Preface

The Workshop on the Foundation of Data Mining was held on May 6, 2002, at the Grand Hotel, Taipei, Taiwan. It was held as part of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD-02) which has been recognized as one of the most important events for the KDD researchers in Pacific-Asia area.

The goal of the workshop organizers was to bring together individuals interested in the foundational aspects of data mining to foster the exchange of ideas, with each other as well as with the more application-oriented researchers. The papers in this issue, which comes from Canada, China (including Hongkong), Czech Republic, Japan, New Zealand, Taiwan and USA, indicate the achievement of the goal.

At this point we would like to express our thanks to all the institutions and individuals which actively supported this workshop and make it possible. These include:

- Institute of Information Science, Academia Sinica, Taipei, Taiwan
- Institute of Information & Computing Machinery, Taiwan
- All contributed authors

We also especially thank Dr. Tsan-sheng Hsu（徐讚昇）for providing us the opportunity to publish the proceeding as the special issue.

We hope that the workshop will be valuable and fruitful to all participants, no matters they would like to uncover the fundamental principles behind data mining or apply the theories to the practical application problems.

April, 2002

Tsau-Young Lin（林早陽）

Churn-Jung Liau（廖純中）

Workshop Organization

Program co-Chairs

T. Y. Lin (San Jose State University , USA)

C. J. Liau (Academia Sinica, Taiwan)

Program Committee

N. Cercone (Waterloo University, Canada)

I. J. Chiang (Index Software, USA)

Le Gruenwald (University of Oklahoma, USA)

Xiaohua Hu (DMW Software, USA)

W. Lee (Georgia Institute of Technology, USA)

T. Y. Lin (San Jose State University, USA)

Larry Kerschberg (George Mason University, USA)

Ernestina Menasalv(Campus de Montegancedo, Spain)

L. Mazlak (UC-Berkeley, USA)

M. C. Shan (Hewlett-Packard Labs)

Z. Ras (University of North Carolina at Charlotte, USA)

B. Thurasingham (National Science Foundation, USA)

Shin-Mu Tseng (National Cheng-Kung University, Taiwan)

S. Tsumoto (Shimane Medical University, Japan)

Y. Y. Yao (University of Regina, Canada)

A. Wasilewska (State University of New York, USA)

N. Zhong (Maebashi Institute of Technology, Japan)

Table of Contents

Intelligent Multi-Objective Evolutionary Algorithm for Editing Minimum Reference Set

Jian-Hung Chen
Department of Information Engineering,
Feng Chia University,
Taichung, Taiwan 407, Republic of China
jh.chen@ieee.org

Shinn-Ying Ho
Department of Information Engineering,
Feng Chia University,
Taichung, Taiwan 407, Republic of China
syho@fcu.edu.tw

Abstract Editing a minimum reference set of training patterns plays an important role for consequently constructing a compact classification system so as to reduce the computation load in the operational phase. Various approaches were proposed for finding a small number of reference patterns from a large number of given patterns considering an overall criterion. In this paper, an intelligent multi-objective evolutionary approach is proposed to editing compact reference sets for nearest neighbor classification considering multiple criteria. An empirical study of various multi-objective evolutionary algorithms demonstrated the efficiency of the proposed approach in terms of both classification rate and number of patterns of the reference set.

1 Introduction

In data mining researches, generation of classification rules is one of the most important issues. Classification is to divide training patterns into subsets according to their attributes such that most of the patterns in the same subset belong to the same class. Therefore, selecting a minimum reference set of training patterns plays an important role for consequently constructing a compact classification system so as to reduce the computation load in the operational phase. During the past decades, various approaches were proposed for finding a compact set of reference patterns such as genetic algorithms (GAs) approaches [1,2] and fuzzy-based design approaches [3,4]. Due to its robustness, theoretical elegance, and feasibility of realization, the *k*-Nearest Neighbor (*k*-NN) rule continues to be one of the most widely used classification techniques. The approaches to reducing the number of training patterns for *k*-NN classification can be classified into two classes: selection and replacement approaches [4]. The selection-based approach using GA-based algorithms can obtain the minimum reference set and higher classification accuracy, compared with other approaches, as pointed out in [1]. In general, two objectives are addressed for solving minimum reference set problems (MRSPs). The first objective is the highest classification accuracy. The second one is the smallest reference set which can reduce the computation load in the operational phase. Two fundamental issues of solving MRSPs using GA-based algorithms are as follows.

1. Weight Selection Problem. Generally, weighted-sum approaches are the most widely used techniques for MRSPs. The general fitness function is defined as follows [2]:

   maximize

   $$fitness(S) = W_{NCP} \cdot NCP(S) - W_s \cdot |S| \quad , \qquad (1)$$

   subject to $S \subset Z$,

   where Z is a set of training patterns, S is a subset of Z, NCP(S) is the number of correctly classified patterns by S, |S| is the numbers of training patterns in S, respectively. W_{NCP} and W_S are positive constant weights. However, weighted-sum approaches has been criticized that prior domain knowledge is required to determine the appropriate weights, and the solution quality is sensitive towards the weights.
2. Large Training Set Problem. In the real-world MRSPs, the size of the training set is larger than the one used in the experiments of the research, reported in the literature. Traditional GA-based algorithms suffer from both the low convergence speed and low accuracy for large-scale problems. It results in the low robustness and reliability of the classifier design. Generally, prior knowledge or heuristic techniques are needed for MRSPs.

Recently, several multi-objective evolutionary algorithms are proposed to solve multi-objective optimization problems directly, and presented more promising results than single-objective optimization techniques theoretically and empirically [5-10]. Therefore, the aim of this paper is to investigate the ability of multi-objective evolutionary algorithms in solving MRSPs. Meanwhile, an intelligent multi-objective evolutionary algorithm (IMOEA) [11,12] is applied to solve multi-objective MRSPs directly. It will be shown empirically that IMOEA outperforms existing multi-objective evolutionary algorithms in solving multi-objective MRSPs.

The organization of this paper is as follows. A brief summary of multi-objective optimization and the mathematical formulation of two-objective MRSPs are described in Section 2. Section 3 illustrates the proposed IMOEA. Section 4 compares IMOEA with existing multi-objective evolutionary algorithms by applying them to solve two-objective MRSPs. Section 5 concludes this paper.

2 Multi-Objective Optimization and Minimum Reference Set Problems

2.1 Multi-Objective Optimization

Mathematically, MOOPs can be repsented as the following vector mathematical programming problems:

$$\text{minimize } F(X) = \{f_1(X), f_2(X), \cdots, f_I(X)\}, \quad (2)$$

where X denotes a solution, $f_i(X)$ is generally a nonlinear objective function. Without loss of generality, the minimization problem is assumed in this paper unless otherwise specified. When the following inequalities hold between two solutions X_1 and X_2, X_2 is a *non-dominated solution* and is said to *weakly dominate* X_1:

$$\forall i : f_i(X_1) \geq f_i(X_2) \quad (3)$$

When the following inequalities hold between two solutions X_1 and X_2, X_2 is a *non-dominated solution* and is said to *dominate* X_1:

$$\forall i : f_i(X_1) \geq f_i(X_2) \text{ and } \exists j : f_j(X_1) > f_j(X_2). \quad (4)$$

A feasible solution X^* is said to be a *Pareto-optimal solution* if and only if there does not exist a feasible solution X where X dominates X^*, and the corresponding vector of Pareto-optimal solutions is called *Pareto front*.

2.2 Formulation of Minimum Reference Set Problems

Let us consider an a-class pattern classification problem in an n-dimensional pattern space $[0, 1]^n$. Assumed that m training patterns $x_p = (x_{b1}, \ldots, x_{bn})$, b=1, 2,…, m, are given from a class (a<<m), and the set of these m patterns is denoted as $Z=\{x_1, \ldots, x_m\}$. The aim of MRSPs is to find a subset S ($S \subset Z$) that minimizes the number of reference patterns in S and maximizes the classification rate CR(S). Since any subset of S of the m training patterns is a feasible solution of this problem, the total number of feasible solutions is 2^m, which means the size of search space increase exponentially with the number of training patterns.

Due to there is a trade-off between the classification rate and the size of the reference set S, this problem can be formulated as the following multi-objective optimization problem:

find S, such that minimizes |S| and maximizes CR(S), subject to $S \subset Z$, (5)

where |S| is the number of reference patterns, NCP(S) is the number of correctly classified patterns by S, and the classification rate CR(S) is calculated by using equation (6).

$$CR(S) = \frac{NCP(S)}{m} \quad (6)$$

3 Intelligent Multi-Objective Evolutionary Algorithm

An intelligent multi-objective evolutionary algorithm (IMOEA) proposed by us is applied to solve multi-objective MRSPs. The advantages of IMOEA are:

1. Elitism: IMOEA incorporates with two populations: the current population and the external elite set.
2. Fitness assignment strategy: The generalized Pareto-based scale-independent fitness function (GPSIFF) can assign discriminative fitness value to individuals.
3. Intelligent crossover (IC): IC is introduced to improve the performance of IMOEA on solving problems with a large number of parameters.

The representation of the chromosome is presented in Section 3.1. The fitness assignment strategy and IC are described in Sections 3.2 and 3.3, respectively. The flow of IMOEA is provided in Section 3.4.

3.1 Chromosome Representation

A subset S of the m training patterns encoded using a binary string consisting of m bits as $S=s_1s_2\ldots s_m$, where s_p=1 denote that p-th pattern of Z is included in subset S, and s_p=0 otherwise.

3.2 Fitness Assignment

The fitness assignment strategy of IMOEA uses a generalized Pareto-based scale-independent fitness function (GPSIFF) considering the quantitative fitness values in Pareto space for both dominated and non-dominated individuals. It is assumed that no information on the preference among objectives is available.

Let GPSIFF fitness value be a tournament-like score obtained from all participant individuals. The fitness value of a solution X can be given by the following score function:

$$\text{score(X)} = p - q + c, \quad (7)$$

where p is the number of solutions which can be dominated by X, and q is the number of solutions which can dominate X. The constant c is used to obtain the positive fitness value. Figure 1 illustrate the example of fitness values of twelve participant individuals for a bi-objective optimization problem (c=12). For example, considering the individual A with fitness value 13, in the rectangle formed by A, two individuals dominates A (q=*2*) and three individuals is dominated by A (p=*3*). Therefore, the fitness of A is *3-2+12=13*.

The merits of GPSIFF are as follows:

1. Simplicity. The goodness of each individual is evaluated by considering the Pareto dominance relationships of both dominated and non-dominated individuals.
2. Generality. GPSIFF makes direct use of general Pareto dominance relationships to evaluate the performance of each participant individual.
3. Effectiveness. GPSIFF intuitively reflects the idea of preferring individuals near the Pareto-optimal front. By means of GPSIFF, each individual has an accurate fitness value that is helpful in the selection step of IMOEA.

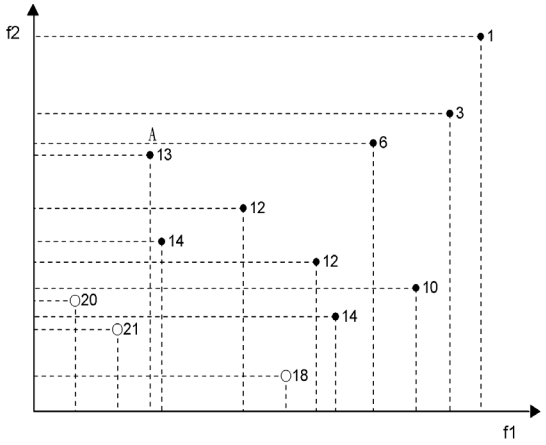


Fig. 1. Fitness values of the participant individuals with c=12 in the objective space.

3.3 Intelligent Crossover

In the conventional crossover operations of GAs, two parents generate two children with a combination of their chromosomes using a *randomly* selected cut point. The merit of IC is that the systematic reasoning ability of orthogonal experimental design (OED) [14,15] is incorporated in the crossover operator to economically estimate the contribution of individual genes to a fitness function, and consequently intelligently pick up the better genes to form the chromosomes of children. Theoretically analysis and experimental studies for illustrating the superiority of IC with the use of OA and factor analysis can be found in [11-13].

3.3.1 OA and Factor Analysis

OA is a factional factorial matrix, which assures a balanced comparison of levels of any factor or interaction of factors. It is a matrix of numbers arranged in rows and columns where each row represents the levels of factors in each experiment, and each column represents a specific factor that can be changed from each experiment. The array is called orthogonal because all columns can be evaluated independently of one another, and the main effect of one factor does not bother the estimation of the main effect of another factor. A two-level OA used in IC is described as follows. Let there be γ factors with two levels for each factor. The total number of experiments is 2^{γ} for the popular "one-factor-at-a-time" study. The columns of two factors are orthogonal when the four pairs, (1,1), (1,2), (2,1), and (2,2), occur equally frequently over all experiments. Generally, levels 1 and 2 of a factor represent selected genes from parents 1 and 2, respectively. To establish an OA of γ factors with two levels, we obtain an integer $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$, build an orthogonal array $L_{\omega}(2^{\omega-1})$ with ω rows and (ω-1) columns, use the first γ columns, and ignore the other (ω-γ-1) columns. Table 1 illustrates an example of OA $L_8(2^7)$. The algorithm of constructing OAs can be found in [16]. OED can reduce the number of experiments for factor analysis. The number of OA experiments required to analyze all individual factors is only ω or O(γ).

Table 1: Orthogonal array $L_8(2^7)$

	Factors							
Exp. no.	1	2	3	4	5	6	7	Function Evaluation value
1	1	1	1	1	1	1	1	y_1
2	1	1	1	2	2	2	2	y_2
3	1	2	2	1	1	2	2	y_3
4	1	2	2	2	2	1	1	y_4
5	2	1	2	1	2	1	2	y_5
6	2	1	2	2	1	2	1	y_6
7	2	2	1	1	2	2	1	y_7
8	2	2	1	2	1	1	2	y_8

After proper tabulation of experimental results, we can further proceed *factor analysis* to determine the relative effects of various factors. Let y_t denote the positive function evaluation value of experiment t, t = 1, 2, … ω. Let $Y_t = y_t$ $(1/y_t)$ if the objective function is to be maximized (minimized). Define the main effect of factor j with level k as S_{jk} :

$$S_{jk} = \sum_{t=1}^{\omega} Y_t \cdot F_k , \qquad (8)$$

where F_k = 1 if the level of factor j of experiment t is k; otherwise, F_k = 0. Notably, the main effect reveals the individual effect of a factor. The most effective factor j has the largest main effect difference (MED) $| S_{j1} - S_{j2} |$. If $S_{j1} > S_{j2}$, the level 1 of factor j makes a better contribution to the optimization function than level 2 does. Otherwise, level 2 is better. After the better level of each factor is determined, a combination consisting of factors with better levels can be efficiently derived.

3.3.2 Procedures of Intelligent Crossover

Two parents breed two children using IC at a time. Let the number of participated genes in a parent chromosome beγ. How to use OA and factor analysis to achieve IC is described as the following steps:

Step 1: Ignore the loci having identical values in two parents such that the chromosomes can be temporally shortened resulting in using a small OA table.

Step 2: Adaptively divide the parent chromosomes into γ pairs of gene segments where each gene segment is treated as a factor.

Step 3: Use the first γ columns of OA $L\omega(2^{\omega-1})$ where $\omega = 2^{\lceil \log_2(\gamma+1) \rceil}$.

Step 4: Let levels 1 and 2 of factor j represent the *j*th gene segment of a chromosome coming from parents respectively.

Step 5: Simultaneously evaluate the fitness values y_t of the γ combinations (by-products) corresponding to the experiments t, where t = 1, 2, …,ω.

Step 6: Compute the main effect S_{jk} where $j = 1, 2, ..., \gamma$ and $k = 1, 2$.

Step 7: Determine the better one of two levels for each gene segment. Select level 1 for the *j*th factor if $S_{j1} > S_{j2}$. Otherwise, select level 2.

Step 8: The chromosome of first child is formed using the combination of the better gene segments from the derived corresponding parents.

Step 9: Rank the most effective factors from rank 1 to rank γ. The factor with large (MED) has higher rank.

Step 10: The chromosome of second child is formed similarly as the first child except that the factor with the lowest rank adopts the other level

3.4 Intelligent Multi-objective Evolutionary Algorithm

IMOEA uses an elite set E whose maximum capacity is E_{max}. The elite set E maintains the best non-dominated solutions among all non-dominated solutions generated so far. The individuals in E will participate in the selection step of IMOEA. The proposed approach, IMOEA, is described as follows:

Step 1: (Initialization) Randomly generate an initial population of N_{pop} individuals and create an empty elite set E and an empty temporary elite set E'.

Step 2: (Update Elitism) Copy the non-dominated solutions in current population and E' to E. Delete the dominated solutions in E and empty E'. If the number of individuals exceeds E_{max}, reduce E by discarding individuals randomly.

Step 3: (Evaluation) Evaluate the fitness values of all individuals by using the GPSIFF.

Step 4: (Selection) Randomly select N_{pop}-N_{ps} individuals from the population and N_{ps} individuals from E to form a new population, where $N_{ps} = N_{pop} \cdot p_s$ and p_s is a selection proportion. If N_{ps} is greater than the number N_E of individuals in E, let $N_{ps} = N_E$.

Step 5: (Recombination) Perform IC operations for all selected pairs of parents with the recombination probability p_c. Copy non-dominated by-products to a temporary elite set E'.

Step 6: (Mutation) Apply a conventional mutation operator (e.g., bit-inverse mutation) to the population with a mutation probability p_m.

Step 7: (Termination test) If a stopping condition is satisfied, end the algorithm. Otherwise, go to Step 2.

4 Experiment Results

In order to investigate the performance of IMOEA, four representative multi-objective evolutionary algorithms, SPEA[13], NSGA[12], NPGA[11] and VEGA[10], are selected and compared in solving multi-objective MRSPs. The coverage ratio of two set (A, B) [13] is used to be a performance metric of different algorithms. The coverage ratio of set (A, B) is calculated as follows:

$$C(A,B) = \frac{\text{the number of solution of B weakly dominated by A}}{\text{the number of solution of B}}$$

The value C(A, B) = 1 means that all individuals in B are dominated by A. The opposite, C(A, B)=0, denotes that none of individuals in B are dominated by A.

Two test data sets, iris data and wine data, are used in the experiments. All the data are available via anonymous ftp from *ftp.ics.uci.edu/pub/machine-learning-databases*. All the training patterns are used as test patterns in designing a compact 1-NN classifier system. The parameter settings of IMOEA are as follows: N_{pop}=20, ther upperbound of TNONS=20, P_s=0.2, P_c=0.8, P_m=0.05, and factor number of OA (γ) equals the total number of patterns in the data set (iris: 150, wine: 178). The parameter settings of SPEA, NSGA, NPGA and VEGA are: N_{pop}=50, P_c=0.8, P_m=0.05, t_{dom}=10 and σ_{share}=0.49. The population size and the external population size of SPEA are 40 and 10, respectively. The sharing factor σ_{share} of NSGA is 65 for iris data set and 70 for wine data set.

4.1 Experiment 1- Iris Data

There are 50 patterns with four attributes in each of three classes, i.e., 150 patterns in total. All the algorithms were performed thirty independent runs for each data set under the same number of function evaluations N_{eval}=20000. The direct comparison of each runs for the different algorithms based on the C measure is depicted in figure 2. Simulation results out of thirty runs are summarized in figure 3.

Generally, the simulation results show that IMOEA and SPEA are better than NSGA, NPGA and VEGA, while

NSGA, NPGA and VEGA suffer low convergence speed and may be trapped in local optimum. Considering the the distribution of non-dominated solutions and the quality of solutions, it shows that IMOEA obtained a well-distributed Pareto front and dominate all the non-dominated solutions of the other test algorithms. The non-dominated solutions (|S|, CR(S)) of IMOEA out from 30 runs are as follows: (1, 33,33%), (2, 66.67%), (3, 98.00%), (4, 98.67%), (6, 99.33%) and (11, 100%). Moreover, from figure 2 and figure 3, the results illustrate that IMOEA is robust and is capable of generating good non-dominated solution efficiently in this test.

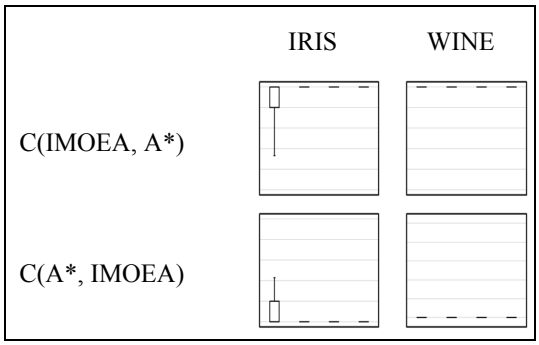


Fig. 2. Box plots based on the C measures for MRSPs. Each rectangle contains four box plots representing the distribution of the C measures for a certain ordered pair of algorithms. A* are SPEA, NSGA, NPGA and VEGA, respectively. The scale is 0 at the bottom and 1 at the top per rectangle.

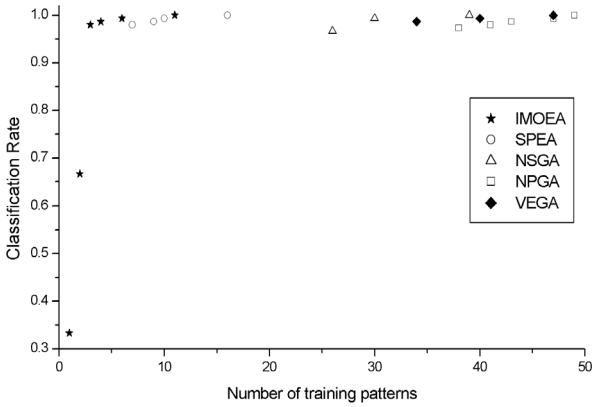


Fig. 3. Simulation results out of 30 runs for iris data under N_{eval} = 20000.

4.2 Experiment 2 – Wine Data

The wine data consist of 178 patterns with 13 continuous features from three classes. The test algorithms were performed thirty independent runs under the same number of function evaluations N_{eval}=30000. The distribution of C measure for each runs is depicted in figure 2. The results out of thirty runs are shown in figure 6. Figure 2 shows that the non-dominated solutions obtained by IMOEA dominate all the non-dominated solutions of SPEA, NSGA, NPGA and VEGA in every run. From figure 4, the results reveal that IMOEA and SPEA outperform other competitive algorithms, and IMOEA achieves the best assessments among all the test algorithms. The non-dominated solutions (|S|, CR(S)) of IMOEA out from thirty runs are (1, 33.14%), (2, 66.29%), (3, 91.57%), (4, 98.31%), (5, 98.88%), (7, 99.44%) and (8, 100%).

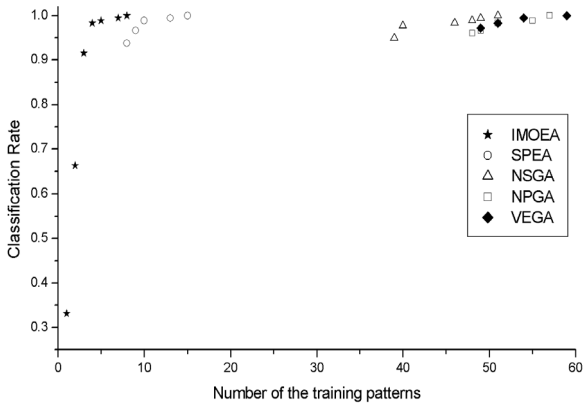


Fig. 4. Simulation results out of 30 runs for wine data under N_{eval} = 30000.

4.3 Discussion of Experimental Results

The objectives of MRSPs are to edit a minimum reference set and to maximize the classification rate. It is well recognized that, when the size of the reference set increases, the classification rate may increase and thus leads to a biased search space. On the other hand, while the size of the reference set is small, the interaction between the reference set and test patterns will become complicated. In order to visualize the landscape of the MRSP, all the solutions generated during the experiments of MRSP with wine data set, are collected and then plotted in bi-objective space, as shown in figure 5. It can be seen that the MRSP of the wine data is likely to have a discontinuous, non-uniform and biased search space. From figures 2-4, it is shown that only IMOEA is capable of generating a better and well-distributed Pareto front in these discontinuous, non-uniform and biased search spaces. Except IMOEA and SPEA,

NSGA, NPGA and VEGA suffers from both the low convergence speed and low accuracy.

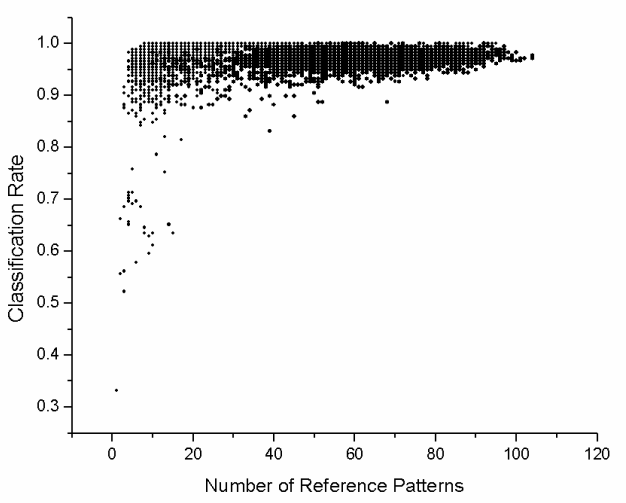


Fig. 5. Scatter plots of solutions in solving MRSP with the wine data set. The darker region has higher density of solutions.

5 Conclusions

In this paper, we examined the ability of several well-known multi-objective evolutionary algorithms, and proposed an intelligent multi-objective evolutionary algorithm (IMOEA) in editing a minimum reference set of training patterns considering multiple objectives. In present study, IMOEA illustrates the strong superiority to existing algorithms, and yields widely distributed Pareto fronts close to the Pareto-optimal fronts. High performance of IMOEA can be obtained without use of traditional auxiliary techniques such as local search, various mutation strategies, problem-dependent heuristic strategies, etc. We believe that the auxiliary techniques, which can improve performance of conventional evolutionary algorithms, can also improve performance of IMOEA. The suitability of parallel implementation for IC is another advantage of IMOEA.

