I.J.Modern Education and Computer Science, **2010, 2, 25-31**
Published Online December 2010 in MECS (http://www.mecs-press.org/)

Data Mining based Software Development Communication Pattern Discovery

Gang Zhang
Faculty of Automation, GuangDong University of Technology, Guangzhou, China
Email: ipx@gdut.edu.cn

Caixian Ye
NiuTaiLai Communication Equipment Co.Ltd., Guangzhou, China
Email: yecaixian@tom.com

Chunru Wang and Xiaomin He
Faculty of Automation, GuangDong University of Technology, Guangzhou, China
Email: wangchunru99@126.com, teacher_smc@163.com

***Abstract*—Smaller time loss and smoother communication pattern is the urgent pursuit in the software development enterprise. However, communication is difficult to control and manage and demands on technical support, due to the uncertainty and complex structure of data appeared in communication. Data mining is a well established framework aiming at intelligently discovering knowledge and principles hidden in massive amounts of original data. Data mining technology together with shared repositories results in an intelligent way to analyze data of communication in software development environment. We propose a data mining based algorithm to tackle the problem, adopting a co-training styled algorithm to discover pattern in software development environment. Decision tree is trained as based learners and a majority voting procedure is then launched to determine labels of unlabeled data. Based learners are then trained again with newly labeled data and such iteration stops when a consistent state is reached. Our method is naturally semi-supervised which can improve generalization ability by making use of unlabeled data. Experimental results on data set gathered from productive environment indicate that the proposed algorithm is effective and outperforms traditional supervised algorithms.**

***Index Terms*—pattern discovery, software communication, decision tree, tri-training, semi-supervised learning**

I. Introduction

Software development is a team activity with specific goals. With effective inter and intra team communication, team collective strength can be greatly realized. Generally speaking, such communication includes sending, receiving, transferring of software development information, and combination between these behaviors [1,10]. Information in software development is characterized as reconcilable, transitive, imprecise, structured and uncertain, which makes the analysis a challenge, and powerful tools are required. IT technology of information processing and storing founds the basic of intelligent analysis of such information.

With application of modern information technology in network environment, there is dramatic improvement in information generation and communication, which has been studied in previous literatures [3]. However, the intelligent level of such procedure is relatively low, which lies in rare application of database and data mining technology. Current software project management pays little concern to software development communication. Investigation showed that a software engineer who works 40 hours per week spends only 16 to 18 hours in development averagely, while the remainder time is occupied by communication [10]. Communication has profound impact on software development, and an effective communication mechanism is beneficial to both project managers and developers. A good communication pattern requires intelligent analysis approach, which leads to a valuable decision support form a team leader. However, traditional communication methods are mainly based on natural language, such as training, technical meeting, internal discussion, which lacks support by advanced information technology, and is of low efficiency and high cost. Towards this end, we attempt to introduce the well established data mining algorithm to improve intelligence level of communication, and make use of huge volume of historic data to find useful pattern to advance decision support.

Data mining is a well established framework, aiming at intelligently discovering knowledge and principles hidden in massive amounts of data. Data mining technology processes data sets gathered from practical environment to find underlying relationship, principles and interesting patterns, or namely knowledge from the data set of huge amount of data. Data gathered from practical environment is often incomplete, noisy, ambiguous, which requires a data preprocessing step to make data well-structured and normalized, that is easily to be processed by formularized algorithm.

There are many successful applications of data mining technology. [6] Applied a association rule based data mining algorithm to find interesting pattern in market transaction data. [7] Proposed a data mining framework

in field of expert system design. They applied different data mining algorithm to score each subscriber in some concerned properties and made a comparison. [8] Investigated different clustering algorithms in analysis of a customer topology for a telecommunication company. Multiple factor analysis and weighted attributes were adopted in their work. They showed that both were effective for the target task. [9] proposed a priori-gen like algorithm to find frequent tree structure on a multi-core system. Their algorithm is essence in tree like data analysis in data mining.

When it comes to software development information analysis, the main point lies in how to collection and normalized data from practical environment, and how to discover the underlying patterns from massive information communication between developers and staff. Note that there is little done in software development information mining, while it is essential similarity between software information mining and other previous successful application [10,11].

The work in this paper is well motivated by the observation of challenge of software development information analysis and various successful applications of data mining technology, accompanied with rapid development of data mining algorithm.

