters exist in this subspace. However, SURFING needs to compute k -nearest neighbors of every data point in every subspace, which is not a trivial task. Our algorithm GCHDS also choose subspaces by checking whether the projections of data points distribute uniformly, but we only check the projections on every single dimension. GCHDS outputs clusters in one subspace at a time, not clusters in different subspaces. This is acceptable because the number of clusters in a time window is usually small and the clusters can be identified in a subspace. The Clusters produced by GCHDS at different time may lie in different subspaces.

CluStream and HPStream have the problem of favoring clusters with spherical shape because they use extended CF vectors that are proposed in BIRCH [8] as the summary for data streams [3, 4]. Grid-based algorithms haven't this shortcoming. WaveCluster [9, 10] is a grid-based algorithm that outperforms BIRCH, CLARANS, and DBSCAN in terms of both efficiency and clustering quality [5]. We find that most virtues of this algorithm come from the grid structure and the connected component analysis. Based on this idea, we introduce grid structure and connected component analysis to data stream domain. A connected component analysis is performed on a grid data structure that is built as a summary of a data stream.

3 Maintaining a Grid Data Structure on a Data Stream

We maintain a grid data structure on a data stream in main memory. To formalize our problem, we expand the definition in WaveCluster algorithm [10] to data stream domain. Let $A = A_1, A_2, \cdots, A_d$ be a set of bounded, totally ordered domains and $S = A_1 \times A_2 \times \cdots \times A_d$ be a d-dimensional data space. $A_1, A_2, \cdots, A_d$ are referred as dimensions of S. The input stream data are d-dimensional points $X = X_1, X_2, X_3, \cdots$, where $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$. The j-th attribute of x_i is drawn from domain A_j.

The initial grid structure GS is built on the first chunk of data points in the data stream. This chunk should fit into main memory. The data points' minimum and maximum on each dimension are found out. Let $l_i^{\max}$ be a value that is little larger than the maximum on dimension A_i, $l_i^{\min}$ be the minimum on dimension A_i, the range of dimension A_i is $[l_i^{\min}, l_i^{\max})$, $1 \le i \le d$. We partition each dimension into k intervals. The intersection of one interval from each dimension forms a hyper-rectangle, which we call a cell. A cell C_i has the form $(c_{i1}, c_{i2}, \cdots, c_{id})$, where $c_{ij} = [l_{ij}, h_{ij})$ represents a right-open interval in the partitioning of A_j. A point $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$ is contained in a cell C_m if $l_{mj} \le x_{ij} < h_{mj}$ for $1 \le j \le d$. The number of points contained in each cell is stored with the cell to summarize the data distribution information. We only store the cells $(c_{i1}, c_{i2}, \cdots, c_{id}, count_i)$ with $count_i > 0$ as initial grid structure GS in main memory.

The grid structure GS is updated incrementally as the stream data flow in. For a new arrived data point $X_i = (x_{i1}, x_{i2}, \cdots, x_{id})$, we try to find a cell C_m that satisfies $l_{mj} \le x_{ij} < h_{mj}$ for $1 \le j \le d$, then assign $count_m + 1$ to $count_m$. If such cell does not exist, the grid structure is expanded to contain the data point. In order to compute the $count$ in each enlarged cell, the original cells cannot be partitioned. Let n represent the multiple which a dimension is expanded to, k represents the number of intervals on each dimension, this condition is satisfied when $n \ge k$. If $n < k$, k should be a number that can be divided exactly by n or $n^{1/m}$ (m is a positive integer). Our algorithm doubles the range of the dimension every time. This requires that k is an even.

The number of cells stored will be far smaller than the number of data points that have arrived when the data show some characters of clustering. The data stream is compressed in this way.

4 The Grid-Based Clustering Algorithm for High-Dimensional Data Streams

All operations of our clustering algorithm are performed on the grid structure GS. To handle high-dimensional problems, a simple heuristic method is used to select dimensions that are useful for clustering. On the sub-grid structure that is spanned by the selected dimensions, the connected cells are labeled as clusters.

4.1 Selecting Dimensions

By analyzing real high-dimensional data sets, we found that on some dimensions, the values of data in different classes cannot be differentiated. Such dimensions can be categorized into two classes, i.e. spike dimensions and smooth dimensions. When almost all of the data points have similar values on a dimension, well call such dimension a spike dimension. The projections of data points on a spike dimension will squeeze together. When the data points distribute uniformly along a dimension, we call such dimension a smooth dimension. Obviously, these two kinds of dimensions are unhelpful for distinguishing clusters. Our method for selecting dimensions is to find out such dimensions and select the remainder dimensions to construct a subspace.