Concerning the large-scale optimization problems with different features, further investigations such as incorporating feature selection in solving MRSPs and using other techniques for accelerating the convergence time of multi-objective GAs.

References

[1] Kuncheva, L. I., Bezdek, J. C.: Nearest prototype classification clustering, genetic algorithms, or random search?. IEEE Trans. SMC-Part C: Application and Reviews, **28**(1) (1998) 160-164

[2] Nakashima, T., Ishibuchi, H.: GA-Based Approaches for Finding the Minimum Reference Set for Nearest Neighbor Classification. In Proc. of IEEE Conf. on Computational Intelligence (1998) 709-714

[3] Yang M.-S., Chen C.-H.: On the edited fuzzy k-nearest neighbor Rule. IEEE Trans. on SMC-part B: Cybernetics **28**(3) (1998) 461-466

[4] Bezdek, J. C. et al.: Multiple-prototype classifier design. IEEE Trans. on SMC- Part C: Applications and Reviews. **28**(1) (1998) 67-79

[5] Goldberg, D. E.: Genetic Algorithms in Search, Optimization and Machine Learning. Addison – Wesley Publishing Company (1989)

[6] Deb, K.: Multi-Objective Optimization Using Evolutionary Algorithms. John Wiley & Sons (2001)

[7] Schaffer J. D.: Multi-objective optimization with vector evaluated genetic algorithms. In Proc. of 1st Int. Conference Genetic Algorithms (1985) 93-100

[8] Horn, J., Nafplotis N., Goldberg, D. E.: A niched Pareto genetic algorithm for multi-objective optimization. In Proc. of 1st IEEE Int. Conference of Evolutionary Computation (1994) 82-87

[9] Srinivas, N., Deb, K.: Multiobjective optimization using non-dominated sorting in genetic algorithms. Evolutionary Computation **2**(3) (1994) 221-248

[10] Zitzler, E., Thiele, L.: Multiobjective evolutionary algorithms: A comparative case study and the strengthen Pareto approach. IEEE Trans. on Evolutionary Computation, **3**(4) (1999) 257-271

[11] Ho, S.-Y., Chang, X.-I.: An efficient generalized multiobjective evolutionary algorithm. In GECCO-99: Proc. of the Genetic and Evolutionary Computation Conference (1999) 871-878

[12] Chen, J.-H., Ho, S.-Y.: Evolutionary multi-objective optimization of flexible manufacturing systems. In GECCO-2001: Proc. of the Genetic and Evolutionary Computation Conference (2001) 1260-1267

[13] Ho, S.-Y., Chen, Y.-C.: An efficient evolutionary algorithm for accurate polygonal approximation. Pattern Recognition **34** (2001) 121-133

[14] Dey, A.: Orthogonal Fractional Factorial Designs. New York, Wiley (1985)

[15] Hicks, C. R., Turner, K. V. Jr.: Fundamental Concepts in the Design of Experiments. 5th edn. Oxford University Press, New York (1999)

[16] Zhang, Q., Leung, Y.-W.: An orthogonal genetic algorithm with quantization for global numerical optimization. IEEE Trans. on Evolutionary Computation, **5**(1) (2001) 41-53

Apply Fuzzy Classifications to Colon Polyp Screening

I-Jen Chiang, Ming-Jium Shieh, Jane Yung-jen Hsu, Jau-Ming Wong

Abstract—

To deal with hightly uncertain and noise data, for example, biochemical laboratory examinations, a classifier is required to be able to classify an instance into all possible classes and each class is associated with a degree which shows how possible an instance is in that class. According to these degrees, we can discriminate the more possible classes from the less possible classes. The classifier or a expert can pick the most possible one to be the instance class. However, if their discrimination is not distinguishable, it is better that the classifier should not make any prediction, especially when there is incomplete or inadequate data. A *fuzzy classifier* is proposed to classify the data with noise. Instead of determining a single class for any given instance, *fuzzy classification* predicts the degree of *possibility* for every class.

Adenomatous polyps are widely accepted to be precancerous lesions and will degenerate into cancers ultimately. Therefore, it is important to generate a predictive method that can identify the patients who have obtained polyps and remove the lesions of them. Considering the uncertainties and noisy in the biochemical laboratory examination data, *fuzzy classification trees*, which integrate decision tree techniques and fuzzy classifications, provide the efficient way to classify the data in order to generate the model for polyp screening.

Keywords— **Fuzzy Classifications, Polyp Screening, Fuzzy Classification Trees, Fuzzy Entropy.**

I. Introduction

Colorectal cancer (CRC) has become one of the leading causes of cancer death in Taiwan, with nearly 2900 new cases and 1900 deaths reported each year. Despite advances in treatment, early detection can probably reduce CRC mortality more than any other approaches. Therefore, it is important to develop a cost-effective cancer screening policy in the hopes of reducing CRC mortality by detecting lesions at any early, curable stage.

The prevalence of adenomatous polyp varies geographically in parallel with the incidence of colorectal cancer and an increasing risk of colorectal cancer [34], [37]. The concept is now widely accepted that adenomas are precancerous lesions and will degenerate into cancers ultimately. Nowadays, the majority of the pathogeneses of the colorectal cancer are attributed to the adenoma-adenocarcinoma sequence. Hence, the identification and removal of the precancerous lesion, an adenomatous polyp, has significant clinical implications and is now commonly recommended for the control of CRC. Endoscopy is considered the most sensitive diagnostic modality for detection of colorectal polyps. However, the effort and eventual cost involved based on this surveillance strategy are potentially enormous and not practical, except for high-risk groups. Owing to the shortage of medical resources at present, it is important to develop a most cost-effective and safe screening method to predict the existence of adenomatous polyps.

I-Jen Chiang is with the Graduate Institute of Medical Informatics, Taipei Medical University Taipei, Taiwan. E-mail: ijchiang@tmu.edu.tw.

Ming-Jium Shieh is with the Department of Biomedical Engineering, National Taiwan University Taipei, Taiwan.

Jane Yung-jen Hsu is with the Department of Computer Science and Information Engineering, National Taiwan University Taipei, Taiwan.

Jau-Ming Wong is with the Department of Biomedical Engineering, National Taiwan University Taipei, Taiwan.

In order to determine the predictive value of the risk factors related to the existence of rectosigmoid colon polyps, physicians evaluate all putative risk factors obtained from checkup items. Bias inevitably occurs from this assumption, in that only factors that have been selected can be shown to have association. A collection of physical checkup data with the patients who underwent sigmoidoscopy enrolled for the polyp screening analysis.

Some classification techniques, e.g. decision trees [12], [20], [21], [22], [29], decision lists [10], [31] work well for pattern recognition and process control. Here, we choose these techniques to apply to colon ploy screening analysis [37]. Through a classification method, a classifier can be constructed from a medical database. This classifier is able to predict which class a new instance is. Many techniques, such as Bayesian classifiers [11], decision trees [29], neural networks [32], rule based learners [24], [28], etc., have been applied to producing classifiers for medical decision support systems [38]. A classifier is produced on a set of training instances and a decision is made automatically on each new instance based upon a forecast of the classification of the instance. Unfortunately, it is hard to clearly classified the data because of the uncertainties and noise. Obviously, a vague classification method is needed to deal with such problems. That is, a classifier is able to classify an instance into all possible classes and each class is associated with a degree which shows how possible an instance is in that class. According to these degrees, we can discriminate the more possible classes from the less possible classes.

Fuzzy decision trees [3], [14], [39], which integrate decision tree techniques and fuzzy classifiers, provide the simple and efficient way to generate the classification model that can suffer from inadequately or improperly expressing and handling the vagueness and ambiguity associated with human thinking and perception [40]. Pedrycz and Sosnowski [25] pointed out that the concept of fuzzy granulation realized via context-based clustering is aimed at the discretization process. For the sake of vagueness, fuzzy classifications are issued. Through it, we can calculate the degree of possibility that the instance belongs to any of the classes. Using information-based measures, there is no need to generate multiple classification trees. Therefore, it requires less time and space than decision forests [19].

This paper introduces to use the fuzzy classification approach to design a medical decision support system for polyp screening. Section 2 gives the definition of classifications and problems of traditional classifiers. The definitions of *fuzzy classifications* and *fuzzy classification trees* are presented in section 3. The attribute selection measures are defined in section 4. Section 5 describes the basic algorithm for constructing a FCT from a data set. The classification process is shown in section 6. The empirical results compared FCT with C4.5 on polyp screening and some UCI repository datasets are shown in section 6, followed by the conclusion.

II. Fuzzy Classifications

Fuzzy classifications are proposed to overcome the difficults that conventional classifiers cannot handle multiple instances with overlapping attribute values that belong to dif-

ferent classes, but keep the efficient as decision tree classifiers.

Definition 1: Given A *fuzzy classifier* $\mathbf{F}$ for a given classification problem $(\mathcal{X}, \mathcal{C})$ defines a total function

$$\mathbf{F} : \mathcal{X} \rightarrow \{\langle p_1, \ldots, p_n\rangle | p_i \in [0,1]\}$$

where p_i is the *possibility* that a given instance $\mathbf{x}$ belongs to class C_i.
For ease of presentation, the function $\mathbf{F}$ is sometimes represented as a vector of functions

$$\langle \wp_1, \wp_2, \ldots, \wp_n\rangle,$$

where $\wp_i$ is a possibility function $\mathcal{X} \rightarrow [0,1]$. For any given instance $\mathbf{x}$, the relation $\wp_i(\mathbf{x}) > \wp_j(\mathbf{x})$ indicates that it is more likely for the instance $\mathbf{x}$ to be in class C_i.

A fuzzy classifier can be readily implemented by a tree structure, such as fuzzy decision trees [3], [9], [14], [39], [40]. In general, those methods can separated into two types, pre-fuzzification and post-fuzzification. However, no matter what the type of fuzzy decision tree methods is, they all unavoid two phases processing to generate the decision rules. They either prefuzzify the data according to domain knowledge or post-fuzzify the decision rules generated by the decision tree methods by some tuning methods. They do not concern the distribution of the data that can make an improper classifications. Therefore, fuzzy classification trees [7], [13], [8] have been presented to solve those problems on pre-fuzzification and post-fuzzification.

This section briefly presents the basic definitions of *fuzzy classification trees* (FCTs). Figure 1 shows a sample FCT that classifies instances into two classes C_1 and C_2.

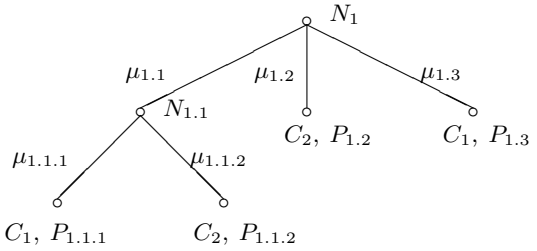


Fig. 1. A sample FCT with $\mathcal{C} = \{C_1, C_2\}$

Let $\mathcal{L}$ be the set of all labels that is defined by a labeling function that uniquely assigns a label to each node and each branch.

Definition 2: Given an *FCT*, each node n in the tree $\mathcal{T}$ is given a label:

$$\text{Label}(n) = \begin{cases} 1 & \text{if } n \text{ is the root;} \\ \text{Label}(n').i & \text{if } n \text{ is the } i\text{th} \\ & \text{child of node } n'. \end{cases}$$

where . is the concatenation operator.

N_L denote the node labeled by $L \in \mathcal{L}$, and B_L denote the branch leading into node N_L. Each non-terminal node in the tree is associated with a test, and the resulting branches, $B_{L.i}$, is associated with a membership function

$$\mu_{L.i} : \mathcal{X} \rightarrow [0,1].$$

Intuitively, the membership defines the degree of possibility that an instance $\mathbf{x} \in \mathcal{X}$ should be propagated down the branch. In our implementation, each test at a node is tested on a single attribute. Therefore, the membership function is defined over the projection on that attribute, that is, $\texttt{projection}(\mathcal{X}, a_L)$, i.e. the domain of the testing attribute $a_L \in A$.

Suppose each node N_L is associated with a class C_L and a possibility function P_L.

Definition 3: Let the label for the parent node of N_L is denoted to be $\hat{L}$. The possibility function $P_L : \mathcal{X} \rightarrow [0,1]$ is defined by composing the membership functions along the path from the root to node N_L. That is,

$$P_L = \begin{cases} 1 & \text{if } N_L \text{ is the root node;} \\ P_{\hat{L}} \otimes \mu_L & \text{if } N_{\hat{L}} \text{ is the parent of } N_L. \end{cases}$$

The composition operator $\otimes$ is defined in terms of some valid operation for combining two membership functions.
Several composition operators, e.g. fuzzy sum, fuzzy product, and fuzzy max, are supported in our implementation. For example,

$$P_L(\mathbf{x}) = P_{\hat{L}}(\mathbf{x}) + \mu_L(\mathbf{x})$$

when the fuzzy sum operator is applied.

Given any instance $\mathbf{x}$ at a terminal node N_L in an FCT, it is classified into class C_L with a possibility $P_L(\mathbf{x})$. As was shown in Figure 1, multiple terminal nodes may be associated with the same class. It follows that an FCT defines a unique fuzzy classifier

$$\mathbf{F} = \langle \wp_1, \ldots, \wp_n\rangle$$

such that the possibility for an instance belonging to class C_i is the *maximum* over all the possibility values at terminal nodes classified as C_i. That is, for $1 \leq i \leq n$,

$$\wp_i(\mathbf{x}) = \max\{P_L(\mathbf{x}) | N_L \text{ is a leaf } \wedge C_L = C_i\}.$$

III. Information-Base Measure

At each node of a fuzzy classification tree, an attribute is used to calculate the membership that an instance should be split into a branch. This attribute is decided at the learning time, that may create the best data clustering at the current node. The *goodness of split* is an important criterion for selecting attributes to expand a fuzzy classification tree. Some information-based measures have been widely applied to classifications for evaluating the goodness of split [1], [4], [18], [26], [29].

In order to evaluate the uncertainties in the data, Shannon has defined the information entropy function that refers to the Boltzmann's H theorem in statistical mechanics [36]. The foundation of Shannon's formula is based on probability theory. Quinlan [29], etc., have used such kind of uncertainty evaluation methods to construct tree classifiers. These information-based evaluation methods can be applied to the construction of probabilistic fuzzy classification trees. However, those methods are well-defined on probability.

According to the original probabilistic entropy defined by Shannon [36] and fuzzy entropy function defined by De. Luca and Termini [16], the information-based measure should satisfy the following criteria. Let The possibility $\wp_i$ for each i define the possibility of an instance, where $\wp_i \in [0,1]$. Five criteria [7], [8] required for attribute selection in terms of an information-based measure of FCT are listed as follows.

[Property 1] Function $H(\wp_1, \wp_2, \ldots, \wp_n)$ should be continuous in $\wp_i$. This property prevents a situation in which a very small change in $\wp_i$ would produce a large (discontinuous) vibration.

[Property 2] Function H must be 0 if and only if all the $\wp_i$ but one are zero. When all but one is possible, there exists no uncertainty in the data.

[Property 3] Function H is the maximum value if and only if the $\wp_i$ are equal because there exists the most uncertainties in the data. That is, no matter what all the $\wp_i$ are, the largest uncertainties happened when all the $\wp_i$ are of the same value.

[Property 4] Function H is a nonnegative valuation on the $\wp_i$.

[Property 5] In order for the purpose that an attribute selection is to reduce the uncertainties in the data, it is necessary that if a choice is broken down into several successive choices, the original H should be no less than the weighted sum of the individual values of H. This property prevents the data been classified to be worse than before.

We can define our fuzzy entropy functions that follow the five criteria. Suppose we have a set of instances S_L at node N_L. Assume there are n classes associated with the possibilities of occurrences $\wp_1, \wp_2, \ldots, \wp_n$. Concerning about the measure of how much *choice* is involved in the selection of the instance in S_L or of how uncertain we are of the outcome, we choose the entropy function to evaluate that.

Definition 4: The entropy for the set of instances S_L at node N_L is defined by

$$\text{Info}(S_L) = -\sum_{\forall c \in \mathcal{C}} \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \times \log_2 \frac{\mathcal{P}_L^c}{\mathcal{P}_L}.$$

where

$$\mathcal{P}_L = \sum_{x \in S_L} P_L(\mathbf{x})$$

is the sum of the possibility value $P_L(\mathbf{x})$ of all instances at node N_L, and

$$\mathcal{P}_L^c = \sum_{\mathbf{x} \in S_L \wedge \text{Class}(\mathbf{x}) = c} P_L(\mathbf{x})$$

is the sum over instances belonging to class c.

The entropy of a set measures the average amount of information needed to identify the class of an instance in the set. It is minimized when the set of instances are homogeneous, and maximized when the set is perfectly balanced among the classes.

A similar measurement can be defined when the set is distributed into b_L subsets, one for each branch based on the test at node N_L. The expected information requirement is the weighted sum over the subsets.

$$\text{Info}_T(S_L) = \sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \times \text{Info}(S_{L.i}).$$

To asses the "benefits" of a test, we need to consider the increase in entropy. The quality

$$\text{Gain}(\text{Test}_L) = \text{Info}(S_L) - \text{Info}_T(S_L).$$

measures the information gain due to the test Test_L. This gain criterion is used as the basis for attribute selection.

A. Choosing the Fuzzy Operations

Five criteria of fuzzy entropy limitate the fuzzy operators that can be used to calculate the possibility of each instance at a node. Here, the *fuzzy t-norm* operator is involved for the possibility evaluation because it can satisfy those criteria, especially, the fifth property.

Since the function, $\log_2$ is a continuous function, the fuzzy entropy defined by $\log_2$ is also a continuous function. It is easy to see that Info satisfies Property 1.

If S_L is the set of instances in N_L that has been purely classified into one class, that is all the $\wp_i$ of each instance but one are zero. Let $\wp_i \neq 0$ for some class C_i, then the possibility

$$\mathcal{P}_L = \sum_{x \in S_L} P_L(\mathbf{x}) = \sum_{x \in S_L} \wp_i(\mathbf{x}).$$

The possibilities $\mathcal{P}_L^c$ of the other classes are zero. Because

$$\mathcal{P}_L^c = \sum_{\mathbf{x} \in S_L \wedge \text{Class}(\mathbf{x}) = c} P_L(\mathbf{x}) = 0$$

for $c \neq C_i$. The entropy value of $\text{Info}(S_L)$ will be zero when all the possibilities $\wp_i$ but one are zero.

Property 3 restricts that the entropy value is maximum when all the class possibilities are equal. According to that, it needs that $\sum_c \mathcal{P}_L^c$ should be no bigger than $\mathcal{P}_L$. Otherwise, this property will not be satisfied. Let $|\mathcal{C}|$ be the number of classes and $\mathcal{P}_L^{C_i} = \mathcal{P}_L^{C_j}$ for $i \neq j$. In the FCT algorithm, the sum operation $\sum$ is defined to be equal to the sum operation in classical (crisp) set.

$$\begin{aligned} \text{Info}(S_L) &= -\sum_{\forall c \in \mathcal{C}} \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \times \log_2 \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \\ &\leq -\sum_{i=1}^{|\mathcal{C}|} \frac{\mathcal{P}_L}{|\mathcal{C}|\mathcal{P}_L} \log_2 \frac{\mathcal{P}_L}{|\mathcal{C}|\mathcal{P}_L} \\ &= -\sum_{i=1}^{|\mathcal{C}|} \frac{1}{|\mathcal{C}|} \log_2 \frac{1}{|\mathcal{C}|}. \end{aligned}$$

Since $0 \leq \mathcal{P}_L^c \leq \mathcal{P}_L$ for all class $c \in \mathcal{C}$, $\log_2 \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \leq 0$ and $\text{Info}(S_L) \geq 0$. Therefore, it is no doubt that the fourth property is also satisfied.

The purpose of attribute selection in FCTs is toward reducing the uncertainties in the data. After the fuzzy classification tree has been further generating, the total entropy of the child nodes should be no greater than the entropy of their parent nodes. In the other word, the total entropy of child nodes from a node should be less than or equal to the entropy of that node before the tree expanded. That is,

$$\text{Info}(S_L) \geq \sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \times \text{Info}(S_{L.i}).$$

This is what the fifth property gives, whic is a strong constraint that restricts the kinds of fuzzy operations and the membership functions. It also limits the clustering methods to generate the membership function from a node.

The membership function is the kernel for fuzzy classifications. To determine the membership function from a data set, the method of clustering is used. Clustering is a well-used method in pattern recognition. It plays a key role in searching for structures in data. There may be different kinds of models simultaneously occurring in the data, that is called *multi-model* [5]. Data could be clustered into differential groups in accordance to their distribution models. The models construct the membership function of the data.

Fuzzy c-means clustering method [2] , which satisfies the weaker requirement, is used to make a properly vague partition. The membership value of each datum defines how possible this

datum is associated with a category. The membership gives a meaningful explanation on this vagueness. Therefore, to deal with the unavoidable observation and measurement uncertainties, fuzzy clustering is a very suitable choice applied to real world applications.

Theorem 1: Let $\otimes$ be the *fuzzy t-norm* operator. If $\sum_{i=1}^{b_L} \mu_L(\mathbf{x}) \leq 1$ for every $\mathbf{x} \in S_L$. Definition 3 satisfies the fifth property of entropy. That is

$$\text{Info}(S_L) \geq \sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \text{Info}(S_{L.i})$$

`Proof` Let α be the maximal membership value for all membership functions. Since $\sum_l \mu_l(\mathbf{x}) \leq 1$ and $\alpha \geq \mu_l(\mathbf{x}), \forall l, \mathbf{x}$ and $\sum_{c\in\mathcal{C}} \mathcal{P}_L^c \leq \mathcal{P}_L$, the right-hand-side of the inequality is derived as follows.

$$\begin{aligned}
\sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \text{Info}(S_{L.i}) &= -\sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \sum_{c\in\mathcal{C}} \frac{\mathcal{P}_{L.i}^c}{\mathcal{P}_{L.}} \log_2 \frac{\mathcal{P}_{L.i}^c}{\mathcal{P}_{L.i}} \\
&= -\sum_{i=1}^{b_L} \frac{\mathcal{P}_{L.i}}{\mathcal{P}_L} \sum_{c\in\mathcal{C}} \frac{\mathcal{P}_L^c \otimes \mu_{L.i}}{\mathcal{P}_L \otimes \mu_{L.i}} \log_2 \frac{\mathcal{P}_L^c \otimes \mu_{L.i}}{\mathcal{P}_L \otimes \mu_{L.i}} \\
&\leq -\sum_{c\in\mathcal{C}} \frac{\mathcal{P}_L^c \otimes \alpha}{\mathcal{P}_L \otimes \alpha} \log_2 \frac{\mathcal{P}_L^c \otimes \alpha}{\mathcal{P}_L \otimes \alpha} \\
&\leq -\sum_{c\in\mathcal{C}} \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \log_2 \frac{\mathcal{P}_L^c}{\mathcal{P}_L} \\
&= \text{Info}(S_L).
\end{aligned}$$

IV. Algorithms

This section presents the learning algorithm for constructing a fuzzy classification tree from a set of training instances containing real-valued attributes. Previous approaches to this problem usually fuzzify the data before they are used to construct a decision tree [40]. The linguistic variables have to be defined ahead of time based on existing domain knowledge.

The main algorithm for FCT construction as shown in Figure 2 takes an input a set S_0 of instances, and starts by creating a root node N_1, adding its label to $\mathcal{L}$, and initializing S_1 to be S_0.

Algorithm *Build_FCT*
[Input] A set of training instances S_0
[Output] An FCT
1. $L \leftarrow 1$
   /* Initialize L to be 1 which is the label at the root node. */
2. $\mathcal{L} \leftarrow \{1\}$
   /* Let $\mathcal{L}$ be the set of labels represented the nodes that have not been expanded. */
3. $S_1 \longleftarrow S_0$
   /* S_1 at the root node is set to be the original set S_0. */
4. **loop** until $\mathcal{L} = \phi$
5. $L \leftarrow \text{random}(\mathcal{L})$
   /* Random select one of the label from $\mathcal{L}$. */
6. $\mathcal{L} \leftarrow \mathcal{L} \setminus \{L\}$
7. $\forall a_i,\ \tau_i \leftarrow \text{Spawn\_New\_Tree}(N_L, a_i)$
8. Find τ_k s.t. $\text{Info}(\tau_k) = \max_j \text{Info}(\tau_j)$
9. $\text{Gain} \leftarrow \text{Info}(\mathcal{T}_L) - \text{Info}(\tau_k)$
10. **if** $\text{Gain} > \epsilon$ **then**
    $\mathcal{L} \leftarrow \mathcal{L} \cup \text{leaf}(\tau_k)$
    Assign subsets of S_L into $S_{L.1}, \ldots, S_{L.k}$

Fig. 2. The algorithm to construct FCTs.

The fuzzy gain ratio evaluation is based on the algorithm in Figure 3.

Algorithm *Evaluate_Entropy*
[Input] An FCT with root node N_L
[Output] The entropy value of $\mathcal{T}_L$
1. $\forall l \in \mathcal{L}$, s.t. N_l is any node in $\mathcal{T}_L$,
   $\text{Info}(S_l) \leftarrow -1$ /* Initialization */
   /* $\text{Info}(S_l)$ is nonnegative, and therefore set a negative value to it first. */
2. $\forall l \in \mathcal{L}$, s.t. N_l is a leaf node,
   $\text{Info}(S_l) \leftarrow -\sum_{c\in\mathcal{C}} \frac{\mathcal{P}_l^c}{\mathcal{P}_l} \times \ln \frac{\mathcal{P}_l^c}{\mathcal{P}_l}$
3. **loop** until $\text{Info}(S_L) \geq 0$
   if $\forall i, 1 \leq i \leq b_l\ \text{Info}(S_{l.i}) \geq 0$ **then**
   $\text{Info}(S_l) \leftarrow \sum_{i=1}^{b_l} \frac{\mathcal{P}_{l.i}}{\mathcal{P}_l} \times \text{Info}(S_{l.i})$
   end
4. **return** $\text{Info}(S_L)$.

Fig. 3. The gain ratio evaluation algorithm.

The procedure $\text{Spawn\_New\_Tree}(N_L, a_i)$ that expands the tree from node N_L according to some attribute a_i is shown in Figure 4.