Decision tree (DT) is a well studied data mining model to classify samples [12]. Information gain is calculated to evaluate the importance of each attribute associated with samples to the final classification task. Traditional DT only deals with labeled data in training procedure, namely the supervised learning. However, in software development information mining, labeled sample is rare and cost of labeling a sample is very high. We adopt semi-supervised learning to tackle this problem. Following [13]'s idea, we use a co-training styled algorithm to determine labels of unlabeled samples. In details, we first train three DTs with slightly different initial parameters by bootstrapping original training data set. Then a majority voting procedure is performed to determine the label of unlabeled data. And then update the training set and retrained DTs with newly training data set. With several iterations, we finally get all DTs into a consistent state, which yields the final predictor by ensemble of them.

The remainder of this paper is structured as following: Section 2 describes the basic concepts of intelligent communication pattern of software development. Section 3 presents the design of a decision tree based co-training styled algorithm to analyze communication data. Section 4 describes the experimental results and discussion. Finally there is a conclusion followed by several issues for future work presented in Section 5.

II. Information Communication Pattern Of Software Development

We introduce data mining and share repositories technology into software development information analysis to find useful pattern intelligently. To cut cost of communication in software development, the essential point is to reduce activity that people are involved directly, and to improve the level of information sharing.

A. Intelligent Information communication Pattern

With data mining and share repositories technology, programmer can obtain consultation and solution from central shared knowledge repositories, instead of consulting to other staff. Higher-level leaders and internal auditors can obtain data from development team members directly and do analysis with project leaders. With historical data, we can evaluate background and features that lead to communication problems, so as to avoid problems that may cause loss. Therefore reducing cost of human communication, maintaining an active communication pattern is good to increase the effectiveness of communication. We propose a framework of intelligent information communication pattern as shown in Fig. 1.

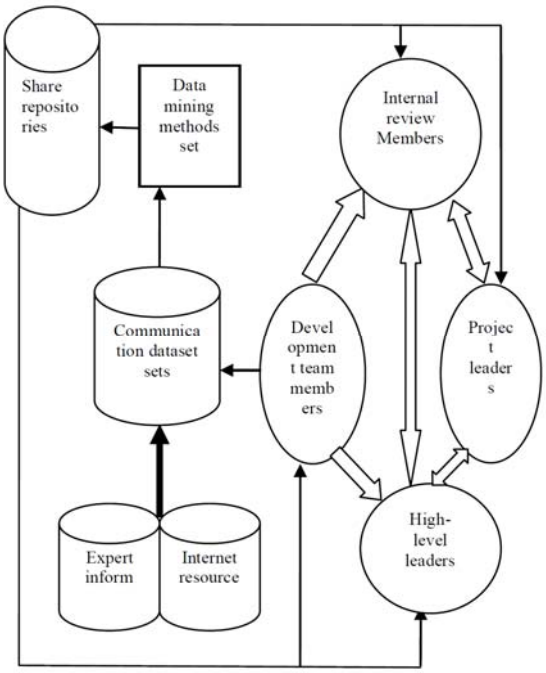

Figure 1. Intelligent Information Communication Framework

Some notes for Fig. 1:

a) Communicated data sets: Experience of development team members, problems encountered, expert information and Internet resources are collected to establish a data warehouse.

b) Share repositories: Data in communicated data sets is preprocessed and mined to get the knowledge and solutions to problems, and then it is stored into the classified shared repositories. Managers make decisions and staff can get a viable solution from share repositories when encountering a technical problem.

c) Data mining methods set: Data mining algorithms and pattern designed by technical staff.

d) Arrows and direction: Direction and line of information transmission.

B. Features of Intelligent Communication Pattern

a) Application of data mining technology. Processed by data mining technology, knowledge can be obtained

from software development communication information and stored into shared repositories by categories. The optimal solution to the communicated question will be searched from share repositories. Moreover, it provides an effective way to identify and evaluate reason and type of problems, which is helpful in controlling and reducing ratio of great accidents.

b) Categorical shared repositories are required. Shared repositories can improve information transmission speed and reduce frequency of direct communication between people. Everyone in software development can acquire knowledge from shared repositories.

c) Strong communication pattern is of natural time loss. Shared repositories make communication channels for the team members smooth, and there is no need to do any direct communication. Direct communication is needed only when shared repositories can't provide such information.