Let $N_{ij}, 1 \le i \le d, 1 \le j \le k$ represents the number of projections in interval j on dimension i, N_i represents the maximum of N_{ij} on dimension i. Then spike dimensions will have large N_i and smooth dimensions will have small N_i. Suppose we need to delete m spike dimensions and n smooth dimensions, where m and n are input parameters. First we find N_i for each dimension A_i, $1 \le i \le d$. Then we rank all dimensions in descending order of N_i. We get $d-m-n$ dimensions by deleting the first m dimensions and the last n dimensions.

When there is a need for clustering, the steps described above are performed first to select dimensions. Based on the distribution of the projections of data on each dimension, our method is insensitive to noise because noise normally distributes uniformly.

4.2 The Main Algorithm

Our main algorithm that realizes clustering on data streams is given below. The history data in the grid structure GS fade by a factor ε. The clustering steps are performed whenever there is a request for clustering. If the number of data points in a cluster is less than parameter p, the cluster is merged to another cluster.

Algorithm GCHDS

```
Input: the data stream $DS$, parameters $k, m, n, p, \varepsilon$
Output: $GS'$ with cluster labels, set $V$ that has $d-m-n$ elements

1. Build initial grid structure $GS$ on the data stream $DS$.
2. When a new data item arrives, update the grid structure $GS$ as described in
   section 3. Then fade all data in $GS$ by factor $\varepsilon$.
3. If there is a request for clustering, go to step 4; else, go to step 2.
4. Using the method described in section 4.1 to select $d-m-n$ dimensions. Re-
   cord these dimensions in a set $V$
5. Project each cell in grid structure $GS$ to the subspace that is spanned by the
   dimensions recorded in $V$. These projections constitute grid structure $GS'$.
6. Label connected cells in $GS'$ as clusters.
7. If the number of data points in a cluster is less than parameter $p$, merge the
   cluster into it's nearest cluster.
8. Go to step 2.
```

In step 6, we use the algorithm for finding connected components [15] to label connected cells as clusters. In order to find connected components in high-dimensional grid space, we expand 8-connectedness in a 2-dimesional space to high-dimensional spaces. Two cells C_m and C_n are considered being neighbors if for each $i, 1 \le i \le d$, at least one of the equations $c_{mi} = c_{ni}$, $l_{mi} = h_{ni}$, $h_{mi} = l_{ni}$ is satisfied.

Step 7 aims at reducing the number of clusters. Let dis_i represents the number of intervals between two cells on dimension i, we define the distance between two cells as the minimum of dis_i for $1 \le i \le d$. The distance between two clusters A and B is the minimum of the distances between cells in A and cells in B.

In our algorithm, the clustering results are represented by non-empty cells in the grid structure GS'. All data points that are projected into a cell will get the cluster label that is recorded in the cell.

5 Experiment Results

In this section, we empirically evaluate our GCHDS algorithm using the KDD-CUP'99 Network Intrusion Detection stream data set in which each record has a class label. We compare GCHDS with HPStream [4] to assess the accuracy and efficiency of GCHDS. We also analyze the effect of the input parameters on GCHDS by using different input value. All the experiments were performed on a PC with Intel Pentium IV processor and 256 MB memory, which run Windows 2000 professional operating system. We implemented our algorithm in Microsoft Visual C++.

In our experiments, we decay the data by ε every time 1000 new data points have arrived. When the $count$ of a cell is less than 0.8, the $count$ is set to 0.

We use the cluster purity that is defined as the average percentage of the dominant class label in each cluster to evaluate the clustering accuracy of GCHDS. Only those subset of points which arrive within a predefined window of time are used to compute the cluster purity. The cluster purity has been used by HPStream to assess the clustering accuracy in [4]. We compare our results with HPStream in Figure 1.

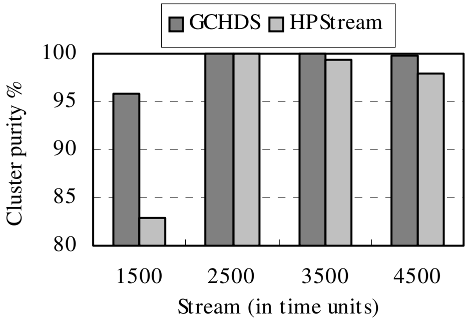


Fig. 1. Quality comparison ($k = 20, m = 14, n = 0, p = 48, \varepsilon = 0.9$)

In Figure 1, the stream speed is set at 100 points per time unit and one time window contains 1000 points. As in [4], we choose the time points when there are some kinds of attack connections happen. We can see that the cluster purity of GCHDS is always better than that of HPStream.

Figure 2 shows the efficiency test results of GCHDS and HPStream. The stream speed is set at 200 points per time unit. The algorithm efficiency is measured by the stream processing rate versus progression of the stream, which is defined as the inverse of the time required to process the last 1000 points [4]. For GCHDS, only the time needed for updating the grid is counted. After the time units 600, the processing rate is too large to be shown in the figure. Normally, the time needed for performing clustering on the grid is about several hundred milliseconds. So our algorithm is fast enough to be performed on-line as the data stream is flowing in.