Algorithm *Spawn_New_Tree*
[Input] An unexpanded node N
An attribute a
[Output] An expanded tree rooted at node N

$\forall i,\ 1 \leq i \leq n$ do the following:
1. *Project* instances at node N of class C_i onto attribute a
2. *Smooth* the resulting histogram using k-median method
3. *Partition* the smoothed histogram into clusters
4. *Create* a new branch from N_L for each cluster
5. *Define* the membership function for each branch

Fig. 4. The algorithm to expand the fuzzy classification trees at each node.

V. Experiments

The dataset selected is from a general population who were admitted for two-day physical checkup at National Taiwan University Hospital (NTUH) since November 1, 1993 to October 31, 1994. All the subjects had no prior history of any colorectal pathology. During this one-year period, 2987 patients were admitted for physical checkup. A total of 2746 patients who underwent sigmoidoscopy enrolled for the polyp screening analysis. There are 264 patients (9.5%) found to have rectosigmoid polyps by 60 cm-flexible sigmoidoscopy. Since the national health insurance system did not cover the fee of physical checkup, most cases were considered from upper and middle socioeconomic classes.

The purpose of this study was to determine the prevalence of distal large bowel polyps, both adenomatous and hyperplastic. At NTUH, there are about 500 checkup records for each patient in a two-day physical checkup. Sigmoidoscopy using 60cm flexible endoscope without sedation was administered by experienced endoscopists on all patients except those who gave up this procedure. If polyps were detected, the endoscopists should describe the size, number and location in details. According to the endoscopic appearance, submucosal tumor, such as leiomyoma, lymphoid follicle, lipoma, and normal mucosa excrescences, was considered as negative findings for this study. Although biopsies might be done at the screening site, it was not mandatory to this study at this stage.

Twenty one attributes, such as blood type, sex, age, body mass index, serum cholesterol, triglyceride, total protein, albumin/globulin, albumin, Zinc Turbit Test, direct bilirubin, total bilirubin, alkaline phosphotatase, acid phosphotatase, alanine aminotransferase, asparate aminotransferase, mean corpuscle volume, hemoglobin, hemoglobin A1C, alcohol consumption, and smoking, are selected for discovering the knowledge about the patients who will get polyp.

A. Cross Validation Estimates

A three-fold cross validation for the polyp screening data set was performed. The original data set are randomly split into two parts. One (2/3) is for training, and the other (1/3) is used for testing. FCT and C4.5 methods have been compared across a variety of learning tasks in each experiment.

These results were obtained according to the F-test under the confident level of 95%. According to Table I, the error rate on

TABLE I
THE ERROR RATES OF THE NTUH CHECKUP DATA SET FOR POLYP SCREENING (1).

	Error Rate	
Method	*False Negative*	*False Positive*
FCT	0.226478 ± 0.087654	0.010175 ± 0.007056
C4.5	0.971804 ± 0.020626	0.007173 ± 0.001265

False Negative of C4.5 is 0.971804 ± 0.020626 which is higher than FCT's 0.226478 ± 0.087654. Since 1 minus the value of *False Negative* is the value for sensitivity, we can conclude that the sensitivity of C4.5 is 0.992827 ± 0.001265 and the sensitivity of FCT is 0.989825 ± 0.007056. It means that FCT is more adapted than C4.5 for polyp screening. About 78% patients who have polyps will get positive response without taking colonscope examinations. However, about 1% patients who do not have polyps will be detected to have polyps by FCT, that is less specific than C4.5. The detail is shown in the following section.

Five attributes, albumin/globulin, albumin, alanine aminotransferase, asparate aminotransferase, and mean corpuscle volume, are substituted by uric acid, Na^+, K^+, Ca^{++}, and Cl^-. After we performing 200 runs, the error rates of FCT and C4.5 is listed in Table II. Those substituted attributes are not important in the polyp screening dataset because they are seldom occurred in a fuzzy classification tree or a C4.5's decision tree, even as they occurred as the tests in the trees are far beneath the root of the tree. However, those five attributes, uric acid, Na^+, K^+, Ca^{++}, and Cl^- are less important than albumin/globulin, albumin, alanine aminotransferase, asparate aminotransferase, and mean corpuscle volume for polyp screening because of the increased error rates.

TABLE II
THE ERROR RATES OF THE NTUH CHECKUP DATA SET FOR POLYP SCREENING (2).

	Error Rate	
Method	*False Negative*	*False Positive*
FCT	0.251768 ± 0.092644	0.176667 ± 0.02075
C4.5	0.971804 ± 0.021626	0.007173 ± 0.001746

In most of the fuzzy classification trees for polyp screening, we found that age, body mass index, triglyceride, Zinc Turbit Test and hemoglobin A1C were at the important locations (root or near the root as possible) for constructing the classification trees. It seems that these five attributes are the key features for polyp screening. If we substituted uric acid, Na^+, K^+, Ca^{++}, and Cl^- for age, body mass index, triglyceride, Zinc Turbit Test and hemoglobin A1C, the error was increased. Table III lists the error rates. Comparing the error rates in Table III with Table II, we can come to the conclusion that some of those attributes, age, body mass index, triglyceride, Zinc Turbit Test and hemoglobin A1C are important for polyp screening.

TABLE III
THE ERROR RATES OF THE NTUH CHECKUP DATA SET FOR POLYP SCREENING (3).

	Error Rate	
Method	*False Negative*	*False Positive*
FCT	0.352234 ± 0.107644	0.182367 ± 0.120444
C4.5	0.986231 ± 0.010325	0.003242 ± 0.002241

From those empirical results on polyp screening, we find that FCT is more suitable than C4.5 for polyp screening. Not only FCT is able to make more precise decision for polyp screening, but also FCT is able to properly reflect the effects of features. C4.5 is not capable of doing them.

VI. Discussion

In some applications, the classifier is advantageous not to produce a classification on every instance. The classifier is needed to produce the reasonable classification to assist a person to perform the final decision. When there is incomplete or inadequate data, a system that make no prediction may provide more useful information than a system that makes its best guess on every case. In addition, for disease screening, the classifier should satisfy the following criteria.

- Due to the limitation of medical resources, the classifier needs to identify the patients who do not get the disease and do not need to take any further diagnosis.
- The classifier is able to distinguish the patients who should take a further diagnosis. That is, the classifier can identify the patients who are at the risk of getting the disease.

A useful data mining tool is not expected to substitute human being. The most important is that the tool can help people filter some impossible results. FCT gives each patient the possibility of being in each class.

TABLE IV
THE RATIO OF THE DIFFERENCE OF THE PREDICTED POSSIBILITIES OF TWO CLASSES THAT IS LESS THAN A THRESHOLD IN THE NTUH CHECKUP DATA SET FOR FCT POLYP SCREENING.

	Difference Between two Classes	
Criterion	*False Negative*	*False Positive*
≤ 0.15	0.0513002	0.001686
≤ 0.1	0.226478	0.010175

VII. Conclusion

The uncertainties and noise make classification difficult. Missing or imprecise information may prevent a case from being classified at all. It is occurred in the boundaries of the data in two more different classes [17], [30]. In the presence of uncertainties, it is often desirable to have an estimate of the degree that an instance is in each class.

Probabilistic tree classifiers [5], [6], [29], [35] have been proposed to deal with uncertainties and noise. However, the *a priori* probabilities are needed to explain the result of classifications. In addition, probabilistic tree classifiers do not give a good solution for data partition. For numerical attributes, discretization [15], [29] makes the data in the overlapped region be classified into only one branch. A test instance falls down a single branch to arrive at a leaf where a probability is associated with each class. Such classifications ignore the possibility that instance could be classified into the other branches. Therefore, several methods, including Buntine's classification trees [5], Rymon's *Set Enumeration* tree [33] have been proposed to address this issue. However, these approaches are inefficient in both time and space.

In a fuzzy classification tree, an instance has a membership value at each leaf node. Instead of determining a single class for any given instance, fuzzy classification trees can predict the degree of *possibility* for every class. Using information-based measures, there is no need to generate multiple classification trees. Therefore, it requires less time and space than decision forests [19].

C4.5 is totally useless for polyp screening. All the patients who have polyp are almost classified into the `healthy` class. Basically, a requirement for disease screening strategies is that few false negative results should be determined. Awfully, C4.5 always makes wrong decisions for the patients who have polyps. Only few instances can be clearly classified. The testing result of the checkup dataset is formally under the consideration of F-test at the confident level of 95%. Using the three-fold cross validation testing, we will see that the error rate on false negative of FCT is less than the error rates on false negative of C4.5. That is, FCT is more sensitive than C4.5. The decisions of C4.5 are always biased to the majority, if only a small proportion of population will get the disease. In medical and financial applications, it is important that a classifier should give the estimate degrees of all potential classes. The classifiers should avoid classifying an instance into only one class. The fuzzy classifier, fuzzy classification trees, can estimate the possible degrees of all classes. According to these possibilities, even if we pick the class with the high possibility to be the patient's class, a much better prediction can be made by FCT than *C*4.5.

References

[1] P. W. Baim. A method for attribute selection in inductive learning system. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(9):888–896, 1988.

[2] J. C. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, 1981.

[3] X. Boyen and L. Wehenkel. Automatic induction of fuzzy decision trees and its application to power system security assessment. *Fuzzy Sets and Systems*, 102:3–19, 1999.

[4] L. Breiman, J. Friedman, R. Olshen, and C. Stone. *Classification and Regression Trees*. Chapman and Hall, London, 1984.

[5] W. Buntine. Learning classification trees. *Statistics and Computing*, 2:63–73, 1992.

[6] R. G. Casey and G. Nagy. Decision tree design using a probabilistic model. *IEEE Transactions on Information Theory*, 30(1):93–99, 1984.

[7] I. Chiang and J. Hsu. Integration of fuzzy classifiers with decision trees. In *Proceedings of Asian Fuzzy Systems Symposium*, pages 65–78, Kenting, Taiwan, 1996.

[8] I. Chiang and J. Hsu. Fuzzy classification trees for data analysis. *Fuzzy Sets and Systems*, 2002.

[9] K. J. Cios and L. M. Sztandera. Continuous ID3 algorithm with fuzzy entropy measures. In *Proceedings of the International Conference on Fuzzy Systems*, pages 469–476, San Diego, CA, 1992.

[10] P. Clark and T. Niblett. The CN2 induction algorithm. *Machine Learning*, 3:261–283, 1989.

[11] R. Duda and P. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

[12] D. Heath, S. Kasif, and S. Salzberg. Learning oblique decision trees. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence*, pages 1002–1007, Chambery, France, 1993.

[13] J. Y. Hsu and I. Chiang. Fuzzy classification trees. In *Proceedings of the Ninth International Symposium on Artificial Intelligence*, pages 431–438, Cancun, Mexico, 1996.

[14] C. Z. Janickow. Fuzzy decision trees: Issues and methods. *IEEE Trans. on System, Man, and Cybernetics B: Cybernetics*, 28(1):1–14, 1998.

[15] R. Kerber. ChiMerge: Discretization of numeric attributes. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pages 123–128, San Jose, CA, 1992.

[16] A. De Luca and S. Termini. A definition of a nonprobabilistic entropy in the setting of fuzzy sets theory. *Information and Control*, 20:301–312, 1976.

[17] R. S. Michalski. Learning flexible concepts: Fundamental ideas and method based on two-tiered representation. In Y. Kodratoff and R. S. Michalski, editors, *Machine Learning: An Artificial Intelligence Approach*, volume III. Morgan Kaufmann, Los Altos, CA, 1990.

[18] J. Mingers. An empirical comparison of selection measures for decision-tree induction. *Machine Learning*, 3:319–342, 1989.

[19] P. M. Murphy and M. J. Pazzani. Exploring the decision forest: An empirical investigation of Occam's razor in decision tree induction. *Journal of Artifical Intelligence Research*, 1:257–275, 1994.

[20] S. K. Murthy. *On Growing Better Decision Trees from Data*. PhD dissertation, The Johns Hopkins University, Baltimore, Maryland, 1995.

[21] S. K. Murthy, S. Kasif, and S. Salzberg. A system for induction of oblique decision trees. *Journal of Artificial Intelligence Research*, 2:1–32, 1994.

[22] S. K. Murthy, S. Kasif, S. Salzberg, and R. Beigel. OC1: Randomized induction of oblique decision trees. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 322–327, Washington, DC, 1993.

[23] M. Pazzani and D. Kibler. The utility of knowledge in inductive learning. *Machine Learning*, 9(1):57–94, 1991.

[24] W. Pedrycz and Z. A. Sosnowski. The design of decision trees in framework of granular data and their application to software quality models. *Fuzzy Sets and Systems*, 123:271–290, 2001.

[25] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1:81–106, 1986.

[26] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[27] J. R. Quinlan. *C*4.5*: Programs for Machine Learning*. Morgan Kaufmann, Los Altos, CA, 1993.

[28] L. Rendell and H. Cho. Empirical learning as a function of concept character. *Machine Learning*, 5(3):267–298, 1990.

[29] R. Rivest. Learning decision lists. *Machine Learning*, 2:229–246, 1987.

[30] D. Rumelhart, G. Hinton, and R. Williams. Learning internal representations by error prpagation. In D. Rumelhart and J. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations*, volume 1, pages 318–362. MIT Press, Cambridge, MA, 1986.

[31] R. Rymon. An SE-tree based characterization of the induction problem. In *Proceedings of the Tenth International Conference on Machine Learning*, pages 268–275, Amherst, MA, 1993.

[32] J. Sauar, G. Hoff, and T. Hausken. Colonoscopic screening examination of relatives of patients with colorectal cancer. *Scandinavian Journal of Gastroenterology*, 27:667–672, 1992.

[33] J. Schuermann and W. Doster. A decision theoretic approach to hierarchical classifier design. *Pattern Recognition*, 17(3):359–369, 1984.

[34] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:379–423, 623–656, 1948.

[35] M. Shieh, I. Chiang, J. Wong, C. Huang, S. Huang, and C. Wang. Prevalence of colorectal polyps in Taiwan: 60cm-sigmoidoscopic findings. *Biomedical Engineering-Application, Basis, Communication*, 7(3):50–55, 1995.

[36] E. H. Shortliffe. Computer programs to support clinical decision making. *Journal of the American Medical Association*, 258:61–66, 1987.

[37] A. Suárez and J. F. Lutsko. Globally optimal fuzzy decision trees for classification and regression. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(12):1297–1311, 1999.

[38] Y. Yuan and M. J. Shaw. Induction of fuzzy decision trees. *Fuzzy Sets and Systems*, 69:125–139, 1995.

Textual Document Indexing and Retrieval via Knowledge Sources and Data Mining

Wesley. W. Chu, Zhenyu Liu and Wenlei Mao
Computer Science Department, University of California, Los Angeles 90095
{wwc, vicliu, wenlei}@cs.ucla.edu

Abstract We present a knowledge-based query expansion technique to improve document retrieval effectiveness. The general concept terms in a query are substituted by a set of specific concept terms used in the corpus that co-occur with the key query concept. Since the expanded query matches with the document index terms much better, experimental results reveal that such query expansion produces better retrieval effectiveness than the unexpanded ones.

We have also developed a new phrase-based indexing technique that combines concepts with word stems. Using word stems can compensate for the incompleteness of the knowledge sources. Experimental results reveal that using phrase-based for indexing produces more accurate document retrieval than using word stems or concepts alone.

We also present an implementation that integrates the query expansion and phrase-based indexing in a medical digital library for the retrieval of patient records, laboratory reports and medical literature.

1. Introduction

Efficient document retrieval based on user query is achieved by indexing. The current technique uses word stem to index a document [1]. Such a technique suffers from the inability to match words in a query with their related words such as synonyms, hypernyms and hyponyms [2] in the documents. Therefore, there are recent attempts to index the document based on conceptual terms. However the content in knowledge sources are usually incomplete. As a result, past research reveals that although using conceptual terms for document indexing can solve some of the problems, it cannot outperform the word-stem-based model [3,4,5,6]. To remedy the incompleteness of the knowledge sources, we propose a phrase-based indexing model where we parse a document into phrases based on the conceptual terms in domain specific knowledge sources, and calculate the similarity between two documents using both the similarity between the concepts and the common word stems in them. Including word stem in addition to concepts in document similarity evaluation compensates for the shortcoming of concept terms caused by the incompleteness of the knowledge sources.

When seeking specific information regarding a particular topic, the user often has to pose a general query with concept terms. For example, not knowing that "contact lenses" is a treatment option for keratoconus, the user has to request for "treatment options for keratoconus" in the query. This results in low retrieval precision since documents are indexed by the specific terms. To remedy such shortcoming, we propose to substitute the general concept terms in the query with the specific terms. The level of relevancy of a specific term in the resulting query is determined by its co-occurrence with the general concept term, which can be mined from the corpus. Based on the query, the knowledge sources can identify the irrelevant conceptual terms and prevent them from inserting into the query. Since the expanded queries match better with relevant documents, the document retrieval performance is improved.

We shall first present the phrase based index technique and the experimental results to show the performance improvements of phrased based index over word stem and concept based index methods. Next we shall present the knowledge based query extension technique and present the performance improvement derived from the query extension. Finally, we present an implementation of integrating the two proposed techniques in a medical digital library for retrieving medical textual records and reports.

2. Phrase-based indexing

To facilitate discussion, we shall use the following sample query in this section: "22 year old with hyperthermia, leukocytosis, increased intracranial pressure, and central herniation. Cerebral edema secondary to infection, diagnosis and treatment." The first part of the query is a brief description of the patient; and the second part is the information need.

2.1 Word stem Indexing

A document is commonly represented as a vector of terms in a *vector space model* (VSM) [1]. The basis of the vector space corresponds to distinct terms in a document collection. Components of the document vector are the weights of the corresponding terms that represent their relative importance in the document. In a naïvest approach, we could treat a word as a term. Yet, morphological variants like "edema" and "edemas" are so closely related that they are usually conflated into a single *word stem*, e.g., "edem," by stemming [1,7]. Our sample query thus consists of word stems "hypertherm," "leukocytos," "increas," "intracran," "pressur," etc. Word stems are usually treated as notational, rather than conceptual entities. Two word stems are considered unrelated if they are different. For example, the stem of "hyperthermia" and that of "fever" are usually considered unrelated despite their apparent relationship. In stem-based VSM, word stems constitute the basis of the vector space. The base vectors are orthogonal to each other because different word stems are considered unrelated. The weight $w^s_{a,u}$ of a word stem u in a document a is determined by the number of times u appears in a (known as the *term frequency*) and the number of documents that contain u (known as the *document frequency*) following the TF-IDF (term frequency, inverse document frequency) scheme [1]. In essence, the more often u appears in a, the more important u is in a. On the other hand, the more documents u belongs to, the less disambiguating power it has, and thus the less important it is.

Word stems are widely used as index terms. To improve retrieval accuracy, it is natural to replace word stems with concepts [3,4,5,6,8] or multiple-word combinations [9,10]. However, previous research showed not only no improvements, but degradation in retrieval accuracy when concepts were used in document retrieval [3,4,5,6] except when documents were very short [8]. When properly used, multiple-word combinations were shown to improve retrieval effectiveness for some special queries [9,10]. However, the retrieval effectiveness improvement for ad hoc queries is still questionable.

2.2 Concept-based VSM

Using word stems to represent documents results in the inappropriate fragmentation of concepts such as "increased intracranial pressure" into its component stems "increas," "intracran," and "pressur." Clearly, using *concepts* instead of single words or word stems as the vector space basis should produce a VSM that better mimics the human thought processes, and therefore should result in more accurate retrieval.

However, using concepts is more complex than using word stems. First, concepts are usually represented by multi-word phrases such as "increased intracranial pressure." More importantly, there exist synonymous and polysemous phrases. Two phrases sharing a concept are *synonymous*, and phrases that could represent more than one concept are *polysemous* [2]. For example, "hyperthermia" and "fever" are synonymous because they share the same concept "an abnormal elevation of the body temperature." At the same time, "hyperthermia" is polysemous, because in addition to the above meaning it also means "a treatment in which body tissues is exposed to high temperature to damage and kill cancer cells." Synonyms can be identified with the help of a dictionary or a thesaurus. Determining which concept a particular polysemous phrase represents is known as *word sense disambiguation* (WSD) [11]. Third, some concepts are related to one another. Hypernym and hyponym relations are important conceptual relations. If we say "an x is a (kind of) y" then concept x is a hyponym of concept y, and y is a hypernym of x [2]. "Hyperthermia" is a hyponym of "high body temperature;" and "high body temperature" is a hypernym of "hyperthermia."

Concept identifiers are usually used to identify concepts. Using UMLS [13] as a knowledge source, our sample query becomes (15967, 203597), (23518), and (151740) etc., representing "hyperthermia," "leukocytosis," and "increased intracranial pressure," etc., respectively.

In concept-based VSM, the basis of the vectors space consists of distinct concepts. To model the relationship of such concepts as "hyperthermia" and "elevated body temperature" we remove the orthogonality constraint on base vectors. Base vectors for two related concepts form an acute angle. Only when we cannot find any reasonable relations between two concepts that we treat their corresponding vectors as orthogonal. The cosine of the angle between two concept vectors is defined as the *conceptual similarity* between the corresponding concepts. The conceptual similarity thus ranges from 0 to 1 with 0 indicating unrelated and 1 indicating highly related concepts.

To study the effects of conceptual similarities, we shall compare two cases. In one case, we assume all different concepts are unrelated. Therefore, all base vectors of the vector spaces are orthogonal to one another. We label this case as "O" for orthogonal. In the other case, we derive conceptual similarities from knowledge sources. The resulting base vectors are no longer mutually orthogonal. We label this case as "NO" for non-orthogonal.

We derive the weight w^c_{a,x_i} of the i^{th} concept x_i in a document a using a slightly modified version of TF-IDF scheme. Higher weights are assigned to longer phrases that correspond to more specific concepts. For example, if the term frequencies and document frequencies for

"increased intracranial pressure" and "hyperthermia" were identical, the former concept would obtain a higher weight than the latter.

2.2 Phrase-based VSM

Conceptual similarities needed in concept-based VSM are derived from knowledge sources. The quality of such VSM therefore depends heavily on the quality of the knowledge sources. The absence of certain conceptual relations in the knowledge sources could potentially degrades retrieval accuracy. For example, treating "cerebral edema" and "cerebral lesion" as unrelated is potentially harmful. Noticing their common component word of "cerebral" in the above phrases, we propose phrase-based VSM to remedy the incompleteness of the knowledge sources.

In phrase-based VSM, a document is represented as a set of phrases. Each phrase may correspond to multiple concepts (due to polysemy) and consist of several word stems. Our sample query now becomes [(15967, 203597), ("hypertherm")], [(23518), ("leukocytos")] and [(151740), ("increas", "intracran", "pressur")] etc.

Following the TF-IDF schemes in stem-based and concept-based VSMs, we can derive the stem weight $w^s_{a,u_{i,k}}$ of the k^{th} stem $u_{i,k}$ and the concept weight $w^c_{a,x_{i,m}}$ of the m^{th} concept $x_{i,m}$ in phrase i of a.

Similar to concept-based VSM, we study two cases, O and NO. In case O, different concepts are unrelated; while in case NO, concepts may be related. In both cases, distinct word stems are assumed to be unrelated.

2.3 Document Similarity

The similarity of two documents a and b is the cosine of the angle between their corresponding document vectors $\vec{a}$ and $\vec{b}$; that is,

$$sim(a,b) = \cos(\vec{a},\vec{b}) = \frac{\vec{a} \bullet \vec{b}}{\sqrt{\vec{a} \bullet \vec{a}}\sqrt{\vec{b} \bullet \vec{b}}} \qquad \textbf{(1)}$$

We shall extend the vector dot product $\vec{a} \bullet \vec{b}$ and denote the *extended dot product* (EDP) as $\vec{a} \circ \vec{b}$ to represent the cases when the components of the vectors $\vec{a}$ and $\vec{b}$ are related. Using the EDP in place of the dot product, we derive document similarity as,

$$sim(a,b) = \frac{\vec{a} \circ \vec{b}}{\sqrt{\vec{a} \circ \vec{a}}\sqrt{\vec{b} \circ \vec{b}}} \qquad \textbf{(2)}$$

EDP Derivation

To derive the EDP in the phrase-based VSM, we first consider concepts without polysemy. Thus,

$$\vec{a} \circ \vec{b} = \sum_{i,j} S^c_{i,j} \qquad \textbf{(3)}$$

where $S^c_{i,j}$ is the conceptual contribution of phrase i in a and phrase j in b to the EDP. Assuming that each phrase represents a single concept, we have,

$$S^c_{i,j} = w^c_{a,x_i} w^c_{b,y_j} s(x_i, y_j) \qquad \textbf{(4)}$$

where $s(x_i, y_j)$ is the conceptual similarity between the i^{th} concept x_i in a and the j^{th} concept y_j in b. In the orthogonal case, $s(x,y)$ is reduced to the Kronecker delta function,

$$d(x,y) = \begin{cases} 1 & x = y \\ 0 & x \neq y \end{cases}$$

In the non-orthogonal case, conceptual similarities are derived from knowledge sources.

Conceptual Similarity, S(x,y)

Given a hypernym hierarchy, the conceptual similarity $s(x, y)$ between two concepts x and y depends on both the distance between them in the hierarchy and their generality. When two concepts are farther apart in the hypernym hierarchy, they are less similar – a concept is less similarity to its grandparent than to its parent in the hypernym hierarchy. Thus we define the conceptual similarity to be inversely proportional to the number of "hops" between x and y, $d(x,y)$. The generality of a concept x can be derived from the number of all its descendants $D(x)$. The more descendants x has, the more general it is. A general concept like "disease" has much more descendants than a more specific concept like "hyperthermia" has. Because of the exponential growth of the number of descendants when a concept moves up a tree structure, we take the logarithm of the number of descendant in conceptual similarity calculation. The conceptual similarity is therefore defined to be inversely proportional to the logarithm of the number of descendants of the two. A final consideration is the boundary case when we reach the leaves of the hypernym tree. Let us assume we have two concepts x_o and y_o, where x_o is the only direct hypernym of y_o, y_o is the only hyponym of x_o, and y_o has no hyponym of its own. Concepts x_o and y_o are so much alike that we define the conceptual similarity between them to be c close to 1, say 0.9, to represent such closeness. As a result, the conceptual similarity between concepts x and y is,

$$s(x,y) = \frac{c}{d(x,y)\log_2(1 + D(x) + D(y))} \qquad \textbf{(5)}$$

In order to use (4) in the presence of polysemy, we need to disambiguate senses. To avoid WSD cost, we use the most popular concept that a phrase represents as the meaning of the phrase. Alternatively, we derive the

conceptual contribution to the similarity between two phrases using an aggregation of (4) over all possible concept pairs, where each pair consists of one concept from each phrase.