III. Mining Communication Pattern

Appling data mining technology to intelligent information communication pattern discovery would play an increasing important role in software development management procedure. Machine intelligence based knowledge shared repository is the best complement of human communication, education and training, which breaks the constraint of time and space, giving real time response and working in parallel. It greatly reduces the loss caused by direct interaction between staff.

A. The Data Mining Model

A data preprocessing procedure is launched to get a clean and normalized data set for model construction.

Data preprocessing [6, 13] stands for processing large volume of incomplete, noisy and inconsistent data gathering from practical environment, including data summarization, data transformation, data integration, data reduction and data cleaning. Data cleaning can be used to remove noise in the data set, fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies; data integration aims at merging multiple sources to form a unified database; data transformation is for normalization of data; data reduction can reduce the overall size of data set by gathering, removing redundant features or clustering data samples. The design of data mining models and algorithms form the framework of data mining of information communication. The data mining model of information communication is shown in Fig. 2.

B. Main Algorithm

Some critical issues frequently appeared in the software development communication, lead to interruption of development procedure, data error and so on, which may cause serious loss. Moreover, such issues would not be easily detected. It would be of very big trouble to software development without real-time detection of such issues.

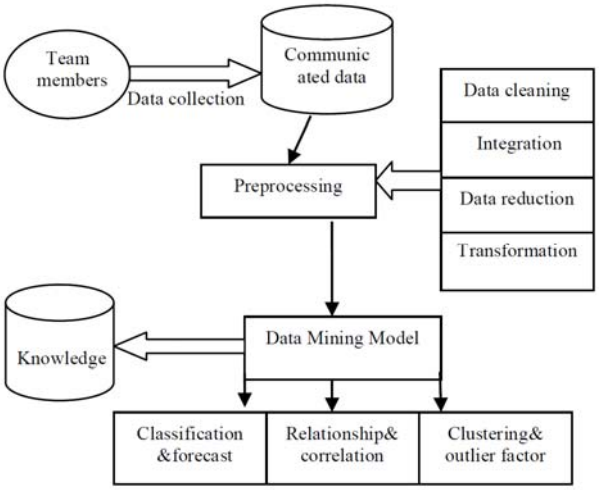

Figure 2. Data Mining Model of Information Communication

Based on the observation of the features of data in communication, we propose a co-training styled Decision Tree based algorithm Decision Tree Voting (DTV). The proposed algorithm is based on a previous successful co-training styled algorithm tri-training proposed in [13]. DTV makes use of human labeled data and unlabeled simultaneously and it is a real semi-supervised algorithm.

1) Decision Tree Attributes Selection

The essence of semi-supervised learning is to improve learner's generalization ability by correlation statistical distribution estimation on massive unlabeled sample. [7] Proposed a semi-supervised learning algorithm named tri-training, whose main idea was to generate three classifiers from the original labeled example set with slightly different initial parameters setting. Then a majority voting procedure is performed to determine labels of unlabeled data. Then the newly labeled data is added to the training set and learners are trained again. With several iterations, it would reach a consistent state finally.

The proposed algorithm requires selection of three initial classifiers. Previous work proved that in a majority voting like ensemble learning, higher accuracy and diversity between individual learner results in better generalization ability, meaning higher accuracy of the final ensemble learner [21]. The unlabeled data set is often large due to the high cost of human labeling. In order to get different based classifier, we consider two strategies. The first is to train based learners by bootstrap training data set. And the second is related to the type of based learners we choose. In our work, we use DT as based learner, thus we choose different splitting properties to get different separation Eigen functions. With both strategies mentioned above, we finally get DTs of high diversity.

A key issue in DT training is the choice of splitting properties. We adopt three metrics in evaluating importance of each property, which are information gain, Gini index and Goodman-Kruskal synthetic index.

a) Information Gain

Let D, be a training set of labeled samples, with class labels, defined as $D=\{(x,y)\,|\,x\in X, y\in Y\}$, in which X is the sample set and Y is the class label set of countable finite values.

Denote $x_i=(x_{i1},x_{i2},...,x_{id})\in X$ for d attributes of each sample in X and $Y=\{c_1,c_2,...,c_m\}$ for m distinct class labels. Let $|D|$ be the total number of samples D in and $|D(c_k)|$ be the number of samples of class label c_k, respectively.

For constructing a DT, each node represents a partition of samples by a certain property. We often choose the property of most information in partition the current samples set. We define information gain for a set of samples in partition as Eq. (1).

$$\text{Info(S)} = -\sum_{i=1}^{m} p_i * \log_2(p_i) \quad (1)$$

where S stands for the samples set to be partitioned, and $p_i=|D(c_i)|/|D|$ which means the probability of any sample belongs to class c_i.