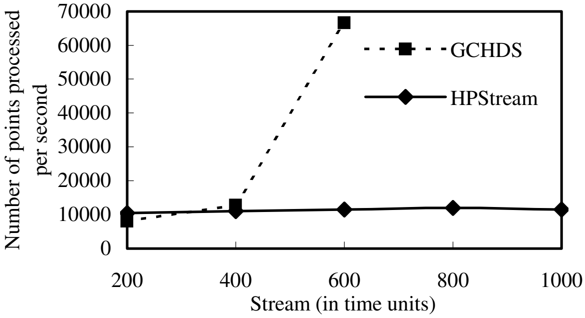


Fig. 2. Stream processing rate ($k = 20, m = 14, n = 0, p = 48, \varepsilon = 0.9$)

Because of the space restriction, here we summarize our experiment results on how the different input parameters affect the accuracy and efficiency of GCHDS. The cluster purity will increase when several spike dimensions or smooth dimensions are deleted, but it will decrease when too many dimensions are deleted. The cluster purity does not change much when the decay parameter ε changes. But the number of clusters will be closer to the number of classes when ε is given an appropriate value. Both the runtime needed for maintaining the grid and the runtime time needed for performing clustering on the grid drop sharply when the decay parameter changes from 1 to 0.95. With $p > 20$ we can effectively reduce the number of clusters to an acceptable level and gain high cluster purity at the same time. The parameter k does not affect the clustering accuracy much as long as it is not set too small.

6 Conclusion

We have presented a grid-based subspace clustering algorithm, GCHDS, for clustering of high-dimensional data streams. It fulfills the three requirements for clustering data streams on-line, i.e. one pass over the data, high processing speed, and consuming a small amount of memory. In this algorithm, a grid data structure is used to summarize the data stream on-line. The time for maintaining the grid structure is so small that it can be ignored when the grid has been enlarged widely enough to contain most of the data in the data stream. By analyzing the data distribution on each dimension, useful dimensions are selected to construct a subspace in which the clustering process is performed. Experiments show that GCHDS can gain high clustering accuracy when the parameters are appropriately set. It outperforms HPStream algorithm that is also a subspace clustering algorithm for high-dimensional data streams in terms of clustering accuracy.

Clustering on high-dimensional data streams is a difficult problem in data mining. The grid structure opens a new direction for solving this problem. The simple method used for selecting dimensions is significant for many applications because the data sets of many applications usually have many irrelevant dimensions for clustering.

References

1. O'Callaghan, L. et al., Streaming-Data Algorithms for High-Quality Clustering. Proc. of the 18th International Conference on Data Engineering, 2002
2. Wang, Z. et al., Clustering Data Streams on the Two-Tier Structure. The Sixth Asia Pacific Web Conference, 2004
3. Aggarwal, C. C. et al., A Framework for Clustering Evolving Data Streams. Proc. of the 29th VLDB Conference, 2003
4. Aggarwal, C. C. et al., A Framework for Projected Clustering of High Dimensional Data Streams. Proc. of the 30th VLDB Conference, 2004
5. Han, J. and Kamber, M. Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2001
6. Guha, S. et al., Clustering Data Streams: Theory and Practice. IEEE Transactions on Knowledge and Data Engineering, 2003, 15 (3): 515-528
7. Barbara, D. Requirements for Clustering Data Streams. ACM SIGKDD Explorations Newsletter, 2002, 3 (2): 23-27
8. Zhang, T., Ramakrishnan, R. and Livny, M. BIRCH: An Efficient Data Clustering Method for Very Large Databases. Proc. of the 1996 ACM SIGMOD International Conference on Management of Data, 1996
9. Sheikholeslami, G., Chatterjee, S. and Zhang, A. WaveCluster: A Multi-Resolution Clustering Approach for Very Large Spatial Databases. Proc. of the 24th VLDB Conference, 1998
10. Sheikholeslami, G., Chatterjee, S. and Zhang, A. WaveCluster: A Wavelet-Based Clustering Approach for Spatial Data in Very Large Databases. The VLDB Journal, 2000, 8 (3-4): 289-304
11. Procopiuc, C. M. et al., A Monte Carlo Algorithm for Fast Projective Clustering. Proc. ACM SIGMOD Int. Conf. On Management of Data (SIGMOD'02), 2002
12. Agrawal, R. et al., Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. Proc. ACM SIGMOD Int. Conf. On Management of Data (SIGMOD'98), 1998
13. Aggrawal, C. C. et al., Fast Algorithms for Projected Clustering. Proc. ACM SIGMOD Int. Conf. On Management of Data (SIGMOD'99), 1999
14. Baumgartner, C. et al., Subspace Selection for Clustering High-Dimensional Data. Proc. 4th IEEE Int. Conf. On Data Mining (ICDM'04), 2004
15. Horn, B. K. P. Robot Vision. The MIT Press, 1986

Author Index