The contribution of word stems to the EDP is the sum of the weight product for those word stems common to both phrases,

$$S^s_{i,j} = \sum_{k,l} w^s_{d,u_{i,k}} w^s_{q,v_{j,l}} d(u_{i,k}, v_{j,l}) \qquad \textbf{(6)}$$

where $u_{i,k}$ and $v_{j,l}$ are the k^{th} word stem in phrase i in a and l^{th} word stem in phrase j in b respectively.

Given the contribution of concepts and stems, (4) and (6), we select the larger of the two as the contribution of phrase i in a and phrase j in b to the EDP and get,

$$\vec{a} \circ \vec{b} = \sum_{i,j} \max(S^c_{i,j}, S^s_{i,j}) \qquad \textbf{(7)}$$

Such selection remedies the incompleteness of the knowledge sources. $\vec{a} \circ \vec{a}$ and $\vec{b} \circ \vec{b}$ can be derived similar to (7). The document similarity can then be computed from (2) using these EDPs.

2.4. Experimental Results

The Test Collection, OHSUMED

OHSUMED [12] is a large test collection used in many information retrieval system evaluations. The test set consists of a reference collection, a query collection, and a set of relevance judgments.

The reference collection is a subset of the MEDLINE database. Each reference contains a title, an optional abstract, a set of MeSH headings, author information, publication type, source, a MEDLINE identifier, and a sequence identifier. The query collection consists of 106 queries. Each query contains a patient description, an information request, and a sequence identifier. The sample query we use in this paper is query 57 in the collection. 14,430 references out of the 348K are judged by human experts to be not relevant, possibly relevant, or definitely relevant to each query.We use the title, the abstract, and the MeSH headings to represent each document; and the patient description, and the information request to represent each query.

The Knowledge Source

UMLS [13] is a medical lexical knowledge source and a set of associated lexical programs. The knowledge source consists of UMLS Metathesaurus, SPECIALIST lexicon, and UMLS semantic network. Especially of interest to us is its central vocabulary component – the Metathesaurus. It contains biomedical phrases from more than 60 vocabularies and classifications. The Metathesaurus contains 1.6M phrases representing over 800K concepts.

A concept unique identifier (CUI) identifies each concept. UMLS tends to assign a smaller CUI to a more popular sense of a phrase. Therefore, we use the concept with the smallest CUI in conceptual contribution calculation (2). Our experiment results show that such heuristic produces retrieval accuracy comparable to that produced by the aggregation approach where we consider all conceptual similarities due to different sense combinations from the phrases.

The Metathesaurus encodes many conceptual relations. We concentrate on hypernym relations. Two relations in UMLS roughly correspond to the hypernym relations: the RB (border than) and the PAR (parent) relations. For example, "hyperthermia" has a parent concept "body temperature change." We combine the 838K RB and 607K PAR relations into a single hypernym hierarchy.

Hypernymy is transitive [14]. For example, "sign and symptom" is a hypernym of "body temperature change" and "body temperature change" is a hypernym of "hyperthermia," so "sign and symptom" is also a hypernym of "hyperthermia." However UMLS Metathesaurus encodes only the direct hypernym relations but not the transitive closure. We derive the transitive closure of the hypernym relation and use (5) to calculate the conceptual similarities.

Phrase Detection

Given a set of documents (106 queries and 14K judged documents of OSHUMED), we need to detect any occurrences in a set of phrases (1.3M phrases in UMLS). We adopt the Aho-Corasick algorithm [15] for the set-matching problem to detect phrases:

First, Aho-Corasick algorithm detects *all* occurrences of any phrase in a document. But we only keep the longest, most specific phrase. For example, although both "edema" and "cerebral edema" are detected in the sample query, we keep only the latter and ignore the former.

Second, to detect multi-word phrases, we match stems instead of words in a document with UMLS phrases. We use Lovins stemmer [7] to derive word stems. To avoid conflating different abbreviations into a single stem, we define the stem for a word shorter than four characters to be the original word.

Third, stop-word removal is performed *after* the multi-word phrase detection. In this way, we correctly detect "secondary to" and "carcinoma" from "cerebral edema secondary to carcinoma." We would incorrectly detect "secondary carcinoma" if the stop-words ("to" in this case) were removed before the phrase detection.

Discussion of Results

To calculate retrieval accuracy using precision-recall [1], we combined the "possibly relevant" and "definitely relevant" judgments in OHSUMED into a single relevant category. Based on the type of VSM, we calculate the document similarity between each of the 14K documents and each of the 105 queries (one query does not have relevant document). For a given VSM

and a query, we rank the documents from the most to the least similar to the query. When a certain number of documents are retrieved, *precision* is the percentage of retrieved documents that are relevant; and *recall* is the percentage of the relevant documents that has been retrieved so far. We evaluate the retrieval accuracy by interpolating the precision values at eleven recall points. The overall effectiveness of different VSM is then compared by averaging over the performance of all the 105 queries (Figure 1). The average of the eleven precision values gives an overview of the effectiveness of each VSM.

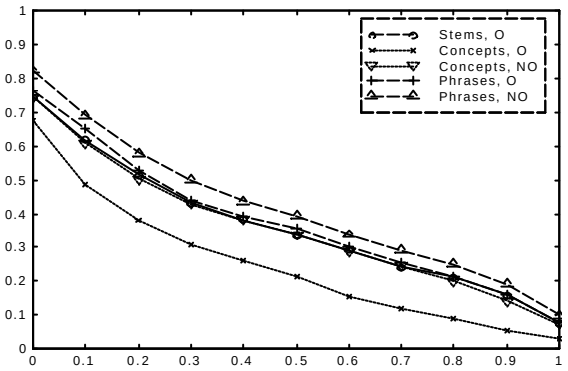


Figure 1 Comparison of the average precision-recall over 105 queries.

1. The baseline labeled (Stems, O) uses stem-based VSM. Its 11-point average precision is 0.363.
2. Considering the contribution of concepts only, and treating different concepts as unrelated (Concepts, O), we arrive at an 11-point average precision of 0.260, which is a 28% decrease from the baseline.
3. Similar to 2, but taking the concept inter-relationship into consideration (Concepts, NO), we achieve a significant improvement over 2. The average accuracy is similar to that of the baseline.
4. Considering contribution of both concepts and word stems in a phrase, but treating different concepts as unrelated (Phrases, O), we arrive at an 11-point average precision of 0.375, a 3% improvement over the baseline.
5. Similar to 4, but taking concept interrelations into consideration (Phrases, NO), we achieve an 11-point average precision of 0.420, which is a significant 16% improvement over the baseline.

Our experiment results reveal that viewing documents as concepts only and treating different concepts as unrelated can cause the retrieval accuracy to deteriorate (case 2). Considering concept inter-relations (case 3) or relating different phrases by their shared word stems (case 4) can both improve retrieval accuracy. The extended dot product combines contributions from the concepts and word stems. The phrase-based VSM utilizes such extended dot product and yields significant improvement in retrieval accuracy.

3. Enhance retrieval performance via query expansion

When posing a query, a user usually has a main objective (*key concept*, c_{key}) in mind and uses additional *general supporting conceptual* terms, c_s, to specify certain aspects of c_{key}. For example, when a user asks "Keratoconus, treatment options.", "Keratoconus" (an eye disease) is the key concept whereas "treatment options" is a general supporting concept. Although such query is easy to form, it does not match well with relevant documents that use such specific supporting concepts as "contact lens" or "keratoplasty".

To remedy this problem, we propose to substitute the general supporting concepts by specific concepts that used in the relevant documents. We need to select the set of specific concepts, and determine the weight of each of these concepts. We shall first present the weight determination method, and then compare two concept selection approaches.

For a specific concept c, its weight should represent the degree of correlation between c and the key concept term, c_{key}. For example, "contact lens" is a treatment option for "Keratoconus" but not "Back pain", and therefore it should assign a larger weight in the expansion of "Keratoconus, treatment options" than that of "Back pain, treatment options". We shall now present a scalable method for such weight assignment. We shall first represent concepts into inverted document vectors, and then use the similarity between the two inverted document vectors to represent the correlation between the two concepts.

Given a corpus of n documents, the inverted document vector for concept c_i, $\vec{q}_{ci}$, is defined as an n dimensional vector. The weight of component c_i in the vector represents the term frequency of concept c_i in each document. For example, if a corpus contains documents D_1, D_2 and D_3, and concept c occurs three, zero, and two times in these documents respectively, then $\vec{q}_{ci}$ =<3, 0, 2>.

We further define the correlation between concepts c_i and c_j as:

$$correl(c_i, c_j) = \cos(\vec{q}_{c_i}, \vec{q}_{c_j}) = \frac{\vec{q}_{c_i} \bullet \vec{q}_{c_j}}{\sqrt{\vec{q}_{c_i} \bullet \vec{q}_{c_i}} \sqrt{\vec{q}_{c_j} \bullet \vec{q}_{c_j}}}$$

The correlation between two concepts ranged from 0 to 1. For example, if $\vec{q}_{c_i}$ =<3, 0, 2>, $\vec{q}_{c_j}$ =<6, 0, 4>, and $\vec{q}_{c_k}$ =<0, 1, 0>, then *correl*(c_i, c_j)=1 and *correl*(c_i, c_k)=0. The correlation between all pairs of concepts can be

computed offline and stored in a concept correlation table (see Figure 3). For query expansion, the weight assigned to c_i is the correlation between a supporting concept c_i and c_{key}.

There are two concept selection approaches: with or without knowledge sources. When no knowledge source is available, all the supporting concept terms that have zero correlation with the key concept, c_{key}, can be viewed as irrelevant concepts and filtered out. Let c_1, c_2, ..., c_k be all the concepts that have nonzero correlation with c_{key}. The expanded query becomes $<(c_{key},1), (c_1, correl(c_1, c_{key})), \ldots, (c_k, correl(c_k, c_{key}))>$

In this expanded query vector, the higher the correlation value $correl(c_i, c_{key})$ is, the more emphasis is placed on the vector component of c_i. Clearly, the weight assigned to c_{key} is 1 since $correl(c_{key}, c_{key}) = 1$.

Since a general supporting concept in a query is usually only relevant to one or two aspects of c_{key}, the second approach uses knowledge sources together with the key concept and the general supporting concept in the query to select the relevant set of specific supporting concepts. For example, "Keratoconus, treatment options" is emphasizing on the treatment options for the disease, instead of diagnosis methods or causes for the disease. ULMS [13] indicates only three categories of medical concepts as potential treatments: "Therapeutic and Preventive Procedures", "Medical Devices" and "Pharmalogical Substance". Thus only concepts that belong to these three categories will be expanded into the query. By filtering out the irrelevant supporting concepts, the computation complexity is greatly reduced. In addition, the precision in the low recall region increases.

Considering the query example, "Keratoconus, treatment options", the top 10 added concepts for both approaches are listed in Tables 1 and 2, ranked by their correlation with c_{key}, "Keractoconus". Due to the application of a knowledge source of UMLS, concepts in the second table relate to the original query more tightly.

Ranking	Concept c_i	$correl(c_i, c_{key})$
1	Cornea	0.594
2	Cornea Transplantation	0.555
3	Contact lens	0.462
4	Penetrating keratoplasty	0.455
5	Epikeratoplasty	0.379
6	Visual Acuities	0.372
7	Myopia	0.320
8	Epikeratophakia	0.291
9	Eye	0.289
10	Combined corneal dystrophy	0.289

Table 1. Top 10 specific concepts added to the example query without the guidance of UMLS.

Ranking	Concept c_i	$correl(c_i, c_{key})$
1	Cornea Transplantation	0.555
2	Contact lens	0.462
3	Penetrating keratoplasty	0.455
4	Epikeratoplasty	0.379
5	Epikeratophakia	0.291
6	Eyeglasses	0.212
7	Buttons	0.193
8	Radial Keratotomy	0.171
9	Trephines	0.159
10	Thermokeratoplasty	0.151

Table 2. Top 10 specific concepts added to the example query with the guidance of UMLS.

To evaluate the performance improvement of query expansion, we select 28 OHSUMED queries [12] that contain general supporting concepts as the test set (see Table 3). Our experimental results for the query set reveal the average query expansion size without knowledge source is 1227 terms per query, while using ULMS, the average expansion size reduced to 82.3 terms per query. This represents more than an order of magnitude reduction in query expansion size.

OHUMED Query ID	Original Query Form
13	**LACTASE DEFICIENCY** *therapy options.*
38	**DIABETIC GASTROPARESIS,** *treatment.*
42	**KERATOCONUS,** *treatment options.*
47	**URINARY RETENTION,** *differential diagnosis.*

Table 3. Sample OHSUMED queries that contain general supporting concepts. Key concepts are shown in capital letters and general supporting concepts are in italics.

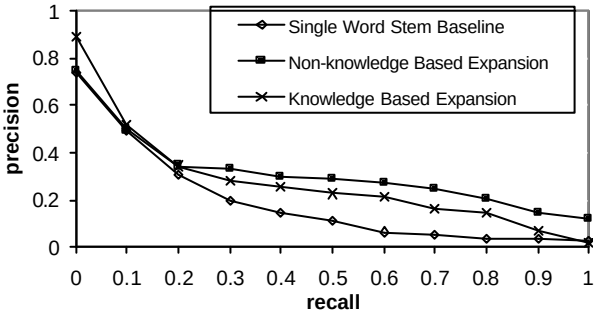


Figure 2. Retrieval performance improvements with query expansion

The retrieval performance improvement for the set of expaned OHSUMED queries is shown in Figure 2. We note that the expansion queries performed better than the non-expansion cases (base line). The query derived from knowledge-based expansion performs better than the cases without knowledge base in the low recall region, and the performance is reverse in the high recall region. This is because the non-knowledge based

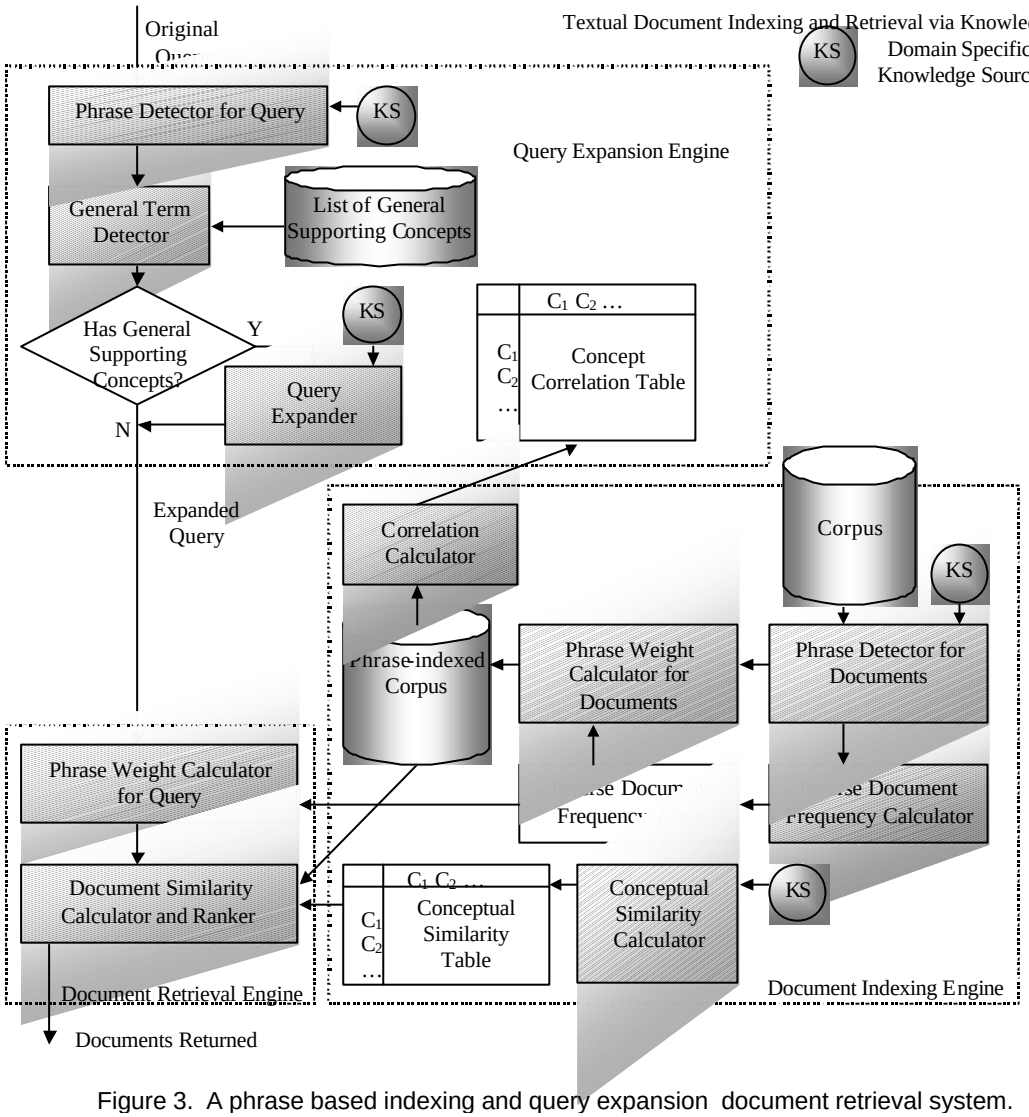


Figure 3. A phrase based indexing and query expansion document retrieval system.

expansion case includes many irreverent specific concepts, which resulted in low precision in the low recall region. Since relevant documents in the high recall region are not typically covered by the highly correlated supporting concepts in the query, adding more terms by the non-knowledge based method yields better precision in that region. Aside from the computation complexity saving of using the knowledge based expansion method, the choice of which expansion method to use depends on the application. When low recall region is more of concern, then knowledge-based method is preferable; and if high recall region is more important, then the non-knowledge based method should be considered.

4. Applications

We shall now present a document retrieval system that integrates the knowledge-based query expansion and phrase-based indexing for a digital medical library. As shown in Figure 3, the system consists of three subsystems: a Document Indexing Engine, a Query Expansion Engine, and a Document Retrieval Engine. The Document Indexing Engine processes the knowledge source and the corpus offline and generates data structure necessary for the online query expansion and document retrieval. When the system receives a user query, the Query Expansion Engine expands the query and the Document Retrieval Engine then returns a set of documents relevant to the query.

The medical knowledge source, UMLS, is used in the system for phrase detection, conceptual similarity derivation, and expansion term filtering. The Document Indexing Engine processes UMLS and the document corpus separately. The Conceptual Similarity Calculator derives the conceptual similarities between concepts from UMLS and stores them in the Conceptual Similarity Table. The Phrase Detector identifies concepts in UMLS from the documents. The Inverse Document Frequency Calculator uses the output of the phrase detector to construct an Inverse Document

Frequency Table. The Phrase Weight Calculator in turn calculates the weights of concepts and word stems in each phrase, and converts the original corpus into a Phrase-indexed Corpus. The Correlation Calculator then derives the conceptual correlations from the phrase-indexed corpus and stores them in the Concept Correlation Table.

When the system receives a user query, Phrase Detector first parses the query into phrases. The General Term Detector then determines if any query expansion is necessary by consulting the General Supporting Concept list. If no general supporting concepts are detected, the original query is directly input into the Document Retrieval Engine. Otherwise, the Query Expander replaces general concept terms with the specific ones in the appropriate categories as specified in UMLS. The weights of the expanded concepts are looked up from the Concept Correlation Table.

The Document Retrieval Engine compares the expanded query with the phrase-index and returns a set of documents to the user. First, the Similarity Calculator consults the Conceptual Similarity Table to calculate the phrase-based document similarities. The Ranker then returns to the user those documents the most similar to the query.

5. Conclusion

We introduced a knowledge-based technique to rewrite a user query containing general conceptual terms into one containing specific terms. These specific supporting terms are mined from the corpus, and are related to the general supporting concept and the query's key concept. Experimental results show that retrieval using such expanded queries is more effective than the original queries. Because the additional concept terms included in the expanded query are chosen from selected categories as indicated by the knowledge source, the average size of the expanded queries in knowledge-based approach is much smaller (reduced by more than an order of magnitude) than that produced by the full query expansion without any knowledge sources and also yield better retrieval performance in the low recall region which is of interest to most applications.

We developed a new vector space model that uses phrases to represent documents. Each phrase consists of multiple concepts and words. Similarity between two phrases is jointly determined by the conceptual similarity and their common word stems. Our experimental result reveals that the phrase-based VSM yields a 16% increase of retrieval effectiveness than the stem-based VSM. This improvement is because multi-word concepts are natural unit of information and using word stems in phrase-based document similarity compensates for the inaccuracy in conceptual similarities derived from incomplete knowledge sources.

We also presented an implementation that integrates the above techniques into a digital medical library at UCLA for the retrieval of patient records, laboratory reports and medical literatures.

References

[1] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*, 1983

[2] G.A. Miller, R. Beckwith, C. Fellbaum, D. Gross and K. Miller. Introduction to WordNet: an On-line Lexical Database. In *WordNet: an Electronic Lexical Database*, 1-19, 1998

[3] M. Mitra, C. Buckley, A. Singhal and C. Cardie. An Analysis of Statistical and Syntactic Phrases. In *Proc. RIAO97*, 200-214, 1997

[4] R. Richardson and A.F. Smeaton. Using WordNet in a Knowledge-based Approach to Information Retrieval. In *Proc. 17th BCS-IRSG*, 1995

[5] M. Sussna. Text Retrieval using Inference in Semantic Matanetworks. *PhD Thesis*, University of California, San Diego, 1997

[6] E.M. Voorhees. Using WordNet to Disambiguate Word Sense for Text Retrieval. In *Proc. 16th ACM-SIGIR.*, 171-180, 1993

[7] J.B. Lovins. Development of a Stemming Algorithm. In *Mechanical Translation and Computational Linguistics*, 11(1-2), 11-31, 1968

[8] A.F. Smeaton and I. Quigley. Experiments on using Semantic Distances Between Words in Image Caption Retrieval. In *19th Proc. ACM-SIGIR*, 174-180, 1996

[9] D. Johnson, W.W. Chu, J.D. Dionisio, R.K. Taira and H. Kangarloo. Creating and Indexing Teaching Files from Free-text Patient Reports. In *AMIA'99*, 1999

[10] J.A. Goldman, W.W. Chu, D.S. Parker and R.M. Goldman. Term Domain Distribution Analysis: A Data Mining Tool for Text Databases. In *2001 IMIA Yearbook of Medical Informatics*, 96-101, 2001

[11] N. Ide and J. Véronis. Word Sense Disambiguation: the State of the Art. In *Computational Linguistics*, 24(1), 1-40, 1998

[12] W. Hersh, C. Buckley, T.J. Leone and D. Hickam. OHSUMED: an Interactive Retrieval Evaluation and New Large Test Collection for Research. In *Proc. 22nd ACM-SIGIR Conf.*, 191-197, 1994

[13] National Library of Medicine. *UMLS Knowledge Sources, 12th edition*, 2001

[14] J. Lyons. *Semantics*, 1977

[15] A.V. Aho and M.J. Corasick. Efficient String Matching: an Aid to Bibliographic Search. In *CACM*, 18(6), 330-340, 1975

On Modal Decision Logics

Tuan-Fang Fan* Churn-Jung Liau† Y.Y. Yao‡

Abstract

Some modal decision logic languages are proposed for knowledge representation in data mining through the notions of models and satisfiability. The models are collections of data tables consisting of a finite set of objects described by a finite set of attributes. Some relationships may exist between data tables in a collection and the modalities of our languages are interpreted with respect to these relations in a Kripkean style semantics.

1 Introduction

Theory of knowledge has been a commonly important topic of many academic branches such as philosophy, psychology, economics, and artificial intelligence, whereas the storage and retrieval of data is the main concern of information science. In the modern experimental science, knowledge is usually acquired from observed data. The data can provide the cause-effect or associational relationship between attributes of the observed objects. However, when the amount of data is large, it becomes a difficult task to analyze the data and extract knowledge from them. With the aid of computers, the large amount of data stored in relational data tables can be transformed into symbolic knowledge automatically. Thus intelligent data analysis has received much attention recently.

While the data mining researches mainly concentrate on the design of efficient algorithms for extracting knowledge from data, to close the semantic gap between structured data and human-comprehensible concepts has been a lasting challenge for the research community[8]. This is called the interpretability problem of intelligent data analysis in [8]. Since the discovered knowledge is useful for a human user only when he can understand its meaning, the knowledge representation formalism will play an important role in the utilization of the induced rules. A good representation formalism should have clear semantics so that a rule can be effectively validated with respect to the given data tables. In this regard, logic is one of the best choices. As indicated by Zadeh[22], human usually compute with words instead of numbers, so if we can incorporate linguistically meaningful terms into the representation formalism, then the induced rules may be more useful to the human decision-makers.

The rough set theory proposed by Pawlak provides an effective tool for extracting knowledge from data tables[15]. To represent and reason about the extracted knowledge, a decision logic (DL) is also proposed in [15]. The semantics of the logic is defined in a Tarskian style through the notions of models and satisfaction. While DL can be considered as an instance of classical logic in the context of data tables, different generalizations of DL corresponding to some non-classical logics are also desirable from the knowledge representation viewpoint. For example, to deal with uncertain or incomplete information, some generalized decision logics have been proposed in [3, 13, 14, 19, 20].

These generalized decision logics, however, concentrate mostly on the representation of knowledge from a single data table. Though in principle, all data can be put into a single table, it is sometimes more natural to represent them by a collection of data tables. For example, in an enterprize database, the business transaction records may be stored as a collection of data tables indexed by dates. To extract knowledge from such structured data tables, we need richer representation languages than the decision logic. Among the traditional logical tools, modal logic would be one of the most appropriate candidates that can meet the need since it is a logic for reasoning about relations in a broad sense[1], whereas the knowledge extracted from multiple data tables usually concerns with the relationship of objects across different tables.