Define $\text{Info}_A(\text{S})$ as information evaluation of partition set S by attribute A, where $S_A(A_j)$ stands for samples in S with value A_j in attribute A.

$$\text{Info}_A(\text{S}) = -\sum_{j=1}^{v}(|S_A(A_j)|/|S|) * \text{Info}(\text{S}_A(\text{A}_j)) \quad (2)$$

$\text{Info}_A(\text{S})$ is the expected information evaluation of introducing attribute A in partition S. Information gain of attribute is then obtained as Eq. (3).

$$\text{gain(A,S)=Info(S)-Info}_A\text{(S)} \quad (3)$$

We choose the attribute that leads to maximum gain(A,S) as the splitting attribute.

b) Gini Index

Gini Index is a metric for measurement of node impurity in a DT. It means in a sample set associated with a node in a DT, the degree of all samples belong to the same class. Formally, we define Gini target function as Eq. (4).

$$\text{Gini(A,S)} = 1-\sum_{i=1}^{m} p_i^2 \quad (4)$$

in which p_i is defined the same as in previous sub section. We choose the attribute that leads to minimum $Gini(A,S)$ as the splitting attribute.

c) Associated index based on Goodman-Kruskal

It is a numerical measurement of partition on finite set. It builds a partition metric by construction a DT, based on Goodman-Kruskal [9,13] coefficient GK. We give a formal definition as following.

Let $P_1=\{B_1,B_2,...,B_m\}$ and $P_2=\{C_1,C_2,...,C_n\}$ are two partitions of a sample set D. The Goodman-Kruskal coefficient GK is defined as Eq. (5).

$$GK(P_1,P_2) = 1-\sum_{i=1}^{m}\max_{1\le j\le n}|B_i\subseteq C_j| \quad (6)$$

We then define a metric for partitions of a set of samples, denoted as dGK. We have:

$$dGK(P_1,P_2) = GK(P_1,P_2)+GK(P_2,P_1) \quad (7)$$

Smaller dGK value means better partition ability of certain attribute. With dGK based attribute selection, we can get DT of smaller size.

2) Critical Issue Mining with DTV algorithm

Let L be labeled sample set and $|L|$ denotes its size. Let U be unlabeled sample set and $|U|$ denotes its size. Three DTs classifiers are training with training data sets generated by bootstrapping the labeled sample set L. A sample is randomly selected from unlabeled sample set U whose label is determined by a majority voting procedure of three DTs. This newly labeled sample is added to the training data set L and three DT classifiers are trained again. [7] Proved that this procedure can reduce noise to an outstanding degree.

Three decision trees voting classification algorithm based on tri-training applied to the critical issues mining is shown as Algorithm 1.

Algorithm 1 DTV

Input: Labeled Set L
Unlabeled Set U
Output: Labels of samples in U

1: for k = 1 to 3
2: $L_i = bootstrap(L)$
3: $DT(i) = train\_DT(L_i)$
4: end
5: $L \to S$
6: While U not empty
7: $L_i = bootstrap(S)$ $i=1,2,3$
8: $U \to U_i$
9: Label U_i with $DT(i)$
10: Select sample subset P_i by optimal strategy
11: $L_i \cup P_i \to L_i$
12: $DT(i) = train\_DT(L_i)$
13: $L_1 \cup L_2 \cup L_3 \to S$
14: Update labels in S by majority voting of $DT(i)$
15: End

IV. Experiment and Discussion

We perform the experiment to evaluate the proposed algorithm on a real software development communication data set, which was gathered from a software R&D company during 2006-2008, targeting at the critical issues for data mining research. The data set contains binary-labels samples only, with values Yes and No.

Each sample in the data set is of 48 attributes which describes several original and statistical properties in communication of software development. To simply the experiment, we launch data preprocess to clean and normalize the original data set. All attributes are converted to categorical ones, which are suitable for Decision Tree. Though some modified version of DT can deal with numerical attributes effectively, we refer to the simple version of DT so as to focus the classification ability of the model.

Since the result of the proposed algorithm would be used by software developers and administrators in the company, we randomly select 4280 records as the experimental data set to fit for the data scale and accuracy requirement of the proposed algorithm. The data set is randomly divided into two parts, in which the first part is training data set with 2020 records and testing set with 2260 records. ID3 is used as the DT training algorithm with three different metrics mentioned in Section 3 as attribute importance evaluation. We train DTs with different metric and evaluate their performance. Experiments repeat 10 times with randomly partition of training and testing data set. Since the proposed DTV algorithm is naturally semi-supervised, a transductive learning procedure is performed, which means testing data set is also used in the model training step.

TABLE I illustrates the accuracy and improvement through semi-supervised learning of unlabeled data.

TABLE I. ACCURACY OF DTV

	Total (%)	DT1		DT2		DT3	
		a %	b %	a %	b %	a %	b %
Set1	6.86	9.12	24.78	11.87	42.21	10.18	32.61
Set2	6.21	10.32	39.83	13.11	52.63	11.67	46.79
Set3	6.00	10.94	45.16	10.01	40.10	9.32	35.62

In TABLE I, column title *Total* stands for the ensemble accuracy of three DTs through majority voting. Sub columns with titles *a* and *b* stand for accuracy of each DT before and after learning from unlabeled data. We can see from the initial experimental setting, unlabeled data can greatly improve the classification accuracy.

We vary the ratio between labeled and unlabeled data set to test the ability of processing unlabeled data of the DTV algorithm. We set the ratio between labeled and unlabeled as 1:4, 1:3 and 1:2. TABLE II, III and IV shows experimental results in these different settings.

TABLE II. ACCURACY OF DTV RATIO 1:4

	Total (%)	DT1		DT2		DT3	
		a %	b %	a %	b %	a %	b %
Set1	25.13	10.76	57.18	12.01	52.21	11.00	56.23
Set2	20.68	10.27	50.34	8.95	56.72	11.02	46.71
Set3	13.15	8.62	34.45	8.85	32.70	9.25	29.06

TABLE III. ACCURACY OF DTV RATIO 1:3

	Total (%)	DT1		DT2		DT3	
		a %	b %	a %	b %	a %	b %
Set1	25.13	10.76	57.18	12.01	52.21	11.00	56.23
Set2	20.68	10.27	50.34	8.95	56.72	11.02	46.71
Set3	13.15	8.62	34.45	8.85	32.70	9.25	29.06

TABLE IV. ACCURACY OF DTV RATIO 1:3

	Total (%)	DT1		DT2		DT3	
		a %	b %	a %	b %	a %	b %
Set1	15.43	7.04	54.37	9.01	41.61	8.05	47.83
Set2	11.45	8.12	29.08	8.52	25.59	7.13	37.73
Set3	10.22	7.02	31.31	7.10	30.53	7.25	29.06

From TABLE II to IV, we can see that with different ratio between labeled and unlabeled data, there is great accuracy improvement of DTs after learning from unlabeled data. Moreover, with fewer amounts of labeled data, there is accuracy cut-down of individual DT. But after ensemble of individual DT, we finally nearly equal accuracy of the target model. This fact indicates the proposed DTV algorithm can improve diversity of individual learners by different training set and splitting attributes selection, which helps improve the final prediction accuracy of ensemble learners. The proposed DTV algorithm outperforms a single DT with the same training data set, while it is able to make use of unlabeled data to further improve the generalization ability.

We use identification rate and accuracy of software communication mining to evaluate the direct impact of the result of the proposed algorithm.

$$identification\_rate = \frac{M}{|D|} \tag{8}$$

$$accuracy = \frac{M}{N} \tag{9}$$

M stands for the number of sample that the model gives correct labels. $|D|$ stands for the total number of testing samples. N stands for the total number of samples that are actually correct. TABLE IV illustrates the prediction result on testing data set.

TABLE IV. PREDICTION RESULT ON TESTING DATA SET

		Prediction		
		Not Critical	**Critical**	**Total**
Real	**Not Critical**	1510	48	1558
	Critical	28	674	702
	Total	1538	722	2260

From TABLE IV, it can be concluded that: $identification\_rate = 674/702 = 96\%$

$$accuracy = 674/722 = 93.4\%$$

Both are higher than 90% which indicates it is effective for the communication data mining with 2006-2008 data sets. The proposed DTV algorithm is fit for key issue mining in software development communication.

To evaluate the metric for splitting attribute selection, we also present some experimental result of the performance of all of these metrics. We use the following experiment setting. The first is that we select top k attributes to construct the model, and test classification accuracy on the testing data set. We vary k from 3 to 15. The comparison result is shown as Fig. 3.