In this paper, we present a formulation of the modal decision logics based on multiple data tables. In the next section, we first review the decision logic proposed by Pawlak. Then, a general modal decision logic(MDL) is presented in section 3, which is followed by three case studies. They are respectively the uncertain, epistemic, and temporal decision logic. In each case, the syntax and semantics of the logics are presented and some illustrative examples are given. Fi-

*Department of Information Engineering, National Penghu Institute of Technology, Penghu, Taiwan. Email: dffan@npit.edu.tw

†Institute of Information Science, Academia Sinica, Taipei, Taiwan. Email: liaucj@iis.sinica.edu.tw. (The corresponding author)

‡Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2. Email: yyao@cs.uregina.ca

nally, the summery is given in the concluding section and some further research directions are also pointed out.

2 Review of Decision Logic

A data table(DT) is usually used as a regular approach for storage of data. A formal definition of data table is given in [15].

Definition 1 *A data table*[1] *is a triplet*

$$T = (U, A, \{a_T \mid a \in A\})$$

where

- U *is a nonempty finite set, called the universe,*
- A *is a nonempty finite set of primitive attributes, and*
- *for each* $a \in A$, $a_T : U \to V_a$ *is a total function, where* V_a *is the domain of values for* a*. Usually, we will simply write* a *instead of* a_T *for the functions.*

Given a data table T, we will denote its universe U and attribute set A by $Uni(T)$ and $Att(T)$ respectively.

In [15], a decision logic(DL) is proposed for the representation of the knowledge discovered from data tables. The logic is called decision logic because it is particularly useful in a special kind of data tables, called *decision table.* A decision table is a data table $T = (U, C \cup D, \{a_T \mid a \in C \cup D\})$, where $Att(T)$ can be partitioned into two sets C and D, called condition attributes and decision attributes respectively. By data analysis, decision rules relating the condition and the decision attributes can be derived from the table. A rule is then represented as an implication between formulas of the logic. However, for a general data table, the acronym DL can also denote *data logic*.

The basic alphabet of a DL consists of a finite set of attribute symbols $\mathcal{A}$ and for $a \in \mathcal{A}$, a finite set of value symbols $\mathcal{V}_a$. The syntax of DL is then defined as follows.

Definition 2

1. *An atomic formula of DL is a descriptor* (a, v)*, where* $a \in \mathcal{A}$ *and* $v \in \mathcal{V}_a$*.*
2. *The well-formed formulas (wff) of DL is the smallest set containing the atomic formulas and closed under the Boolean connectives* $\neg, \wedge$*, and* $\vee$*.*

A data table $T = (U, A, \{a_T \mid a \in A\})$ is for a given DL if there is a bijection $f : \mathcal{A} \to A$ such that for every $a \in \mathcal{A}$, $V_{f(a)} = \mathcal{V}_a$. Thus, by somewhat abusing the notation, we will usually denote an atomic formula as (a, v), where $a \in A$ and $v \in V_a$, if the data tables are clear from the context. Intuitively, each element in the universe of a data table corresponds to a data record and an atomic formula, which is in fact an attribute-value pair, describes the value of some attribute in a data record. Thus the atomic formulas (and so the wffs) can be verified or falsified in each data record. This gives rise to a satisfaction relation between the universe and the set of wffs.

Definition 3 *Given a DL and a data table* $T = (U, A, \{a_T \mid a \in A\})$ *for it, the satisfaction relation* $\models$ *between* $x \in U$ *and wffs of DL is defined inductively as follows.*

1. $(T, x) \models (a, v)$ *iff* $a(x) = v$
2. $(T, x) \models \neg\varphi$ *iff* $(T, x) \not\models \varphi$
3. $(T, x) \models \varphi \wedge \psi$ *iff* $(T, x) \models \varphi$ *and* $(T, x) \models \psi$
4. $(T, x) \models \varphi \vee \psi$ *iff* $(T, x) \models \varphi$ *or* $(T, x) \models \psi$

If φ *is a DL wff, the set* $m_T(\varphi)$ *defined by:*

$$m_T(\varphi) = \{x \in U \mid (T, x) \models \varphi\}, \tag{1}$$

is called the meaning of the formula φ *in* T*. If* T *is understood, we simply write* $m(\varphi)$*.*

A formula φ is said to be valid in a data table T, written $T \models \varphi$ or $\models \varphi$ for short when T is clear from the context, if and only if $m(\varphi) = U$. That is, ϕ is satisfied by all individuals in the universe.

A DL wff talks about the properties of individuals in the universe, so it is satisfied by some individuals but falsified by the others. However, the mined knowledge is usually regarding the aggregated or statistical information of all individuals. Obviously, the wffs valid in a data table represent a kind of knowledge that can be induced from the table since they hold for all individuals. However, not all kinds of useful information are in the form of valid wffs. Sometimes, even probabilistic rules are very useful from the viewpoint of knowledge discovery. To quantify the usefulness of the mined rules, some measures have been proposed in [21, 19].

In contrast with DL, where extra meta-level measures must be attached to the wffs, these measures can also be internalized to the language by the so-called generalized quantifiers[4, 9]. This is the approach adopted by the monadic observational predicate calculus(MOPC) in [5]. A wff in DL corresponds to the open formula of MOPC, however, there is not yet the counterpart for the closed formulas of MOPC in DL. To define the corresponding extension in DL, let us call the above-defined DL wffs individual formulas and fix a set of unary and binary quantifiers in advance, then the *aggregate formulas* for a data table T are defined by the following formation rules:

[1] Also called knowledge representation system, information system, or attribute-value system

1. if φ is an individual formula and q is an unary quantifier, then $(q)\varphi$ is an aggregate formula,

2. if φ and ψ are individual formulas and q is a binary quantifier, then $(q)(\varphi, \psi)$ is an aggregate formula,

3. if φ and ψ are aggregate formulas, so are $\neg\varphi$, $\varphi \wedge \psi$, and $\varphi \vee \psi$

Each quantifier q is interpreted by its truth function Tr_q according to [5]. For each unary quantifier q, $Tr_q : N^2 \to \{0, 1\}$ is a 2-place function from natural numbers to $\{0, 1\}$ and for the binary one, $Tr_q : N^4 \to \{0, 1\}$ is a four-place function. Then the satisfaction of an aggregate formula with respect to a data table T is defined as follows:

1. $T \models (q)\varphi$ iff $Tr_q(|m(\varphi)|, |m(\neg\varphi)|) = 1$,

2. $T \models (q)(\varphi, \psi)$ iff $Tr_q(|m(\varphi \wedge \psi)|, |m(\varphi \wedge \neg\psi)|, |m(\neg\varphi \wedge \psi)|, |m(\neg\varphi \wedge \neg\psi)|) = 1$,

3. $T \models \neg\varphi$, $T \models \varphi \wedge \psi$, and $T \models \varphi \vee \psi$ are defined inductively as in the case of individual formulas.

Note that the classical quantifiers $\forall$ and $\exists$ are defined with truth functions $Tr_\forall(n_1, n_2) = 1$ iff $n_2 = 0$ and $Tr_\exists(n_1, n_2) = 1$ iff $n_1 > 0$.

3 General Modal Decision Logic

Just like the models of DL are data tables, those for modal decision logic (MDL) will be structured sets of data tables.

Definition 4 *Let I and J be two fixed sets of indices, then a structured set of data tables (SSDT) is a pair*

$$\mathcal{S} = (\{T_i \mid i \in I\}, \{R_j \mid j \in J\}),$$

where each T_i is a data table and each R_j is a binary relation over $\{T_i \mid i \in I\}$.

In this paper, we will consider only the SSDT $\mathcal{S} = (\{T_i \mid i \in I\}, \{R_j \mid j \in J\})$ satisfying the following assumptions:

- fixed attribute assumption:

  $$\forall i, j \in I, Att(T_i) = Att(T_j),$$

  namely, we assume the data tables in an SSDT are homogeneous. This assumption is necessary since the atomic formula of our logic language will depend on the set of attributes, so to make our language interpretable in all data tables, they must have the same set of attributes.

- constant domain assumption: for

  $$\forall i, j \in I, Uni(T_i) = Uni(T_j).$$

  In other words, we assume the set of individuals keeps unchanged between different data tables. This assumption is not essential, however, it will simplify the semantics of our logic.

- finite table assumption: I is finite. This is a practical assumption since we will consider only finite amount of data in the knowledge discovery process.

The syntax of MDL is an extension of DL with the following rule:

- if φ is an individual (resp. aggregate) formula, so are $[j]\varphi$ and $\langle j\rangle\varphi$ for any $j \in J$.

Given an SSDT $\mathcal{S} = (\{T_i \mid i \in I\}, \{R_j \mid j \in J\})$, the satisfaction of individual formulas are defined by

1. $(T_i, x) \models_\mathcal{S} [j]\varphi$ iff for all T such that $(T_i, T) \in R_j$, $(T, x) \models_\mathcal{S} \varphi$

2. $(T_i, x) \models_\mathcal{S} \langle j\rangle\varphi$ iff there exists T such that $(T_i, T) \in R_j$ and $(T, x) \models_\mathcal{S} \varphi$

3. the satisfaction of classical formulas is defined as in the case of DL.

The satisfaction of aggregate formulas can be analogously defined and is denoted by $T \models_\mathcal{S} \varphi$.

4 Case Studies

In MDL, there is a set of modal operators $[j]$ which are interpreted semantically by the binary relations R_j over the data tables of an SSDT, however, it remains unspecified how the binary relations are constructed. In the following sections, we will study some cases in which the binary relations between data tables arise naturally from the application problems.

4.1 Uncertain decision logic

The first problem we trying to deal with is regarding uncertain data tables.

Definition 5 *An uncertain data table is a triplet*

$$T = (U, A, \{a_T \mid a \in A\})$$

where

- *U and A are defined as in the standard data tables and*

- *for each $a \in A$, $a_T : U \to (2^V_a - \{\emptyset\})$ is a set-valued function, where V_a is the domain of values for a.*

For each $x \in U$, $a_T(x)$ denotes the set of possible values for its attribute a. Since $a_T(x)$ may contain more than one values, this means that we do not have the exact knowledge about what the value is. In particular, if $a_T(x) = V_a$, then we have null information for the particular x on its attribute a. Given an uncertain data table $T = (U, A, \{a_T \mid a \in A\})$, a *possible realization* of T is a data table $T' = (U, A, \{a_{T'} \mid a \in A\})$ such that for any $x \in U$ and $a \in A$, $a_{T'}(x) \in a_T(x)$. Let $\Xi(T)$ denote the set of all possible realization of T, then the SSDT for T is defined as

$$\mathcal{S} = (\Xi(T), R_u)$$

where R_u is the universal relation, i.e., for each T_i and $T_j \in \Xi(T)$, $(T_i, T_j) \in R_u$.

Thus the language of uncertain modal logic(UDL) contains only two modalities $[u]$ and $\langle u \rangle$ and we will denote them by the ordinary alethic modalities $\Box$ and $\Diamond$ respectively.

Example 1 The following table is simplified from one in [10] and used in the evaluation of researchers for a leadership in a computer science grant.

Researcher	Talent	Grade	d
1	{math, cs}	{B, MSc, Ph.D}	good
2	{cs}	{Ph.D}	excel.
3	{math}	{MSc}	good
4	{math, phil.}	{B, MSc, Ph.D}	good

where d denotes the decision attribute. There are in total 36 possible realizations for the uncertain data table. Among them is the following one

Researcher	Talent	Grade	d
1	math	Ph.D	good
2	cs	Ph.D	excel.
3	math	MSc	good
4	math	Ph.D	good

Thus according to the semantics of UDL, the following aggregate formula can be verified in each possible realization,

$$\Diamond\forall((\text{Talent}, \text{math}) \supset (\text{d}, \text{good}))$$

■

4.2 Epistemic decision logic

For a data table $T = (U, A, \{a_T \mid a \in A\})$ and a subset of attributes $B \subseteq A$, let us define the information function Inf^T_B as a mapping from U to the vectors of attribute values by

$$Inf^T_B(x) = (a_T(x))_{a \in B}.$$

Then an equivalence relation $\equiv^T_B$ over U can be formulated with respect to T and B by $x \equiv^T_B y$ iff $Inf^T_B(x) = Inf^T_B(y)$. For each subset E of U, let $Inf^T_B(E)$ denote the multiset $\{Inf^T_B(x) \mid x \in E\}$. Then two data tables $T_1 = (U, A, \{a_{T_1} \mid a \in A\})$ and $T_2 = (U, A, \{a_{T_2} \mid a \in A\})$ are said to be epistemically indistinguishable with respect to $B \subseteq A$ iff $\equiv^{T_1}_B$ is the same as $\equiv^{T_2}_B$ and for each equivalence class E of $\equiv^{T_1}_B$, $Inf^{T_1}_{A-B}(E) = Inf^{T_2}_{A-B}(E)$.

Now, given a data table T, we can define its epistemic SSDT as

$$\mathcal{S} = (\mathcal{T}, \{R_B \mid B \subseteq A\})$$

where $\mathcal{T}$ is the set of all data tables which are epistemically indistinguishable with T with respect to some $B \subseteq A$ and for each R_B and any two tables $T_i, T_j \in \mathcal{T}$, $(T_i, T_j) \in R_B$ iff T_i and T_j are epistemically indistinguishable with respect to B. Following the notations of epistemic logic[2], the modalities $[B]$ will be denoted by K_B.

The epistemic decision logic is useful for the reasoning of data security in the KDD process. The main challenge is to protect personal sensitive information in the release of microdata set, i.e. a set of records containing information on individuals. To achieve this, the re-identification of individuals have to be avoided. In other words, it is necessary to prevent the possibility of deducing which record corresponds to a particular individual even the explicit identifier of the individual is not contained in the released information. This problem has been studied in some literatures[6, 7, 16, 17, 18].

Since useful knowledge can be induced from the data tables, it is desirable that they can be released to the public. To protect the privacy of the individuals whose personal information is contained in the table, the attributes of a data table can be divided into three sets. The first one consists of the *key attributes*, which can be used to identify to whom a data record belongs. So they are always masked off before the table is released. Since the key attributes uniquely determine the individuals, we can assume that they are associated with elements in the universe U and omit them from now on. Second, we have a set of *public attributes*, the values of which are known to the public. For example, in [18], it is pointed that some attributes like birth-date, gender, ethnicity, etc., are included in some public databases such as census data or voter registration lists. These attributes, if not appropriately generalized, may be used to re-identify an individual's record in a medical data table, and this will cause privacy violation. The last kind of attributes is the *confidential ones*, the values of which we have to protect. It is often the case that there is an asymmetry between the values of a confidential attribute. For example, if the attribute is the HIV test result, then the revelation of a '$+$' value may cause serious privacy invasion, whereas it does not matter to know an individual have a '$-$' value.

Example 2 Let B be the set of public attributes in a table of medical records T and $Att(T) - B = \{HIV\}$ be the confidential one. Thus if φ denotes a piece of sensitive information, say $\varphi = (HIV, +)$, then $(T, x) \models_{\mathcal{S}} K_B\varphi$ means that the release of data table will result in the privacy leakage of x, where $\mathcal{S}$ is the epistemic SSDT for T. ■

4.3 Temporal decision logic

Definition 6 *A (linear-time) temporal SSDT is of the form*

$$\mathcal{S} = (\{T_i \mid 0 \leq i \leq n-1\}, \{R_+, R_<\})$$

where

- *each T_i is a data table,*
- *$(T_i, T_j) \in R_+$ iff $j = i + 1$, and*
- *$(T_i, T_j) \in R_<$ iff $i < j$.*

The modalities $[+]$ and $[<]$ corresponds to the "next" and "future" operators in ordinary temporal logic and will be denoted by $\bigcirc$ and $\Box$ respectively. Furthermore, we abbreviate a sequence of n modal operators $\bigcirc$ by $\bigcirc^n$

The temporal decision logic may be applied to the mining of sequence patterns. For example, in [11, 12], sequence pattern mining is used in the construction of intrusion detection rules. The main techniques for intrusion detection are misuse detection and anomaly detection. For the former, the "signatures" of known attacks, i.e., the patterns of attack behavior and effects are used to identify a matched activity as an attack instance, whereas the latter uses established normal profiles, i.e., the expected behavior, to identify any unacceptable deviation as possibly the result of an attack. In [12], the data mining technique is applied to a set of audit records. One kind of data they considered is the BSM data developed and distributed by MIT Lincoln Lab for the 1999 DARPA evaluation of intrusion detection systems. The data contains audit records of all *sendmail* sessions during a period of time. Each audit record corresponds to a UNIX system call made by *sendmail*. The attributes of each record include the system call name, the user and group IDs, the name of object accessed by the system call, and arguments, etc. The expected patterns to be discovered is of the probabilistic rule $Pr(s_n \mid s_0, \cdots, s_{n-1})$ which is the probabilistic prediction of the $(n+1)$-th system call given the previous n system calls in a session.

Example 3 To model the intrusion detection application, we consider the universe U as the set of all sessions during a period of time. For the purpose of simplification, we assume all sessions start at the same time. The attributes of each data table are just those for the system calls in the audit records. For $0 \leq i \leq n$, the data table T_i contains the system calls made at time i by each session. Then the expected patterns to be mined will be expressed by the following formula

$$(\varphi_0 \wedge \bigcirc\varphi_1 \wedge \cdots \wedge \bigcirc^{n-1}\varphi_{n-1}) \Rightarrow_r \bigcirc^n\varphi_n$$

where each φ_i's denote the properties of system calls and $\Rightarrow_r$ is a binary quantifier defined by $Tr_{\Rightarrow_c}(n_1, n_2, n_3, n_4) = 1$ iif $\frac{n_1}{n_1+n_2} \geq r$. ■

5 Conclusion

Just like DL is used in the knowledge representation for data mining of a single data table, the MDL provides a uniform framework for representing knowledge mined from a collection of multiple data tables. The sets of data tables are structured in the sense that some relationship exists between their elements. We interpret the MDL formulas in such structured sets of data tables. In particular, the modalities are interpreted with respect to the relations between the data tables according to the Kripke semantics. Three instances of MDL are presented to illustrate the application potentials of the MDL representation formalism.

References

[1] P. Blackburn. "Representation, reasoning, and relational structures: a hybrid logic manifesto". *Logic Journal of IGPL*, 8(3):339–365, 2000.

[2] R. Fagin, J.Y. Halpern, Y. Moses, and M.Y. Vardi. *Reasoning about Knowledge*. MIT Press, 1996.

[3] T.F. Fan, W.C. Hu, and C.J. Liau. "Decision logics for knowledge representation in data mining". In *Proceedings of the 25th Annual International Computer Software and Applications Conference(COMPSAC)*, pages 626–631. IEEE Press, 2001.

[4] P. Gärdenfors, editor. *Generalized Quantifiers*. Dordrecht: Reidel Publishing Company, 1987.

[5] P. Hájek. "Logics for data mining (GUHA rediviva)". *Neural Network World*, 10:301–311, 2000.

[6] T.-s. Hsu, C.-J. Liau, and D.-W. Wang. A logical model for privacy protection. In *Proceedings of the 4th International Conference on Information Security*, LNCS 2200, pages 110–124. Springer-Verlag, 2001.

[7] A.J. Hundepool and L.C.R.J. Willenborg. "μ- and τ-ARGUS: Software for statistical disclosure control". In *Proceedings of the 3rd International Seminar on Statistical Confidentiality*, 1996.

[8] R. Kruse, C. Borgelt, and D. Nauck. “Fuzzy data analysis: challenges and perspectives”. In *Proceedings of the 8th IEEE International Conference on Fuzzy Systems*, pages 1211–1216, San Francisco, CA, 1999. IEEE.

[9] M. Krynicki, M. Mostowski, and L.W. Szczerba, editors. *Quantifiers: Logics, Models and Computation*. Kluwer Academic Publishers, 1995.

[10] M. Kryszkiewicz and H. Rybiński. “Reducing information systems with uncertain attributes”. In Z. W. Raś and M. Michalewicz, editors, *Proceedings of the 9th ISMIS*, LNAI 1079, pages 285–294. Springer-Verlag, 1996.

[11] W. Lee, S.J. Stolfo, and K.W. Mok. A data mining framework for building intrusion detection models. In *Proceedings of the 1999 IEEE Symposium on Security and Privacy*, pages 120–132, 1999.

[12] W. Lee and D. Xiang. Information-theoretic measures for anomaly detection. In *Proceedings of the 2001 IEEE Symposium on Security and Privacy*, pages 130–143, 2001.

[13] C.J. Liau and D.R. Liu. “A logical approach to fuzzy data analysis”. In J.M. Zytkow and J. Rauch, editors, *Proceedings of the Third European Conference on Principles of Data Mining and Knowledge Discovery*, LNAI 1704, pages 412–417. Springer-Verlag, 1999.

[14] C.J. Liau and D.R. Liu. “A Possibilistic decision logic with applications”. *Fundamenta Informaticae*, 46(3), 2001.

[15] Z. Pawlak. *Rough Sets–Theoretical Aspects of Reasoning about Data*. Kluwer Academic Publishers, 1991.

[16] P. Samarati. “Protecting respondents’ identities in microdata release”. *IEEE Transactions on Knowledge and Data Engineering*, 13(6):1010–1027, 2001.

[17] P. Samarati and L. Sweeney. Protecting privacy when disclosing information: k-anonymity and its enforcement through generalization and suppression. Technical report SRI-CSL-98-04, Computer Science Laboratory, SRI International, 1998.

[18] L. Sweeney. “Guaranteeing anonymity when sharing medical data, the Datafly system”. In *Proceedings of American Medical Informatics Association*, 1997.

[19] Y.Y. Yao and C.J. Liau. “A generalized decision logic language for granular computing”. In *Proceedings of the 11th IEEE International Conference on Fuzzy Systems*, page to appear. IEEE Press, 2002.

[20] Y.Y. Yao and Q. Liu. “A generalized decision logic in interval-set-valued information tables”. In N. Zhong, A. Skowron, and S. Ohsuga, editors, *New Directions in Rough Sets, Data Mining, and Granular-Soft Computing*, LNAI 1711, pages 285–293. Springer-Verlag, 1999.

[21] Y.Y. Yao and N. Zhong. “An analysis of quantitative measures associated with rules”. In *Proceedings of the 2nd Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 479–488. IEEE Press, 1999.

[22] L.A. Zadeh. “Fuzzy logic = computing with words”. *IEEE Transactions on Fuzzy Systems*, 4:103–111, 1996.

Mining a Complete Set of Interesting Generalized Fuzzy Association Rules

Tzung-Pei Hong
Dept. of Electrical Engineering
National Univ. of Kaohsiung
tphong@nuk.edu.tw

Kuei-Ying Lin
Chunghwa Telecom Lab.
ying120@cht.com.tw

Shyue-Liang Wang
Dept. of Info. Management
I-Shou University
slwang@isu.edu.tw

Abstract

Most conventional data-mining algorithms identify the relationships among transactions using binary values and find rules at a single concept level. Transactions with quantitative values and items with hierarchy relations are, however, commonly seen in real-world applications. In the past, we proposed a fuzzy mining algorithm for extracting interesting generalized association rules from transactions stored as quantitative values. In that algorithm, each attribute used only the linguistic term with the maximum cardinality in the mining process. The fuzzy generalized association rules derived in this way are however not complete, meaning some possible fuzzy association rules may be missed. This paper thus modifies it and proposes a new algorithm for extracting all possible fuzzy interesting knowledge from transactions stored as quantitative values. The proposed algorithm can derive a more complete set of rules but with more computation time than the previous method. Trade-off thus exists between the computation time and the completeness of rules. Choosing an appropriate mining method thus depends on the requirements of the application domains.

1. Introduction

Deriving association rules from transaction databases is most commonly seen in data mining [1][2][6][8-10] [17-18]. Most previous studies concentrated on showing how binary-valued transaction data on a single level of items may be handled. However, transaction data in real-world applications usually consist of quantitative values and items are often organized in a taxonomy, so designing a sophisticated data-mining algorithm able to deal with quantitative data on multiple levels of items presents a challenge to workers in this research field.

Relevant taxonomies of data items are usually predefined in real-world applications and can be represented by hierarchy trees. Terminal nodes on the trees represent actual items appearing in transactions; internal nodes represent classes or concepts formed by lower-level nodes. A simple example is given in Figure 1.

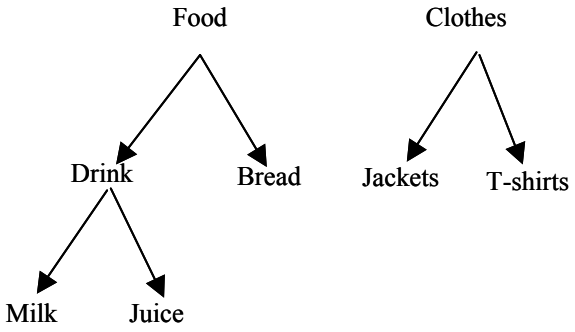


Figure 1. An example of taxonomic structures.

In this example, food is assumed to fall into two classes: drink and bread. Drink can be further classified into milk and juice. Similarly, clothes are divided into jackets and T-shirts. Only terminal items (milk, juice, bread, jacket and T-shirt) can appear in transactions.

Srikant and Agrawal proposed a method for finding generalized association rules on multiple levels [19]. Their mining process can be divided into four phases. In the first phase, all the ancestors of the items in each given transaction are added according to the predefined taxonomy. In the second phase, candidate itemsets are generated and counted by scanning the expanded transaction data. In the third phase, all possible generalized association rules are induced from the large itemsets found in the second phase. In the fourth phase, uninteresting association rules are pruned away and interesting rules are output.

Recently, the fuzzy set theory has been used to effectively process quantitative data. This theory was first proposed by Zadeh in 1965 [21] and is primarily concerned with quantifying and reasoning using natural language in which words can have ambiguous meanings. It is being used more and more frequently in intelligent systems because of its simplicity and similarity to human reasoning [16]. Several fuzzy learning algorithms for inducing rules from a given set of data have been designed and used with good effect in specific domains [3-5, 9, 11-13, 20].