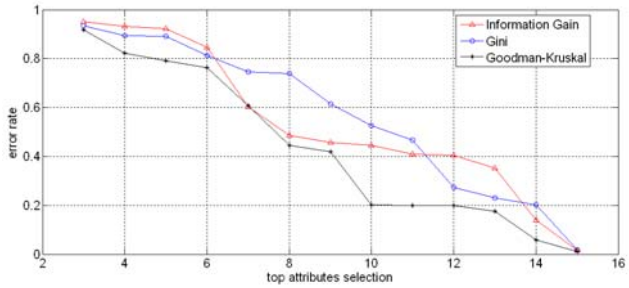

Figure 3. Comparison of Three Metrics for Attribute Selection

From Fig. 3 we can see that three metrics have difference performance in attribute selection. Goodman-Kruskal index give the lowest error rate of all. But for the cases of preserving 3 and 15 attributes, they perform almost equally. This is mainly because they all consider the information gain while Goodman-Kruskal and Gini take some additional information into account.

For the second experiment setting, we aim at testing whether these metrics can make use of unlabeled data. We do the experiment in a semi-supervised learning framework. The size of DT is set to 20 nodes. We train DTs with different ratio between labeled and unlabeled data and record the accuracy on testing data set. Fig. 4 shows the classification accuracy of DTs with different attribute selection metrics.

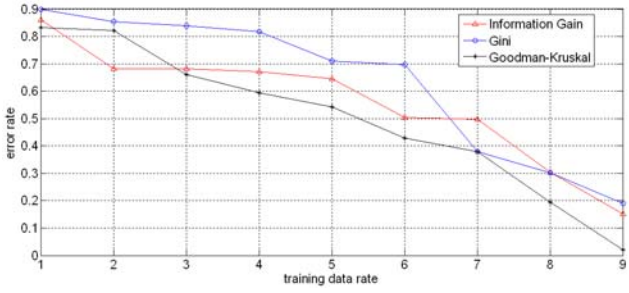

Figure 4. Three Metrics for Semi-Supervised Learning

In Fig. 4 the X-axis is percentage of training data within the whole data set for training (labeled and unlabeled data). We can see that Goodman-Kruskal is of lowest classification error rate and more stable than the other two metrics.

V. Conclusion

In this paper, a framework for intelligent information communication pattern is presented. We propose a co-training styled algorithm DTV to perform semi-supervised learning based on the previous successful tri-training algorithm. The proposed algorithm focuses on mining critical issues in software development communication. Experimental results show that the proposed algorithm is effective to discover underlying pattern in practical environment, which would help managers and administrators make decision.

At the same time, the proposed DTV algorithm is of theoretical benefit. It overcomes the shortcomings of single DT algorithm in improving generalization ability by unlabeled data, which leads to a complete semi-supervised method.

Future work includes further study on the structure of software development communication information, and evaluation of importance of several issues to the final result. Towards different application framework, various well studied data mining and machine learning algorithms can be applied by significant modification of such methods.

Acknowledgment

This work was supported in part by a grant from GDUT Higher Education Research Fund (2009C01, 2009D06).

References

[1] Vineeth Mekkat, Ragavendra Natarajan. Performance characterization of data mining benchmarks. In Proceedings of TKDD, 2010, 3.

[2] Ted E. Senator, On the efficacy of data mining for security applications. In Proceedings of International Conference on Knowledge Discovery and Data Mining, 2009, 6.

[3] Huang Ming, Niu Wenying, Liang Xu. An improved decision tree classification algorithm based on ID3 and the application in score analysis, Chinese Control and Decision conference, 2009 , 1876-1878.

[4] Carson Kai-Sang Leung, Efficient algorithms for mining constrained frequent patterns fromuncertain data, ACM 2009, 6.

[5] Damon Fenacci, Björn Franke, John Thomson. Workload characterization supporting the development of domain-specific compiler optimizations using decision trees for data mining, SCOPES, 2010, 5.

[6] Hilderman, R. J., Carter, C. L., Hamilton, H. J., Cercone, N. Mining market basket data using share measures and characterized itemsets. In Proceedings of PAKDD-98. Melbourne, Australia. 72–86.

[7] Hung, S., Yen, D., Wang, H. Applying data mining to telecom churn management. In: Expert Systems and Applications 31, pp.515-524, 2006.