In the past, we proposed a sophisticated fuzzy data-mining algorithm able to deal with quantitative data under a given taxonomy [14]. In that algorithm, each

attribute used only the linguistic term with the maximum cardinality in the mining process. The number of items was thus the same as that of the original attributes, making the processing time reduced. The fuzzy association rules derived in that way are however not complete, meaning some possible fuzzy association rules may be missed. This paper thus modifies it and proposes a new fuzzy data-mining algorithm for extracting all possible fuzzy interesting association rules from transactions stored as quantitative values. The proposed algorithm considers all the important linguistic terms in the mining process, thus being able to derive a more complete set of rules but with more computation time than the previous method. Trade-off thus exists between the computation time and the completeness of rules. Experiments are also made to show the trade-off effects.

The remaining parts of this paper are organized as follows. Fuzzy-set concepts are briefly reviewed in Section 2. Notation used in this paper is defined in Section 3. An algorithm for mining a complete set of fuzzy interesting association rules is proposed in Section 4. An example to illustrate the proposed algorithm is given in Section 5. Experiments are described in Section 6. Discussion and conclusion are stated in Section 7.

2. Review of fuzzy set concepts

Fuzzy set theory was first proposed by Zadeh in 1965 [21]. Fuzzy set theory is primarily concerned with quantifying and reasoning using natural language in which words can have ambiguous meanings. This can be thought of as an extension of traditional crisp sets, in which each element must either be in or not be in a set.

Formally, the process by which individuals from a universal set X are determined to be either members or non-members of a crisp set can be defined by a characteristic or discrimination function. For a given crisp set A, this function assigns a value $\mu_A(x)$ to every $x \in X$ such that

$$\mu_A(x)=\begin{cases}1 & \text{if and only if } x\in A\\ 0 & \text{if and only if } x\notin A.\end{cases}$$

Thus, the function maps elements of the universal set to the set containing 0 and 1. This function can be generalized such that the values assigned to the elements of the universal set fall within specified ranges, referred to as the membership grades of these elements in the set. Larger values denote higher degrees of set membership. Such a function is called the membership function, $\mu_A(x)$, by which a fuzzy set A is usually defined. This function is represented by

$$\mu_A : X \to [0,1],$$

where [0, 1] denotes the interval of real numbers from 0 to 1, inclusive. The function can also be generalized to any real interval instead of [0, 1].

A special notation is often used in the literature to represent fuzzy sets. Assume that x_1 to x_n are the elements in fuzzy set A, and μ_1 to μ_n are, respectively, their grades of membership in A. A is then represented as follows:

$$A = \mu_1/x_1 + \mu_2/x_2 + \ldots + \mu_n/x_n.$$

An *α-cut* of a fuzzy set A is a crisp set A_α that contains all elements in the universal set X with membership grades in A greater than or equal to a specified value α. This definition can be written as:

$$A_\alpha = \{x \in X \mid \mu_A(x) \geq \alpha\}.$$

The *scalar cardinality* of a fuzzy set A defined on a finite universal set X is the summation of the membership grades of all the elements of X in A. Thus,

$$|A| = \sum_{x \in X} \mu_A(x).$$

Among operations on fuzzy sets are the basic and commonly used *complementation*, *union* and *intersection,* as proposed by Zadeh.

The complementation of a fuzzy set A is denoted by $\neg A$, and the membership function of $\neg A$ is given by:

$$\mu_{\neg A}(x) = 1 - \mu_A(x), \quad \forall x \in X.$$

The intersection of two fuzzy sets A and B is denoted by $A \cap B$, and the membership function of $A \cap B$ is given by:

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in X.$$

The union of fuzzy sets A and B is denoted by $A \cup B$, and the membership function of $A \cup B$ is given by:

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in X.$$

The above concepts will be used in our proposed algorithm to mine a complete set of generalized fuzzy association rules.

3. Notation

The following notation is used in this paper:

n: the number of transactions;
D_i: the i-th transaction, $1 \leq i \leq n$;
m: the number of expanded items (including the original items);

I_j: the j-th expanded item, $1 \le j \le m$;
h_j: the number of fuzzy regions for I_j;
R_{jl}: the l-th fuzzy region of I_j, $1 \le l \le h_j$;
v_{ij}: the quantitative value of I_j in D_i;
f_{ij} : the fuzzy set converted from v_{ij};
f_{ijl}: the membership value of region R_{jl} in D_i;
$count_{jl}$: the summation of f_{ijl} values for $i = 1$ to n;
α: the predefined minimum support threshold;
λ: the predefined minimum confidence threshold;
R: the predefined interest threshold;
C_r: the set of candidate itemsets with r items (regions);
L_r: the set of large itemsets with r items (regions).

4. Mining a complete set of interesting generalized fuzzy association rules

In the past, we proposed a fuzzy generalized mining approach [14], in which each attribute used only the linguistic term with the maximum count in the mining process. In this paper, all the linguistic terms are used. Linguistic terms belonging to the same attribute cannot, however, belong to the same itemset. The computation is more complex than that in [14] since all possible linguistic terms are used in calculating large itemsets, but the derived set of association rules is more complete.

The proposed fuzzy mining algorithm first forms expanded transactions as Srikant and Agrawal's method did [19]. It then uses membership functions to transform each quantitative value into a fuzzy set in linguistic terms and adopts an iterative search approach to find large itemsets. The algorithm then calculates the scalar cardinality of each linguistic term on all the transaction data. A mining process using fuzzy counts is then performed to find fuzzy generalized association rules. Fuzzy interest requirements, which are similar to those proposed by Srikant and Agrawal [19], are then checked to remove uninteresting rules. Details of the proposed fuzzy mining algorithm are stated below.

The mining algorithm for fuzzy interesting generalized association rules:

INPUT: A body of n quantitative transaction data, a set of membership functions, a predefined taxonomy, a predefined minimum support value α, a predefined minimum confidence value λ, and a predefined interest threshold R.

OUTPUT: A complete set of fuzzy interesting generalized association rules.

STEP 1: Add ancestors of appearing items to transactions and calculate their quantities.

STEP 2: Transform the quantitative value v_{ij} of each expanded item name I_j in transaction datum D_i into a fuzzy set f_{ij} represented as:

$$\left(\frac{f_{ij1}}{R_{j1}} + \frac{f_{ij2}}{R_{j2}} + ... + \frac{f_{ijh}}{R_{jh}} \right)$$

using the given membership functions, where h is the number of fuzzy regions for I_j, R_{jl} is the l-th fuzzy region of I_j, $1 \le l \le h$, and f_{ijl} is v_{ij}'s fuzzy membership value in region R_{jl}.

STEP 3: Collect the fuzzy regions (linguistic terms) with membership values larger than zero to form the candidate set C_1.

STEP 4: Calculate the scalar cardinality $count_{jl}$ of each fuzzy region R_{jl} in C_1 from all the transaction data as:

$$count_{jl} = \sum_{i=1}^{n} f_{ijl} \ .$$

STEP 5: Check whether the value $count_{jl}$ of each fuzzy region R_{jl} in C_1 is larger than or equal to the predefined minimum support value α . If R_{jl} satisfies the above condition, put it in the set of large 1-itemsets (L_1). That is:

$$L_1 = \left\{ R_{jl} \middle| \, count_{jl} \ge \alpha, R_{jl} \in C_1 \right\}.$$

STEP 6: IF L_1 is not null, then do the next step; otherwise, exit the algorithm.

STEP 7: Generate the candidate set C_2 from L_1. Each 2-itemset in C_2 contains two regions in L_1, and these two regions cannot have the same item name. Also, the item names of these two regions may not have hierarchy relations in the taxonomy. All the possible 2-itemsets are collected as C_2.

STEP 8: Do the following substeps for each newly formed candidate 2-itemset s with regions (s_1, s_2) in C_2:

(a) Calculate the fuzzy value of s in each transaction datum D_i as:

$$f_{is} = f_{is_1} \Lambda f_{is_2} \ ,$$

where f_{is} is the membership value of region s_j in D_i. If the minimum operator is used for the intersection, then:

$$f_{is} = \min(f_{is1}, f_{is2}) \ .$$

(b) Calculate the scalar cardinality of s in all the transaction data as:

$$count_s = \sum_{i=1}^{n} f_{is} \ .$$

(c) If $count_s$ is larger than or equal to the predefined minimum support value α, put s in L_2.

STEP 9: IF L_2 is null, then exit the algorithm; otherwise, do the next step.

STEP 10: Set $r = 2$, where r is used to represent the number of regions stored in the current large itemsets.

STEP 11: Generate the candidate set C_{r+1} from L_r in a way similar to that in the *apriori* algorithm [4]. That is, the algorithm first joins L_r and L_r assuming that r-1 items in the two itemsets are the same and the other one is different. Store in C_{r+1} the itemsets which have all their sub-r-itemsets in L_r.

STEP 12: Do the following substeps for each newly

formed (r+1)-itemset s with regions (s_1, s_2, ..., s_{r+1}) in C_{r+1}:

(a) Calculate the fuzzy value of s in each transaction datum D_i as

$$f_{is} = f_{is_1} \Lambda f_{is_2} \Lambda \ldots \Lambda f_{is_{r+1}},$$

where f_{is_j} is the membership value of region s_j in D_i. If the minimum operator is used for the intersection, then:

$$f_{is} = \underset{j=1}{\overset{r+1}{Min}} f_{is_j}.$$

(b) Calculate the scalar cardinality of s in all the transaction data as:

$$count_s = \sum_{i=1}^{n} f_{is}.$$

(c) If $count_s$ is larger than or equal to the predefined minimum support value α, put s in L_{r+1}.

STEP 13: If L_{r+1} is null, then do the next step; otherwise, set $r=r+1$ and repeat STEPs 11 to 13.

STEP 14: Construct the fuzzy association rules for all the large q-itemset s containing regions (s_1, s_2, ..., s_q), $q \geq 2$, using the following substeps:

(a) Form all possible association rules, thusly:

$$s_1 \Lambda \ldots \Lambda s_{k-1} \Lambda s_{k+1} \Lambda \ldots \Lambda s_q \rightarrow s_k, \; k=1 \text{ to } q.$$

(b) Calculate the confidence values of all association rules using the formula:

$$\frac{\sum_{i=1}^{n} f_{is}}{\sum_{i=1}^{n} (f_{is_1} \Lambda \ldots \Lambda f_{is_{K-1}}, f_{is_{K+1}} \Lambda \ldots \Lambda f_{is_q})}.$$

STEP 15: Keep the rules with confidence values larger than or equal to the predefined confidence threshold λ.

STEP 16: Output the rules without ancestor rules (by replacing the items in a rule with their ancestors in the taxonomy) to users as interesting rules.

STEP 17: For each remaining rule s represented as

$$s_1 \Lambda s_2 \Lambda \ldots \Lambda s_r \rightarrow s_{r+1},$$

find the closest ancestor rule t, represented as

$$t_1 \Lambda t_2 \Lambda \ldots \Lambda t_r \rightarrow t_{r+1};$$

calculate the support interest measure $I_{support}(s)$ of s as:

$$I_{support}(s) = \frac{count_s}{\frac{\Pi_{k=1}^{r+1} count_{s_k}}{\Pi_{k=1}^{r+1} count_{t_k}} \times count_t},$$

and the confidence interest measure $I_{confidence}(s)$ of s as:

$$I_{confidence}(s) = \frac{confidence_s}{\frac{count_{s_{r+1}}}{count_{t_{r+1}}} \times confidence_t},$$

where $confidence_s$ and $confidence_t$ are respectively the confidence values of rules s and t; output the rules with their support interest measures or confidence interest measures larger than or equal to the predefined interest threshold R to users as interesting rules.

Note that in Steps 16 and 17, an ancestor of a fuzzy rule is formed by replacing the items in the rule with their ancestors in the taxonomy, but the linguistic terms in both the rules must be the same. The rules output after STEP 17 can then serve as meta-knowledge concerning the given transactions.

5. An example

Assume a data set includes the six transactions shown in Table 1. Each transaction includes a transaction ID and some purchased items. Each item is represented by a tuple (item name, item amount).

Table 1. Six transactions in this example

Transaction ID	Items
1	(Milk, 3) (Bread, 4) (T-shirt, 2)
2	(Juice, 3) (Bread, 7) (Jacket, 7)
3	(Juice, 2) (Bread, 10) (T-shirt, 5)
4	(Bread, 9) (T-shirt, 10)
5	(Milk, 7) (Jacket, 8)
6	(Juice, 2) (Bread, 8) (Jacket, 10)

Assume that the predefined taxonomy is the same as that in Figure 1. For convenience, simple symbols are used to represent items and groups. For example, symbol A represents milk and symbol B represents juice. The new representation of the given taxonomy is as shown in Figure 2.

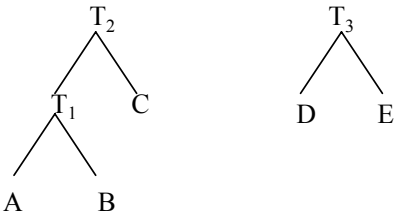


Figure 2. The new taxonomy representation

Also assume that the fuzzy membership functions are the same for all the items and are as shown in Figure 3. For the transaction data in Table 1, all the expanded transactions are first formed. They are then transformed into fuzzy sets. The fuzzy count of each region is checked against the predefined minimum support value α. Assume in this example, α is set at 1.5. After Step 5, the set of L_1 is shown in Table 2.

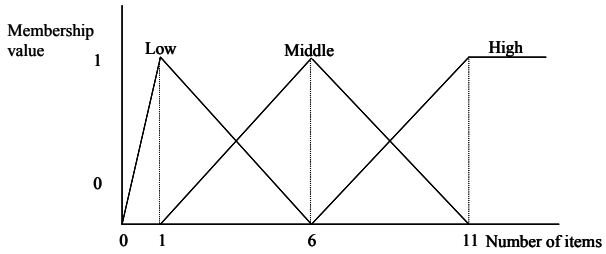


Figure 3. The used membership functions

Table 2. The set of large 1-itemsets

Itemset	count	Itemset	count
B.Low	2.2	T_1*.Middle*	2.0
C.Middle	2.6	T_2*. Middle*	2.4
C.High	2.0	T_2*.High*	3.6
D.Middle	1.6	T_3*.Middle*	2.8
T_1*.Low*	2.8	T_3*.High*	2.2

After Step 13, the set of L_2 includes *(B.Low, T_3.Middle), (C.Middle, T_1.Low), (C.Middle, T_3.Middle), (T_1.Low, T_3.Middle), (T_1.Middle, T_3.Middle), (T_2.High, T_3.Middle)* and *(T_2.High, T_3.High)*. L_3 is an empty set.

The association rules are then constructed for each large itemset. Assume the given confidence threshold λ is set at 0.75 in this example. The following four rules are thus kept:

1. If *C = Middle, then* T_1 *= Low,* with a support value of 2.0 and a confidence value of 0.77.
2. If T_1 *= Middle,* then T_3 *= Middle,* with a support value of 1.6 and a confidence value of 0.8.
3. If T_3 *= Middle*, then T_2 *= High,* with a support value of 2.4 and a confidence value of 0.86.
4. If T_3 *= High,* then T_2 *= High,* with a support value of 1.8 and a confidence value of 0.82.

All the above rules have no ancestor rules mined out. They are thus output as interesting rules.

6. Experiments

The experiments were implemented in C on a Pentium-III 700 personal computer. The number of levels was set at 3. There were 64 purchased items (terminal nodes) on level 3, 16 generalized items on level 2, and 4 generalized items on level 1. Each non-terminal node had four branches. In each data set, numbers of purchased items in transactions were first randomly generated. The purchased items and their quantities in each transaction were then generated. An item could not be generated twice in a transaction. Various min-support and min-confidence values were tested.

The proposed approach adopted in this paper is compared with the approach in [14], which uses only the linguistic term with the maximum cardinality for each quantitative item and thus focuses on the most important linguistic terms. The proposed approach here uses all possible linguistic terms in the mining processes, and can thus derive a more complete set of rules. The relationships between numbers of rules mined and minimum support values for an average of 10 purchased items in 10000 transactions and a minimum confidence value set at 0.7 by these two approaches are shown in Figure 4.

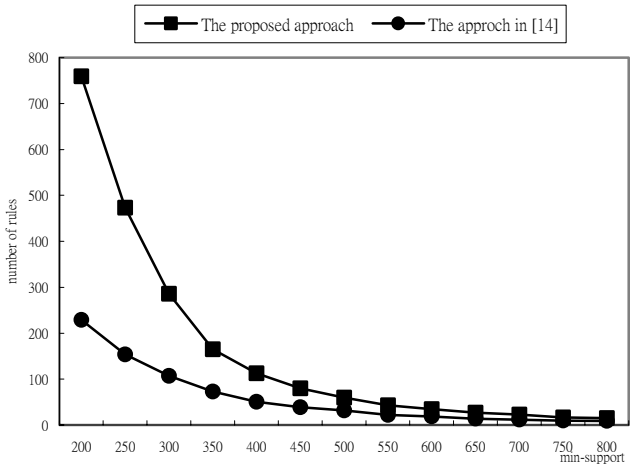


Figure 4. A comparison for the relationships between numbers of rules and minimum support values.

From Figure 4, it is easily seen that the numbers of rules by the proposed approach here are greater than those by the approach in [14]. This is consistent with our derivation. The execution-time relationships are shown in Figure 5.

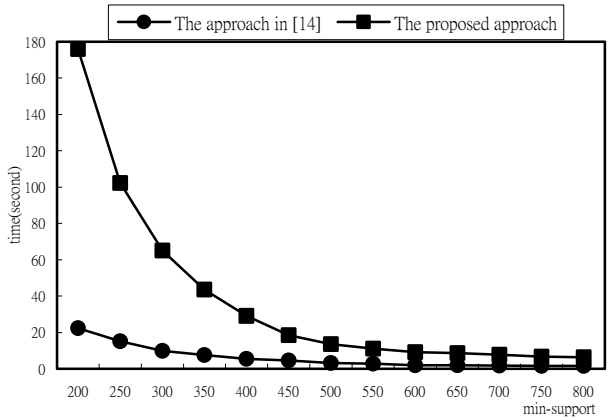


Figure 5. A comparison for the relationships between execution times and minimum support values.

From Figure 5, it is easily seen that the execution times needed by the proposed approach here are more than those needed by the approach in [14]. This is also quite consistent with our derivation.

7. Conclusion

In this paper, we have proposed a fuzzy mining algorithm for processing transaction data with quantitative values and discovering interesting generalized association rules from them. The proposed algorithm can derive a more complete set of rules than the method proposed in [14] although it needs more computation time. Trade-off thus exists between the computation time and the completeness of rules. Choosing an appropriate mining method thus depends on the requirements of the application domains.

References

[1] R. Agrawal, T. Imielinksi and A. Swami, “Mining association rules between sets of items in large database,“ *The 1993 ACM SIGMOD Conference,* Washington DC, USA, 1993.

[2] R. Agrawal, T. Imielinksi and A. Swami, “Database mining: a performance perspective,” *IEEE Transactions on Knowledge and Data Engineering,* Vol. 5, No. 6, 1993, pp. 914-925.

[3] A. F. Blishun, “Fuzzy learning models in expert systems,” *Fuzzy Sets and Systems,* Vol. 22, 1987, pp. 57-70.

[4] L. M. de Campos and S. Moral, “Learning rules for a fuzzy inference model,” *Fuzzy Sets and Systems*, Vol. 59, 1993, pp. 247-257.

[5] R. L. P. Chang and T. Pavliddis, “Fuzzy decision tree algorithms,” *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 7, 1977, pp. 28-35.

[6] M. S. Chen, J. Han and P. S. Yu, “Data mining: an overview from a database perspective,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 8, No. 6, 1996.

[7] M. Delgado and A. Gonzalez, “An inductive learning procedure to identify fuzzy systems,” *Fuzzy Sets and Systems*, Vol. 55, 1993, pp. 121-132.

[8] T. Fukuda, Y. Morimoto, S. Morishita and T. Tokuyama, "Mining optimized association rules for numeric attributes," *The ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, June 1996, pp. 182-191.

[9] J. Han and Y. Fu, “Discovery of multiple-level association rules from large database,” *The International Conference on Very Large Databases*, 1995.

[10] T. P. Hong, C. S. Kuo and S. C. Chi, "A data mining algorithm for transaction data with quantitative values," *Intelligent Data Analysis,* Vol. 3, No. 5, 1999, pp. 363-376.

[11] T. P. Hong and J. B. Chen, "Finding relevant attributes and membership functions," *Fuzzy Sets and Systems,* Vol.103, No. 3, 1999, pp. 389-404.

[12] T. P. Hong and J. B. Chen, "Processing individual fuzzy attributes for fuzzy rule induction," *Fuzzy Sets and Systems*, Vol. 112, No. 1, 2000, pp. 127-140.

[13] T. P. Hong and C. Y. Lee, "Induction of fuzzy rules and membership functions from training examples," *Fuzzy Sets and Systems***,** Vol. 84, 1996, pp. 33-47.

[14] T. P. Hong, K. Y. Lin and S. L. Wang, “Mining generalized association rules from quantitative data”, *The International Workshop on Intelligent Systems Resolutions in The Eighth Bellman Continuum*, 2000, pp. 75-78.

[15] T. P. Hong and S. S. Tseng, “A generalized version space learning algorithm for noisy and uncertain data,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 9, No. 2, 1997, pp. 336-340.

[16] A. Kandel, *Fuzzy Expert Systems*, CRC Press, Boca Raton, 1992, pp. 8-19.

[17] R. Srikant, Q. Vu and R. Agrawal, “Mining association rules with item constraints,” *The Third International Conference on Knowledge Discovery in Databases and Data Mining,* Newport Beach, California, August 1997, pp.67-73.

[18] R. Srikant and R. Agrawal, “Mining quantitative association rules in large relational tables,” *The 1996 ACM SIGMOD International Conference on Management of Data*, Monreal, Canada, June 1996, pp. 1-12.

[19] R. Srikant and R. Agrawal, “Mining Generalized Association Rules," *The International Conference on Very Large Databases*, 1995.

[20] C. H. Wang, J. F. Liu, T. P. Hong and S. S. Tseng, “A fuzzy inductive learning strategy for modular rules,” *Fuzzy Sets and Systems*, Vol.103, No. 1, 1999, pp. 91-105.

[21] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, Vol. 8, No. 3, 1965, pp. 338-353.

Incorporating Generalization and Specialization Mutation into GEC with Micro Partitioning of Continuous Data

William W. Hsu and Ching-Chi Hsu
Department of Computer Science and Information Engineering,
National Taiwan University
Taipei 106, Taiwan
{r7526001, cchsu}@csie.ntu.edu.tw

Abstract

The original Genetic Evolved Classifier (GEC) [12] that uses a population-based approach has been proven to be effective on evolving a set of rules working together to solve classification problems. Based on the partitioning method in GEC, we include the proposed micro partitioning mechanism to calibrate the resolution of the partitions. For classes that lie closely together, increasing resolution can help distinguish members within these classes. To compensate the increase of search space complexity after the increase in resolution, we incorporate generalization and specialization mutation operator in attempt to speed up the evolution process and improve the classification rate. By using the generalization and specialization mutation operators and the micro partitioning method together, we could achieve higher classification rates. This is the Extended GEC (EGEC). Experiment results show that EGEC model is superior to GEC. With the new operators being effective, EGEC is also adequate in handling classification tasks. Besides having better performance than GEC, EGEC is also a general framework since it is based on GEC.

1. Introduction

Classification of data in to classes is one of the major tasks in data mining, i.e., bank loaning applications can be classified into either 'accept' or 'reject' classes. A classifier provides functions that map/classifies a data item/instance into one of the several predefined classes [7]. The automatic induction of classifiers from data provides both a classifier that can be used to map new instances to their classes and a human characterization of the classes.

Genetic algorithms [9] have been used successfully in a variety of search and optimization problems. Two general approaches of genetic algorithm-based learning have been used. The Pittsburg approach [15] uses a traditional genetic algorithm in which each entity in the population is a set of rules representing a complete solution to the learning problem. The Michigan approach [10] has generally used a distinctly different evolutionary mechanism in which the population consists of individual rules, each of which representing a partial solution to the overall learning task.

Providing a mechanism to convert data representation into chromosome representation, GEC [12] is capable of handling any type of classification problems. GEC is an evolutionary approach that uses genetic algorithm to evolve classification rules. It is a successful step forward in pioneering the possibilities of using soft computing in data mining tasks.

In this work, we focus on applying generalization and specialization mutation operators and the micro partitioning technique into GEC. Based on the results of GEC, we include the proposed micro partitioning mechanism to increase the resolution of the partitions. For classes that lie closely together, increasing resolution can help distinguish members within these classes. With the increase of resolution that increases the search space, we incorporate generalization and specialization mutation operator in attempt to speed up the evolution process and improve the classification rate. We shall call this new model the Extended Genetic Evolved Classifier (EGEC).

2. Micro Partitioning of Continuous Attributes

There should be no problem enumerating each possible value for categorical data one by one. We use a single bit for each possible value of an attribute to represent it as in GABIL [5]. For numerical attributes that are continuous over a range, directly enumerating each one of the possible values is impossible and impractical. We use the approach proposed in GEC [12], with modification to apply micro partitioning. We also obtain the following values:

N_{max}: The maximum value of the numerical attribute.
N_{min}: The minimum value of the numerical attribute.
R: The range of the numerical attribute, i.e., N_{max} - N_{min}
σ: The standard deviation of the gathered numerical values.

μ: The mean of the gathered numerical values

δ: Parameter tuning the partition size.

$$...,\left(\mu-\frac{3}{2}\frac{\rho}{\delta},\mu-\frac{1}{2}\frac{\rho}{\delta}\right),\left(\mu-\frac{1}{2}\frac{\rho}{\delta},\mu+\frac{1}{2}\frac{\rho}{\delta}\right),\left(\mu+\frac{1}{2}\frac{\rho}{\delta},\mu+\frac{3}{2}\frac{\rho}{\delta}\right),... \quad (1)$$

$$\delta\left\lfloor\frac{R}{\rho}\right\rfloor+1 \quad (2)$$

Partition is done with the mean μ as the center and the standard deviation σ/δ as an interval. The new discrete intervals generated will look like (1) when assuming δ, and the total number of partition generated will be estimated to (2). N_{max} will lie in the last interval and N_{min} will lie in the first interval. This partitioning method is done under the assumption that many natural phenomena carry the normal distribution property.

During the classifying phase, if the numerical data lies out of the range, i.e., the R we acquired during the training phase, we consider it as an outlier and ignore it (this is possible because the training set may not contain the whole sampling range).

The parameter δ decides the size (resolution) of each partition. The larger δ is, the smaller each partition will be. This approach is the micro partitioning mechanism. This is required for some data in which large partition size will not distinguish items of different class. Only by cutting the partitions into smaller pieces can then the items be decided. The tradeoff of this action is the complexity of search space. Take Figure 1 as an example; there are three classes to distinguish from: circle, triangle and star. For the circle class, using this partitioning resolution is adequate, but for the triangle and star class, the partitioning resolution is not enough. We must increase the resolution, i.e., the parameter δ. Shown in Figure 2 is the example of increasing δ by 2 times. Now we can see clearly that circle takes partitions 1 and 2, triangle takes partition 3 and star takes partition 4 and 5.

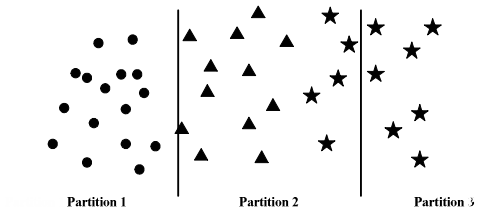


Figure 1. A 3 class example with partitioning inadequate resolution

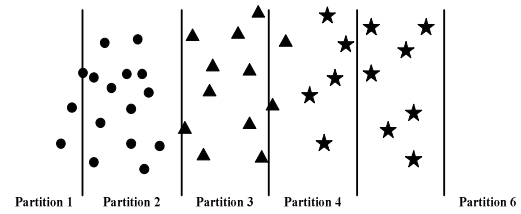


Figure 2. Increasing the partitioning resolution of the example in Figure 1

3. The Extended Genetic Evolved Classifier

The outline of our EGEC is shown in Figure 3. We use a divide and conquer approach. Rules are evolved for each class separately and separate genetic algorithms (GAs) are executed in hope to discover rules to cover the whole domain. The outline of the GA evolution is shown in Figure 4. It is based GEC [12] with modifications of adding generalization and specialization mutation operators.

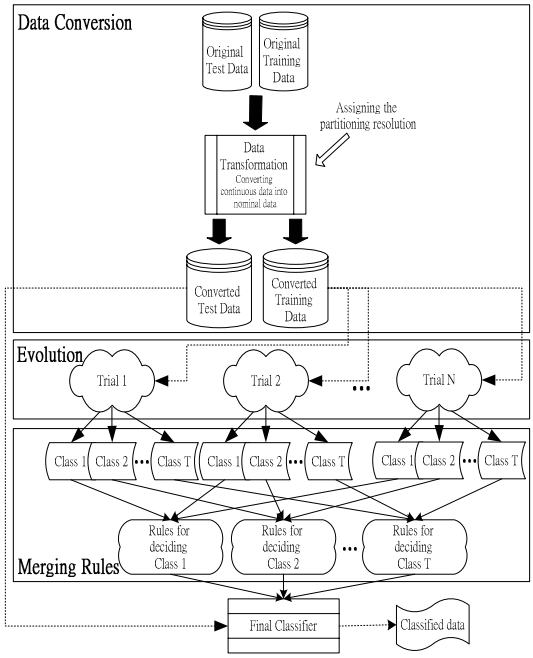


Figure 3. Outline of the EGEC

3.1. Genetic Algorithm Operations

The fitness evaluation method used here is the same as the one used in GEC [12]. The concept in here is that the more data match, the less data mismatched, and the more data covered, the higher the fitness value will be.

As usual, we do not wish the rule to be biased in any direction or to be dominated by a certain chromosome prematurely. Although uniform selection is simple and straightforward, it does provide a fair chance for every member to develop and evolve. Thus we chose to use uniform selection in EGEC.

The crossover mechanism we used here is single point crossover. For mutation, there are 3 types of mutation. The first type is simple mutation. For each bit in the chromosome, the mutation rate is set to 0.5. Taking such a high mutation rate will result in a more diverse set of rules to exploit. The other 2 mutation mechanisms are the generalized and specialized mutation, which will be described in the next subsection.

```
Procedure Evolve_Rule()
{
  Initialize population to size N for each class X;
  For each distinct class X
  {
    For # of Generations
    {
      For # of populations
      {
        Choose two distinct members A, B from class X as parent;
        Single point crossover A, B producing 2 new offsprings R, S;
        Choose a random member C;
        Mutate(C) producing T;
        For R, S, T Do
        {
          If( Fitness >= Average fitness of the population )
            Generalize();
          Else
            Specialize();
        }
        Add R, S, T to the new member pool;
      }
      Merge the original population with the new member pool;
      /* size of the population 4N */
      Select the best N member as the basis for the next generation;
    }
    Output the population of N members;
    /* This produces N classification rules for class X */
  }
}
```

Figure 4. Outline of the genetic algorithm for evolving classification rules

3.2. Generalization and Specialization Mutation

Two new operators have been introduced into the EGEC: generalization and specialization mutation. These operators either expand the coverage of a rule or narrow the contents of a rule. Our heuristic is as follows: If the fitness value of a rule, deciding class *X*, produced from crossover or mutation is below the average fitness of the set of rules deciding class *X*, then this rule will execute an generalize step, increasing the coverage in hoping of increasing the fitness. Otherwise, it will execute a specialize step.

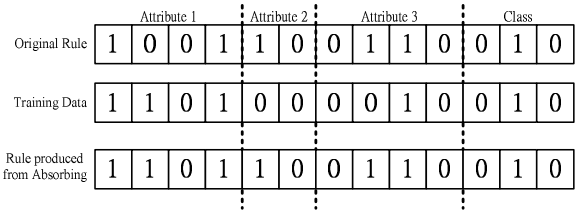


Figure 5. The generalization mutation in EGEC

During the generalization step, the rule randomly chooses 1 training case from the training data set which it is suppose to decide but did not and add it to itself, i.e., making itself deciding the rule. This is shown in Figure 5. Assume the rule is to decide if a data belongs to class 2. The training data shown is not decided because the bit position 2 in attribute 1 does not match the rule (remember that the rule is actually a if-then rule in CNF). The original rule then absorbs this piece of data because it also belong to class 2 and the result is setting bit position 2 in attribute 1 of the original rule to 1. It is a simple bitwise OR operation.

The specialization step is similar to the generalization step in the reverse way. The rule randomly chooses a training data that it decides but which it should not and extracts it out of itself, i.e., decreasing the number of false votes in the final decision of the whole EGEC. This process is shown in Figure 6. Using the same rule which decides class 2, it matches the training data and says that this piece of data belongs to class 2. But the accurate class of the training data is suppose to be class 1. The rule casts a false vote. It now spits this data out of itself by executing a bitwise XOR operation. The final rule produced after spitting will not match the training data then.

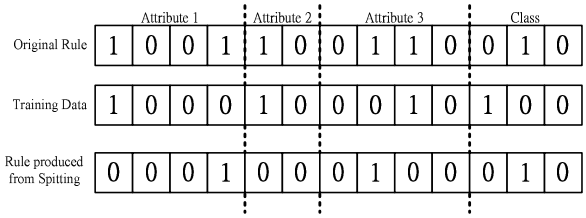


Figure 6. The specialization mutation in EGEC

One special case of the generalization mutation operation is that when there is no rules to generalize, i.e., the coverage of the rule covers every correct data belong to a specific class, it executes a specialization step then. The attempt here is to remove the parts of the rule that misclassifies data.

4. Experiment Results

4.1. Parameter Settings

For an instance containing n attributes with a_i representing the number of possible values for attribute I (continuous attributes are converted into discrete partitions by now), if it is to be classified into *x* classes, then the length of the chromosome is (4).

$$x + \sum_{i=1}^{n} a_i \tag{4}$$

The total number of possible rules is (5). Although some of the rules are contained within others (compression of rules mentioned in Section 2), this provides a rough estimate on how large our search space is. In this formula, δ represents the size of each partition, *A'* represents number of attributes which are continuous.

$$\delta^{A'} 2^{x+\sum_{i=1}^{n} a_i} \tag{5}$$

To keep our population size as small as possible to speed up the search process. We have used the following parameters in our experiment:

1. Population size is set equivalent to the number

of attributes, i.e., for the adult census database, there are 15 attributes (including the 'class') and so the population size is set to 15.

2. Maximum number of generations per trial is set to 100. The sampling of the result is done every 10 generations.
3. Crossover and mutation is always done with uniform selection described above.
4. Experiments were done by setting δ to 1, 4, 16 and 32.

Experiments are carried out on 6 databases and are to be compared with past results: they are the adult census database from [13], yeast classification database from [11], iris and wine database from [3]. Results are averaged from 20 independent executions. The legend of the following Figures 7-12 represents the partition size taken, i.e., the δ in formula (1). The x-axis of these tables represents the number of trials combined to form the whole classifier and the y-axis represents the accuracy rate.

Each test case receives a vote from each rule saying that in which class the test case belongs to. The final decision is made by using majority voting, i.e., deciding in which class the test case has the most votes. In case of a tie, we consider this test case as undecidable.

4.2. Experiment Results

Experiment has been conducted on 6 different databases. These data has been acquired from the UCI (University of California at Irvine) – Machine Learning Repository [1]. The properties of these databases have been listed in Table 1. By this simple listing, we can show that our EGEC can handle not only continuous attributes, but also multi-class classification tasks. Besides, it can handle classification tasks containing purely of continuous/nominal attributes or a mixture of both types of attributes. In order to make comparisons, three-fold cross validation is used (except for the adult census, which the training and testing set is already provided). Each trail uses a different training and testing set, i.e., a new three-fold set is produced each time.

Table 1. Summarization of the databases used for our experiment

Database Name	Nomial Attributes	Continuous Attributes	Instances	Number of Classes
Adult	8	6	48842	2
Yeast	0	8	1484	10
Wine	0	13	178	3
Iris	0	4	150	3
Dematology	33	1	366	6
Breast Cancer	9	0	286	2

We can see in Figures 7, 8, 10 and 11 that as we decreased the partition size, we get better results. Taking Figure 7 (the adult census database) as an example, we can see that when δ changes from 1 to 4 or from 4 to 16, the result curve changes dramatically. EGEC using micro partitioning technique is able to classify more data correctly leading to an increase of accuracy. This proves that when we shrink the partition size, some classes originally lying in the same unshrunked partition can be split apart.

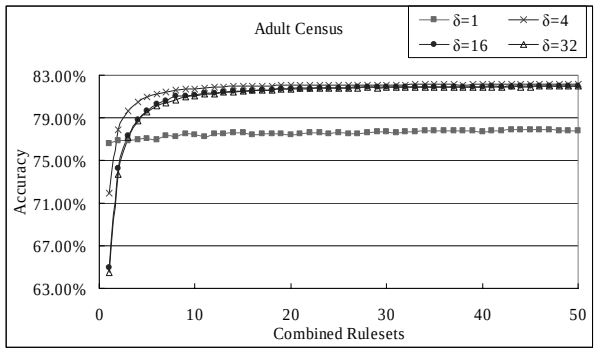


Figure 7. Result of EGEC on the adult census database

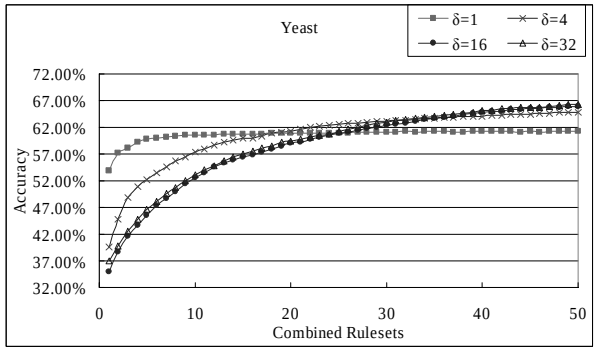


Figure 8. Result of EGEC on the yeast database

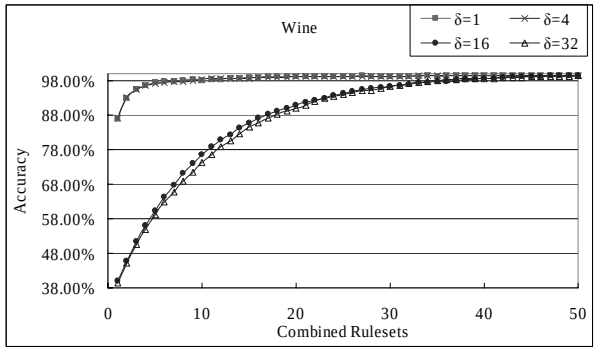


Figure 9. Result of EGEC on the wine database

For Figures 9 and 11, especially Figure 9, we did not see too much improvement on average (the curves at the end nearly coincide), but using the micro partitioning mechanism, we were able to achieve perfection in some occasions during the experiment trials, i.e., 100%

classification rate. For Figure 11 (the breast cancer database), there is no change. This is because it contains no continuous attributes and thus no matter what δ is, the number of partitions for it is always the same. This is due to that micro partitioning is for continuous attributes only. To increase the accuracy of this test case, using other partitioning techniques is required.

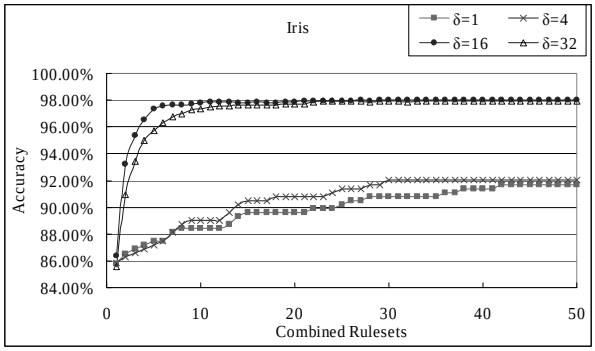


Figure 10. Result of EGEC on the iris database

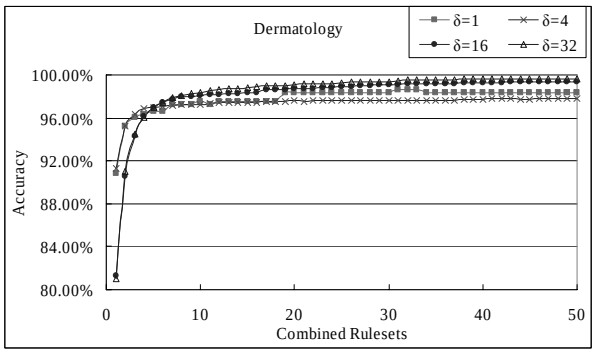


Figure 11. Result of EGEC on the dermatology database

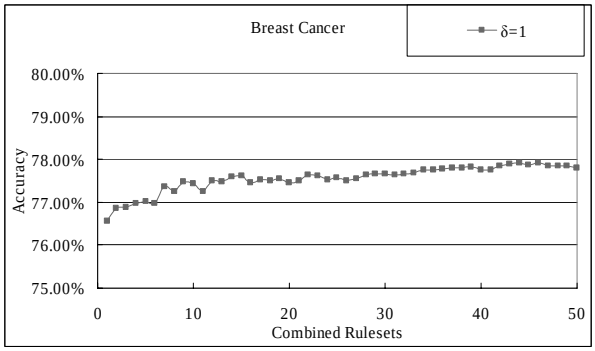


Figure 12. Result of EGEC on the breast cancer database

Generally, as we increase the number of generations, decrease the size of partitions and the number of rulesets to combined, i.e., the number of trials executed in total, the capability of EGEC increases. This phenomenon, as expected, is seen in all of the results. Comparisons of the EGEC to other methods are listed in Table 2. Abbreviations used are as follows:

- NB: Naïve-Bayesian classifiers.
- APM: The Ad Hoc Structured Probability Model. Experiment results are directly obtained from Horton [11].
- F-ID3: Fuzzy ID3, decision tree method. Results directly obtained from Chen [3]. F-ID3(Best) represents the best result obtained using the F-ID3.
- Fidel.: A GA based method which evolves IF-THEN comprehensible classification rules proposed by Fidelis [8].
- Cest.: A knowledge-elicitation tool proposed by Cestnik [2].
- GEC: The Genetic Evolved Classifier model by Hsu [12]. GEC(Best) represents the best result obtained using GEC.
- EGEC: Our method proposed in this work.

Table 2. Comparasion of various works

	Adult	Yeast	Wine	Iris	Dermat-ology	Breast Cancer
C4.5	84.86%	N/A	94.50%	95.00%	N/A	N/A
NB	83.88%	N/A	N/A	N/A	N/A	N/A
APM	N/A	55.00%	N/A	N/A	N/A	N/A
F-ID3	N/A	N/A	92.30%	96.00%	N/A	N/A
F-ID3 (Best)	N/A	N/A	96.50%	98.00%	N/A	N/A
Fidel.	N/A	N/A	N/A	N/A	95.00%	67.00%
Cest.	N/A	N/A	N/A	N/A	N/A	78.00%
GEC	81.60%	61.79%	94.69%	92.00%	N/A	N/A
GEC (Best)	82.30%	62.94%	97.17%	92.00%	N/A	N/A
EGEC	81.99%	66.32%	99.41%	98.00%	99.65%	77.80%
EGEC (Best)	82.33%	69.61%	100.0%	98.67%	100.0%	80.51%

We can see that EGEC performed well in all of the databases listed. For the adult census database, EGEC and GEC performed nearly the same. Both EGEC and GEC are quite comparable with C4.5 and NB. For the yeast database, EGEC outperformed GEC 4.53% on average. In the best-case analysis, EGEC outperformed APM and GEC by 6.67% and 14.61% respectively. EGEC is excellent on the iris, wine and dermatology database; average cases are 99.41%, 98% and 99.65% respectively. Considering the best case, EGEC has 100% accuracy appearing on the wine and dermatology databases. On the wine database, EGEC outperformed C4.5, F-ID3 and GEC in the wine database by 5.5%, 3.5% and 2.83% respectively. Looking at the iris database, EGEC outperformed C4.5, F-ID3, and GEC by 3.67%, 0.67% an 6.67% respectively. Finally yet importantly, on the breast cancer database, EGEC has major performance of 13.51% increase compared to Fidelis [8] 13.51% and a

small increase of 2.51% compared to Cestnik [2]. In general, the performance of EGEC is acceptable when compared to previous works done.

5. Conclusion

EGEC is a model based on GEC, which each rule within is considered as an single entity and joins up to form a metabiosis body. This is just as how simple organisms join to form a colony. In a microscopic point of view, there may be conflicts within rules, but from the macro view the whole body reports an consistent result. This decision is done by using majority voting.

Like the GEC, EGEC is a general framework. Besides being able to handle multiclass classification tasks that is confirmed in [12], the newly introduced generalization and specialization mutation operators has increased the power of GEC into EGEC. By using generation, the evolution process can be speeded up towards an objective. By using specialization, helping the search process to jump out of local extremes is possible because undesired results can be removed.

Change the partition size can affect classification accuracy if continuous attributes are present. This is an exchange between time and accuracy. By using small partitions, the search space increases exponentially. A compromise between time and accuracy must be made here.

A trivial phenomenon of the EGEC model is that it produces many rules. Although a large proportion of these rules are the same in some way (identical of subsets of one another), experiments done to purge them leads to a worse performance. This is because the whole classifier is now an independent body containing many rules in it. Although we can reverse the encoding of the chromosome into a human readable IF-THEN rule format, we do not know how each of the rules interact within the whole body. Since this relation is unknown, we may not remove rules from the body. Further research on the refinement of these rules is required.

6. References

[1] C. L. Blake and C. J. Merz. UCI Repository of machine learning databases Irvine, CA: University of California, Department of Information and Computer Science, 1998. [http://www.ics.uci.edu/~mlearn/ MLRepository.html]

[2] G. Cestnik, I. Konenenko and I. Bratko, “Assistant-86: A Knowledge-Elicitation Tool for Sophisticated Users,“ *Progress in Machine Learning*, pp. 31-45, 1987.

[3] H. M. Chen and S. Y. Ho, “Designing an Optimal Evolutionary Fuzzy Decision Tree for Data Mining,” *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 943-950, 2001.

[4] P. Clark and T. Niblett, “Introduction in Noisy Domains,” *Progress in Machine Learning* (from the Proceedings of the 2nd European Working Session on Learning), pp. 11-30, 1987.

[5] K. A. De Jong, W. M. Spears, D. F. Gordon, “Using Genetic Algorithms for Concept Learning,” *Machine Learning*, vol. 13, no. 2, pp. 161-188, 1993.

[6] G. Demiroz, H. A. Govenir and N. Ilter, “Learning differential diagnosis of erythemato-squamous disease using voting feature,” *Artificial Intelligence in Medicine*, v. 13, pp. 147-165, 1998.

[7] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, “From data mining to knowledge discovery: An overview,” *Advances in Knowledge Discovery and Data Mining*, chap. 1, pp. 1-34, AAAI Press and MIT Press, 1996.

[8] M. V. Fidelis, H. S. Lopes and A. A. Freitas, “Discovering Comprehensible Classification Rules with a Genetic Algorithm,” *Proceedings of the 2000 Congress on Evolutionary Computation*s, pp. 805-810, 2000.

[9] J. H. Holland, *Adaptation in Natural and Artificial Systems*, Univ. of Michigan Press (Ann Arbor), 1975.

[10] J. H. Holland, “Escaping brittleness: the possibilities of general-purpose learning algorithms applied to parallel rule-based systems,” *Machine Learning, an artificial intelligence approach*, 2, 1986.

[11] P. Horton and K. Nakai, “A Probablistic Classification System for Predicting the Cellular Localization Sites of Proteins,” *Intelligent Systems in Molecular Biology*, pp. 109-115, 1996.

[12] W. W. Hsu and C. C. Hsu, “GEC: An Evolutionary Approach for Evolving Classifiers,” *to appear in Proceedings of the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, Taipei, Taiwan, 2002.

[13] R. Kohavi, “Scaling Up the Accuracy of Naïve-Bayes Classifiers: a Decision-Tree Hybrid,” *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, pp. 202-207, 1996.

[14] C. H. Liu, C. C. Lu and W. P. Lee, “Document Categorization by Genetic Algorithms,” *IEEE International Conference on Systems, Man and Cybernetics (SMC)*, pp. 3868-3872, 2000.

[15] S. F. Smith, *A Learning System Based on Genetic Adaptive Algorithms*, PhD Thesis, Univ. of Pittsburgh, 1980.

A Data Mining Approach for Retailing Bank Customer Attrition Analysis

Xiaohua Hu
DMW Software, 504 E. Hillsdale Ct., San Mateo, California 94403
And
Dept. of Math and Computer Science, San Jose State University
San Jose, CA 95192
Email: xiaohua_hu@acm.org; tonyhu@mathcs.sjsu.edu

Abstract

In this paper, we present a data mining approach for analyzing retailing bank customer attrition. We discuss the challenging issues such as highly skewed data, time series data unrolling, leaker field detection etc, and the procedure of a data mining project for the attrition analysis for retailing bank. We explain the advantages of lift as s proper measure for attrition analysis and compare the lift of data mining models of decision tree, boosted naïve Bayesian network, selective Bayesian network, neural network and the ensemble of classifiers of the above methods. Some interesting findings are reported. Our research work demonstrates the effectiveness and efficiency of data mining in attrition analysis for retailing bank.

1. Introduction

In this paper our discuss on applying data mining techniques to help retailing banks for the attrition analysis. The goal of attrition analysis is to identify a group of customers who have a high probability to attrite, and then the company can conduct marketing campaigns to change the behavior in the desired direction (change their behavior, reduce the attrition rate). If the data mining model is good enough and target criteria are well defined, the company can contact a much small group of people with a high concentration of potential attriters [7]. The paper represents the initial findings report on the data mining phase for a retailing bank attrition analysis. The purpose is the identification of rules, trends, patterns and groups that can serve as potential indicators of attrition. These results, in conjunction with existing business, risk, profitability and segmentation data available form the basis for the future deployment of a retention unit. The paper is organized as follow: we first define the problem and formulation of business problems in the area of customer retention, data review and initial, then data gathering, cataloging and formatting, data unfolding and time-sensitive variable definition. Then we discuss sensitivity analysis, feature selection and leaker detection. Next we describe data modeling via decision trees, neural networks, Bayesian networks, selective Bayesian network and an ensemble of classifier with the above four methods. Finally we conclude with our findings and next steps

2. Business problem

Our client is one of the leading retailing banks in the US. It offers many type of financial retail products to various customers. The product we discussed in this paper belongs to certain type of loan service. Over 750,000 customers currently use this service with $1.5 billion in outstanding, the product has had significant losses. Revenue is constantly challenged by a high attrition rate: every month, the call centers receive over 4500 calls from customers wishing to close their accounts. This, in addition to approximately 1,200 write-ins, "slow" attriters (no balance shown over 12 consecutive months) and pirated accounts constitutes a serious challenge to the profitability of the product, which totals about 5,700/month mostly due to rate, credit line, and fees. In addition to that, many customers will use the product as long as the introductory or "teaser" rate (currently at 4.9%) is in effect and lapse thereafter. Currently, our client doesn't have a proactive or reactive retention effort. However, the situation described above has motivated the business and technology executives of our client to review the possibility of setting a knowledge based retention effort through a combination of effective segmentation, customer profiling, data mining and credit scoring that can retain more customers, while maximizing revenue.

There are different types of attriters in the product line:

- Slow attriters: Customers who slowly pay down their outstanding balance until they become inactive. Attrition here is understood comprehensively, where voluntary attrition can show more than one behavior.
- Fast attriters: Customers who quickly pay down their balance and either lapse it or close it via phone call or write in.

- Pirating: Identify customers likely to transfer their relationship to competing products and away from our client.

We decided to concentrate two attrition problems

1. Utilizing data on accounts that remained continuously open in the last 4 months, predict, with 60 days advance notice, the likelihood that a particular customer will opt to voluntarily close his/her account either by phone or write-in.
2. Utilizing data on accounts that remained continuously open in the last 4 months, predict, with 60 days advance notice, the likelihood that a particular customer will have his account transferred to a competing institution. The account may or may not remain open.

The focus of the modeling process, and subsequent campaigns, will revolve around the resolution of the two classes of business problems related to improving customer retention and activation for the product line as identified by the business:.