[8] Abascal, E., Lautre, I., Mallor, F. Data mining in a bicriteria clustering problem. 17. In Proceedings of European Journal of Operational Research 173, pp.705-716, 2006.

[9] Shirish Tatikonda, Srinivasan Parthasarathy. Mining tree-structured data on multicore systems. In Proceedings of the VLDB Endowment, 2009.

[10] Sumathi, S., Sivanandam, S. Introduction to Data Mining and its Applications. Springer-Verlag Berlin Heidelberg, 2006.

[11] Berry, M., Linoff, G. Data Mining Techniques. For marketing, sales and customer. Relationship Management. Wiley Publishing Inc., 2004.
[12] Jiawei Han, Michelin Kamber, Data Mining Concepts and Techniques, Second Edition.
[13] Zhi-Hua Zhou, Ming Li. Tri-training: Exploiting Unlabeled Data Using Three Classifiers. IEEE Trans on knowledge and Data Engineering, 2005.17(11).
[14] TheWinPcapTeam,WimPcnp. http://www.wimpcap.org/docs/docs31/html/main.html,2005.12
[15] Thomas Bernecker, Hans-Peter Kriegel, Matthias Renz, Florian Verhein, and Andreas Zuefle. Probabilistic frequent itemset mining in uncertain databases. In KDD, 2009.
[16] Chen Cheny, Xifeng Yanz, Feida Zhuy, and Jiawei Hany. gapprox: Mining frequent approximate patterns from a massive network. In ICDM, 2007.
[17] Maitreya Natu, Vaishali Sadaphal, Sangameshwar Patil, and Ankit Mehrotra. Mining frequent subgraphs to extract communication patterns in data-centres. In Proceedings of the 12th ICDCN'11, Springer-Verlag, Berlin, Heidelberg, 239-250, 2011.
[18] G. Valiente. Efficient Algorithms on Trees and Graphs with Unique Node Labels. In Studies in Computational Intelligence, Springer- Berlin, vol. 52, pp. 137-149, 2007.
[19] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
[20] G. Brown, J. L. Wyatt, P. Tiňo, Managing diversity in regression ensembles, JMLR, 6, pp. 1621-1650, 2005.
[21] W. Wang and Z.-H. Zhou. Analyzing co-training style algorithms. In Proceedings of the 18th ECML, pp. 454–465, Warsaw, Poland, 2007.
[22] M. Hein and M. Maier. Manifold denoising. In B. Sch¨olkopf, J. C. Platt, and T. Hoffman, editors, Advances in NIPS 19, pages 561–568. MIT Press, Cambridge, MA, 2007.

Gang Zhang, was born in 1979, Guangzhou, received the Master's Degree in computer software and theory in SUN YAT-SEN University (Guangzhou, China) in 2005. Now he is a Ph.D candidate in SUN YAT-SEN University, majored in data mining and machine learning.

He is a lecturer of Faculty of Automation, GuangDong University of Technology (Guangzhou, China), teaching Java Programming, C++ Programming and Computer Network. He has published several papers in international conferences and journals on his major research fields, some of which are indexed by SCI, EI and ISTP. His current research interests include data mining in structured data, semi-supervised learning, multi-instance learning and manifold learning.

Caixian Ye, was born in 1980, Guangzhou, received the Master's Degree in software engineering in HuaZhong University of Science and technology (Wuhan, China) in 2007.

She is an IT project director of NiuTaiLai Communication Equipment Co.Ltd. (Guangzhou, China)，in charge of system analysis, project management, data engineering, project resource arrangement and product / solution design. She is experienced in Java Web development, such as JSP, Servlet and EJB. She is also professional in common database systems such as Oracle, DB2, Sybase and SQL Server.

Chunru Wang, was born in 1968, received the PhD in Control Theory and Control Engineering in South China University of Technology (Guangzhou, China) in 2009.

She is a lecturer of Faculty of Automation, GuangDong University of Technology (Guangzhou, China), teaching Digital Logic, PLC and Control Network. She has published several papers in international conferences and journals in her major research fields, some of which are indexed by EI. Her current research interests control network and WSN.

Xiaomin He, was born in 1961, received the Master's Degree from South China University of Technology (Guangzhou, China).

She is an associate professor of Faculty of Automation, GuangDong University of Technology (Guangzhou, China), teaching Ensemble Language, Configuration Software, and Principles of Computer Organization. She is professional in research of Control Science and Control Engineering. Her current research interests include SCM, e-learning and operation system. She has published several papers in her research fields.