2.3 Data Selection

Like in all data mining exercises, the identification of relevant data in the right quantity and over a significant period of time is critical for the development of meaningful models. Given this and worked with the domain expert, we proceeded to identify the necessary data sources available and those readily accessible for initial review:

(1) DDS Warehouse: Credit Card Data Warehouse containing about 200 product specific fields. Originating at various points. The data is compacted according to a set of operational rules that reduce size for non-changing fields. The Warehouse contains 6 months of data and is rotated on a monthly basis. In some cases, additional attributes allow for data to cover up to 18 months. For the current exercise, the period includes 4 month history information

(2) Third Party Data: A set of account related demographic and credit bureau information. The data is available from an external provider.

(3) Segmentation files: Set of account related segmentation values based on our client's segmentation scheme which combines Risk, Profitability and External potential

2.4 Data Preprocessing Goals

The data preprocessing state consists of the series of activities necessary to create a compacted file that:

- Reflects data changes over time.
- Recognizes and removes statistically insignificant fields
- Defines and introduces the "target" field
- Allows for second stage preprocessing and statistical analysis.

This was accomplished through three steps, detailed in the sections below:

- Time series "unrolling"
- Target value definition
- First stage statistical analysis

2.4.1 Time Series "Unrolling"

In our application, historical customers records are used to group customers into two classes – those who are attriters and those who are not. In order to save space, every month a query checks every field against the previous month. If there is no change, no rows are added and the value of Effective Start Date (EFF_START_DT) remains as that during which a change was last recorded (which is the same as "a new row was inserted"). If any attribute changes, a whole new row is added with the corresponding EFF_START_DT updated. Therefore, it is very likely that some of the accounts will have less than the corresponding number of months in cases where no activity is recorded. For example, if an account has had no activity since December '2001, the last row will be the one for that month and it is up to the user to extrapolate it all the way to the current month. In this example, it would mean that the particular customer has not used his account in 4 months. If we wanted to understand the activity for the last 16 months we would have to add the number of zero corresponding to the last 4 months and merge them with those for the previous 12. Understanding this when arranging the files is critical to developing the attrition model.

The data format used required for the implicit data to be made explicit and the time periods to be itemized into individual fields. To accomplish this, the time sensitive variables were assigned a time prefix. So, for example, the variable *CURRENT_BALANCE* for the period of December 2001 to March 2002 is redefined as:

Table 1: Naming Convention for Time Sensitive DDS Data for the 4 months Period

Period	Nomenclature
Current Month (March 2002)	T0_CURRENT_BALANCE
One Month Back (Feb 2002)	T1_CURRENT_BALANCE
Two Month Back (Jan 2002)	T2_CURRENT_BALANCE
Three Month Back (Dec 2001)	T3_CURRENT_BALANCE

.

2.4.2 Target value definition

Like many real data mining applications, normally there is no data mining target field defined directly in the data warehouse. It is part of the data mining procedure to define the proper target field based on the business objective for the data mining analysis. With the help of the business domain experts, we define the target value in terms of existing data and, with these, define the value of the target variable, i.e., the variable that determines the voluntary attriters, hereby defined as **VA_ACCTS**. It is defined in terms of:

1. Status code (*NON_CRD_ST_CD*)
2. Status change date (*NON_CRD_STATUS_CHANGE_DATE*)
3. Closed reason code (*NON_CRD_CLS_REA_CD*)

2.4.3 First stage statistical analysis

The statistical analysis, the first in a series, is done in order to obtain an initial understanding of the data quality: number of unknown fields, relative frequency, early indicators, averages and target data distribution. As an initial field discrimination step, the fields where a single value appeared in more than 99.8% of all records was deemed statistically insignificant and removed from the set of attributes. These fields are removed from both the data and metadata files to ensure their removal from the modeling process, thus reducing the computing time required.

2.5 Data premodeling

The data premodeling stage is the next critical step in the generation of the files used for modeling. This stage consists of three main steps, namely: (1) field sensitivity analysis to filter fields with low correlation to target the field and detect data *"leakers",* (2) field reduction to create a compacted file with highly relevant fields, (3) file set generation of all balanced and unbalanced sets required for training, testing and iterative verification of results and model refinement.

2.5.1 Field Sensitivity Analysis

The field sensitivity analysis is used to determine each attribute's "contribution" to the modeling process. Using a customized program, each field can be used to predict the target value in order to determine its impact on the predicted value. When the relative value is low, the field can conceivably be removed from the set. On the other hand, a field whose accuracy is very high, it is considered to be a potential *leaker*. Leakers are fields that "leak" information on the target. For example, a field with a value representing account closure could leak information on attrition, and would confound modeling efforts.

While some leakers are readily explained, many times they are included in business rules whose relation to the target is not apparent. In this case, the best way to determine if a field is indeed a leaker is to discuss the findings with those familiar with the data schema and the business problem. In many circumstances, field names and values are not always representative of their function, and need clarification. One the other hand, fields that are suspected but turn out *not* to be leakers constitute potential predictors in the model.

2.5.2 Field Reduction

Using our homegrown feature selection component, results from the field sensitivity analysis can be used to discard fields that provide very little contribution to the prediction of the target field. Contribution is defined by the accuracy of the single field prediction. A threshold accuracy of 45% was used to discard fields (i.e.: fields with a predicted error rate greater than 45% were discarded). In some cases, the values for a field are constant (i.e.: have a standard deviation of zero) and thus have no predictive value. These fields should be removed in order to improve data mining processing speed and to generate better models. For example, through this effort, the initial set of 309 attributes in the data set was reduced to 142 after processing.

2.5.3 Files Set Generation

Our sample file comprises of 468000 records, based on the historical data of the recent 4 months, the attrition rate is around 2%. In order to build a good model from this highly skewed data set, we need to build a more balanced representation of attriters and non-attriters in the training data set. The reason is that in the original data file, we have high non-attriters percentage (98%) vs. a very low attriter rate (2%); a learning model can achieve high accuracy by always predicting every customer to be a non-attriters. Obviously, such a high accurate model is useless for our attrition analysis. We created a random sample file where we include about 938 attriters and then we add enough non-attriters into it to make it a dataset with 50-50 percentage of each class category (attriters vs non-attriters), then file was divided into *balanced, train* and *test* files as well as *raw* (i.e., unbalanced*) test* and *held aside* files for verification purpose. The *balanced train file* consisted of 50% of the records containing target values, i.e., for whom VA_ACCTs=1. The *balanced test, raw test*, and *raw held aside files* consisted of approximately 1/6 of the targets each. As defined earlier in Section 2.4.2, targets in the raw files represent 2% of the total number of records for the files being reviewed. These files were handed over to the data mining component for further statistical analysis, data mining and clustering work.

3. Model Development Process

The goal of the attrition analysis is not to predict the behavior of every customer, but find a good subset of customers where the percentage of attriter is high. As pointed in [5,6,7], prediction accuracy, which was used to evaluate the machine learning algorithm, cannot be used as a suitable evaluation criterion for the data mining application such as attrition analysis. The main reason is that classification errors (false negative, false positive) must be dealt with differently. So it is required that learning algorithms need to classify with a confidence measurement, such as a probability estimation factor or certainty factor (also called scores in attrition analysis). The scores will allow us to rank customers for promotion or targeting marketing. Lift instead of the predictive accuracy is used as an evaluation criterion. As pointed in [5], if the data mining model is good enough, we should find a high concentration of attriters at the top of the list and this higher proportion of attriters can be measured in terms of "lift" to see how much better than random the model-based targeting is. Generally, lift can be calculated by looking at the cumulative targets captured up to p% as a percentage of all targets and dividing by p% [6]. For example, the top 10% of the sorted list may contain 35% of likely attriters, then the model has a lift of 35/10=3.5.

Lift measures the increased accuracy for a target subset based on a model-scored ranked list. Using past information collected over several months on usage of the financial service, our task is to build a model for predicting the customer class in the next two months and apply it to the whole customers. The prediction model is used to rank the customers based on their likelihood of attrition. As shown in section, the attrition rate for our clients is low (2%) and it is difficult or impossible to predict with high accuracy for all customers, and usually it is not necessary to predict all the customers because in practice, for attrition analysis, it is a good practice to contact a small percentage of customers and hope this small percentage of customers contains a high concentrated percentage of attriters than random sample. We are interested in models that maximize lift. A good model in our analysis should concentrate the likely attriters near the top in the sorted list based on the attrition scores generated by the model. We need to use learning algorithms that can produce scores in order to rank the testing examples. Algorithms such as Naïve Bayesian, decision tree, neural network satisfy our requirement. We performed several data mining analyses using four different data mining algorithms and an ensemble of classifiers. These are:

1. Boosted Naïve Bayesian (BNB)
2. NeuralWare Predict (a commercial neural network from NeuralWare Inc)
3. Decision Tree (based on C4.5 with some modification)
4. Selective Naïve Bayesian (SNB).
5. An ensemble of classifier of the above four methods

3.1 Bootstrapped Naïve Bayesian Networks

The BNB data mining method combines boosting and naive Bayesian learning [2]. Boosting is a general method of improving the predictive accuracy of any two-class learning algorithm, which works in successive stages. In the first stage, all the training examples are weighted equally and the two-class learning algorithm is used to acquire a classifier. In the second stage, the examples that are misclassified by this first classifier are upweighted, and a second classifier is learned that focuses on these examples. In the third stage, the examples misclassified by the second classifier are upweighted, and a third classifier is learned. The boosting process can be repeated for as many stages as desired. Applied with naive Bayesian learning, generally five to twenty stages are beneficial. The results described here use just five stages.

Like other software, our BNB software identifies which attributes are most predictive of an example being a target. Unlike most other software, BNB reports which values (or numerical ranges) of an attribute are most predictive. For example, BNB automatically identifies that the value 2 of the attribute T1_NON_CRD_ACCOUNT_FORMAT is an important predictor. According to the supplied documentation, this value 2 signifies "account which has been active but is currently not active." Also unlike other software, BNB evaluates the statistical significance of the predictors that it reports. The significance of a predictor depends on both its lift (i.e. predictive benefit) and of its coverage (i.e. number of examples to which it applies). BNB does not report predictors that may be spurious, because they have low coverage or low lift.

Pct	cases	Hits boosted BN	% hits	lift	Hits no model
1	70	3	4.3%	1.9	1.5
4	283	24	8.5%	3.9	6.2
8	567	51	9.0%	4.1	12.5
9	638	55	8.6%	3.9	14.0
10	709	62	8.7%	4.0	15.6
15	1063	71	6.7%	3.0	23.4
20	1418	78	5.5%	2.5	31.2
25	1772	93	5.2%	2.4	39.0

3.2 Decision Trees

Decision tree methods build a collection of rules for use as a predictive model [9]. The advantage of this approach is that the rules are easy to understand, and they are frequently useful for discovering underlying business processes. The disadvantage of decision tree approaches is that these models usually do not perform as well as other models. We have developed a proprietary modification for standard decision tree algorithms for use in "lift" problems where, for example, we want to minimize performance in the top 25% of the predicted data (and care less about performance elsewhere). This is the situation for common problems, such as attrition and targeted mailings.

PCT	lines	Hits decision tree	% hits	lift	Hits no model
1	70	6	8.6%	3.9%	1.5
4	283	25	8.8%	4.0%	6.2
8	567	47	8.3%	3.8%	12.5
9	638	56	8.8%	4.0%	14.0
10	709	60	8.5%	3.8%	15.6
20	1418	95	6.7%	3.0%	31.2
25	1772	101	5.7%	2.6%	39.0

3.3 Neural Networks

Neural networks are a well-established approach for modeling data. The advantage of this approach is that neural network models tend to be among the most predictive models. The disadvantage of neural network models is that it can be harder to understand their output. For our work we have used a commercial package (NeuralWare Predict)

PCT	Cases	Hits Neural Net	% hits	lift	Hits no model
1	70	9	12.9%	5.8	1.5
4	283	41	14.5%	6.6	6.2
8	567	48	8.5%	3.8	12.5
9	638	48	7.5%	3.4	14.0
10	709	53	7.5%	3.4	15.6
15	1063	73	6.9%	3.1	23.4
20	1418	86	6.1%	2.8	31.2
25	1772	105	5.9%	2.7	39.0

3.4 Selective Naïve Bayesian Networks

The naive Bayesian classifier is a probabilistic, predictive model that assumes that all attributes are conditionally independent of each other given the target variable i.e. within each class, the attributes are unrelated. The naïve Bayesian classifier is simple, inherently robust with respect to noise, and scales well to domains that involve many irrelevant features. Moreover, despite its simplicity and the strong assumption that attributes are independent within each class, it has been shown to give remarkably high accuracies in many natural domains. The selective naive Bayesian classifier that we used is an extension to the naive Bayesian classifier designed to perform better in domains with highly correlated (redundant) features. The intuition is that, if highly correlated features are not selected, the classifier should perform better given its feature independence assumptions. Attributes are selected by starting with an empty set of attributes, and then incrementally adding that single attribute (from the set of unselected attributes) the attribute that most improves the accuracy of the resultant classifier on the test set. Attributes are selected until the addition of any other attribute results in a fall in accuracy of the classifier.

PCT	Cases	Hits SelectiveBN	% hits	lift	Hits no model
1	70	5	7.1%	3.2	1.5
4	283	27	9.5%	4.3	6.2
8	567	53	9.3%	4.2	12.5
9	638	60	9.4%	4.3	14.0
10	709	69	9.7%	4.4	15.6
15	1063	83	7.8%	3.5	23.4
20	1418	92	6.5%	2.9	31.2
25	1772	105	5.9%	2.7	39.0

3.5 A hybrid approach: An ensemble of classifiers

An ensemble of classifiers is to generate a set of classifiers instead of one classifier for the classification of new object, hoping that the combination of answers of multiple classifiers result in better accuracy. Ensemble of classifiers has been proved to be a very effective way to improve classification accuracy because uncorrelated errors made by the individual classifier can be removed by voting. A classifier, which utilizes a single minimal set of classification rules to classify future examples, may lead to mistakes. An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way to classify new examples. Many methods for constructing ensembles of classifiers have been developed, some are general and some are specific to particular algorithms [3]. We adopted a hybrid approach: we first built 4 classifiers using Boosted Naïve Bayesian (BNB), NeuralWare predict, Decision Tree, Selective Naïve Bayesian (SNB), then we ensemble an classifier based on the majority vote of these 4 classifiers.

Pct	cases	Hits Ensemble of classifier	% hits	lift	Hits no model
1	70	4	5.7%	2.6	1.5
4	283	25	8.9%	4.0	6.2
8	567	52	9.3%	4.2	12.5
9	638	61	9.6%	4.4	14.0
10	709	63	8.9%	4.0	15.6
15	1063	81	7.7%	3.5	23.4
20	1418	96	6.5%	3.0	31.2
25	1772	104	5.9%	2.6	39.0

4. Data Mining Findings

To test the effectiveness of the data mining models, our client conducted a field test on their customers. The test wanted to show two points: (1) the top percentage of the customer attrition list does contain concentrated attriters, (2) the data mining based marketing approach is effective for retention purpose. They ran the model generated from the ensemble of classifiers approach on the current customers and then sorted the customers based on the attrition scores. They decided to contact the top 5% customers from the list, which has around 20000 customers. They divided the customers into 2 groups randomly, each with 10000 customers and took different proactive actions to each group: for group1, the marketing department contacted each customers and offered some incentive packages to encourage the customers to stay with the company, for group 2, there is no contact. After two months later, they examined the list and found out, for group 1, there attrition rate is very low (0.8%), for group two, the attrition rate is very high, almost 10.6%, while the average attrition rate is 2.2%, thus achieved a lift of 4.8 (consistent with the list of 4.6 in the test data set). The lower attrition rate among group 1 did indicate, if the proactive action is in time and proper, it does have an impact on the customers' behavior, the high attrition rate among group 2 demonstrate that our data mining model is accurate and the top 5% captured a high concentrated proportion of attriters.

5. Conclusion

In this paper, we present a data mining approach for retailing bank customer attrition analysis. We discuss the challenging issues such as highly skewed data, time series data unrolling, leaker field detection etc, and procedure of a data mining task for the attrition analysis for retailing bank. We discuss the use of lift as a proper measure for attrition analysis and compare the lift of data mining model of decision tree, boosted naïve Bayesian network, selective Bayesian network, neural network and the ensemble of class of the above methods. Our initial findings show some interesting results. Next step, based on above results and new source files available on segmentation, we will review the voluntary attrition trends on a segment-by-segment basis. A thorough clustering study is planned for the data to review the natural grouping of the data and how it lines up with the segmentation in terms of incidence, variables and number of groups.

6. References

[1] Bhattacharya, S. "Direct Marketing Response Models Using Genetic Algorithms", Proc. Of the 4th International Conference on Knowledge Discovery and Data Mining, pp144-148

[2] Elkan, C. Boosted and Naïve Bayesian Learning. Technical Report No. CS97-557, September 1997, UCSD.

[3] Hu, X., Using Rough Sets Theory and Database Operations to Construct a good Ensemble of Classifiers for Data Mining Application, Proc. of the 2001 IEEE International Conf. on Data Mining (IEEE ICDM2001)

[4] Hughes, A. M., The Complete database marketer: second-generation strategies and techniques

for tapping the power of your customer database. Chicago, IL: Irwin Professional

[5] Charles Ling, Chenghui Li, "Data Mining for Direct Marketing: Problem and Solutions", Proc. Of the 4th International Conference on Knowledge Discovery & Data Mining,

[6] Brij Masand, Gregoey Piatetsky-Shapiro, "A Comparison of Approaches for Maximizing Business Payoff of Prediction Models", Proc. Of the 2nd International Conference on Knowledge Discovery and Data Mining

[7] Gregoey Piatetsky-Shapiro, Brij Masand, " Estimating Campaign Benefits and Modeling Lift", Proc. Of the 5th SIGKDD International Conference on Knowledge Discovery and Data Mining, pp185-193

[8] Provost, F., and Fawcett, T., "Analysis and Visualization of Classifiers Performance: Comparison Under Imprecise Class and Cost Distribution", Prod. Of the 3rd International Conference on Knowledge Discovery and Data Mining, pp 43-48

[9] Quinlan, J.R, Induction of Decision Tree, Machine Learning, 1(1), 81-96

A Novel Approach of Forecasting Association Rules by Genetic Programming and Biochemical-based Synthesis

C.M. Hung, Y.M. Huang, T.S. Chen

Department of Engineering Science, National Cheng Kung University, Taiwan, ROC
hon@mail.tnb.com.tw, {Raymond, tsch}@mail.ncku.edu.tw

Abstract

To forecast association rules is a time-consuming work when the number of items becomes very huge and an exhaustive search is employed to build a learning model. In this paper, a heuristic model for improving the performance of the forecast is proposed by synthesizing active components and then utilizing grammar-base genetic programming (GP) making it evolutionary to eventually control their physical properties. In our research, these activity components are called as virtual lattice (VL) that satisfies partial definitions of lattice in discrete mathematics. The VL not only simulates a behavior of polymers but also acts as an individual in GP. Based on the physical properties of VL It is possible to find out the forecast directions or guide the future VLs for mining association rules. By selecting activity components, our algorithm can deal with the transactions of on-online database in multivariate time series divided into many segments. The segments of transactions can induce large itemsets by Apriori algorithm. These large itemsets form some streams with different quantities and are treated as the initial population of GP. The fitness function is for finding out the best large itemsets that are defined as some itemsets owning to the longest time laxity between two large itemset streams (LISs). Our analytic results indicate that the proposed algorithm is better than the Apriori algorithm in terms of time complexity, although it faces losing some accuracy due to using a heuristic model. This research will become very interesting on solving some time-consuming problems according to biochemistry theorem in the future.

Key words: data mining, association rules, genetic programming, lattice, incremental

1. Introduction

1.1. Problem

Recently, the techniques of data mining have been widely applied to new applications of enormous database. Many studies have dedicated to improve the performance and accuracy of data mining continuously. Meanwhile they have tried to make a new application of data mining, especially on the interested and comprehensive models.

Suppose a merchant promotes some combination of sales items. He hopes to know what kind of combinations having more opportunity to sale. In this case, it is very easy to obtain such rules by a veteran. It is worthless to spend too much computational resource for finding some trivial rules on a huge database by data mining. Furthermore, The mined rules may become out-of-date in Multivariate Time Series (MTS) [2]. In fact, the final purpose of data mining is how to improve the operational environment for a salesman, then to make a gain by competing against the other trades. The major objective of our research is to discover a heuristic and sound method to solve both high dimension and NP-complete problems.

There are three requirements as follows. 1) Avoiding to consume considerable quantities but to find a lot of low interested rules. 2) The prediction of finding large itemset in association rules is still workable as well during multivariate time series. It may speedup search large itemset by a known direction of distribution. 3) How to decide a pattern of the parameter to improve the performance under the reasonable cost. If the assumption of which a material bounds to own properties itself is satisfied, then the physical properties of a material bound to be predicted, vice versa. We now employ the concept of one virtual lattice (VL) to represent a basis of those materials of own properties itself. In this study, a VL is an overlapping representation for a group of large item sets. In other words, it is a truly feature of a dynamic database for mining association rules.

1.2. Association rules

Within the area of data mining, the problem of deriving associations from data has recently received a great deal of attention. In general, the algorithm of searching a set of association rules decreases exponentially its performance on processing an enormous datasets of which have too many transactions, items, and large itemsets. In particular, it is so complex and becomes infeasible after involving the factor of time. About the scope of association rules, there were many improvements on the basis of Apriori algorithm in most literatures. The main procedure is to count a large itemset L_{k-1} and to join their L_{k-1}* L_{k-1} in order to generate next candidate itemset C_k, where k is the size of a large itemset. It seems easy to implement but

suffer a low performance if the k is very large, such as 10,000. The improvement of Apriori algorithm was proposed such as the method of dynamic itemsets counting (DIC) [1]. It may decrease many passes of scanning database by counting different sizes of itemsets simultaneously under the assumption of homogenous data distribution. The other method concentrates the improvement of saving memory, computing quickly, and rule pruning with greater than the threshold of support and confidence. In theorem, a time complexity of the exhaustive search algorithm for association rules is $O(2^n)$ mostly if not to consider a distribution of dataset. However, if this problem involves a MTS [2] requirement then it will incur an increasing complexity in incremental mining, which makes the problem into NP-complete category. Other related improvements about finding association rules were first formulated by Agrawal et al., [3][4][5][6][7][8][9]. On the other hand, Some literatures were proposed by the basis of heuristic algorithms [4][10][11].Generally speaking, these algorithms first generate a candidate set of large itemsets based on some heuristics, and then discover the subset that indeed contains large itemsets. This process can be done iteratively. Those large item sets will be used as the basis to generate the candidate set for the next iteration. For example, in [10], a heuristic function is used to expand some large k-itemsets into an (k + 1)-itemset, if certain constraints are satisfied.

1.3. Genetic algorithms (GA) and genetic programming (GP)

Genetic algorithms were developed by John Holland of the University of the Michigan beginning in the early 1960s. The basic genetic algorithm's key steps in the selective breeding of a population of individuals. The process of fitness proportionate selection chooses parents from the population on the basis of their fitness. Fitness is a problem specific property that describes an individual's performance quantitatively. Genetic recombination, or crossover, combines traits from pairs of parents to create offspring, which enter a new population, forming the next generation of individuals.

1.4. The concept of Bio-Chemistry Synthesis (BCS)

In this section, we state how a biochemistry synthesis principle can be applied to ameliorate the performance of evolution in GP. Firstly, we construct an object in a representative model for association rules. These objects are designed to simulate an active individual called protoplasm in biochemistry. Those individuals conform to a mechanism of activities as a basis of life, such as reproduction, crossover, and mutation. On the other hand, these objects simulate certain organism called polymers. For a process of synthesis of the organism, it is critical to understand the outside property of which an interaction of molecules shows up with tuning environment parameter. For instance, many properties of mechanism and rheology may be predicted if the configuration of molecular chains is known. Therefore, we develop an overlapping algorithm to simulate a structure of molecules in stereochemistry in order to construct a virtual lattice. The detail definitions for virtual lattice will be made later. These virtual lattices are endowed with a measure called a general characteristic value, which it is equivalent to an average molecular weight of polymers. These general characteristic value M_w will be used to as fitness of GP for selecting a better individual. We utilize a reasonable assumption of which a 'good' large itemset indicates that this occurrences of a large itemset appears frequently and its variation is relative lower within nearby past in a neighbors of LISs. In other words, a large itemset should be stably presented at LISs. Once one stable feature F_i is evolved through a synthesis of feature $F_1..F_n$, then the last feature F_σ will be generated finally. These good individual are selected, in which their fitness is better in a GP part, depending on during the synthesis environment parameter¥. Hence, fitness of GP of the best simple formula is shown as the following:

Fitness = $(M_w$+¥$)$

To conclude that the following five requirements may not be satisfied simultaneously if only traditional algorithms are used for association rules:

I. A huge web-like database of which has numerous and dynamic online transactions. It must efficiently process a small itemset into a large itemset during incremental update without scanning overall database.
II. The model can predict the distribution of large itemsets in the future.
III. The model can learn the feature of database through several passes of incremental data mining.
IV. The distribution of large itemsets depends on a transition of time.
V. The effective factor must be found for a distribution of large itemsets.

2. Design framework

Hence, we design a model which can satisfy five requirements in above section. ◦ As shown in Figure 1, we combine Genetic Programming (GP) and Biochemical Synthetic (BCS) principle to be the kernel of the system. Firstly, the transactions were processed into many different sets of large itemsets called as large itemsets streams (LISs) S_1~S_k by means of an algorithm of association rules segment by segment during different time slices. The LISs are arranged for the overlapping algorithm as initial population of GP. Any efficient algorithms of association rules such as Apriori algorithm should be applied to

generate those LISs. Next, these n passes of GP will separately evolve into generating a delegated feature primary feature F_1~F_n for that population. Finally, The distribution of large itemsets within the database will be predicted by F_σ of which through several passes of synthesis process. Next, we observe whether r our algorithm is feasible. Suppose the transactions are divided into segments with the average size *m* and then input into the system in MTS. Each population needs average k times of evolutions, so the total needs k**m* time quantity. In fact, a genetic algorithm can set a terminated condition such as running k cycles, where k is constant. As a whole, the time complexity of our algorithm is *O(m*). It is independent of the number of items.

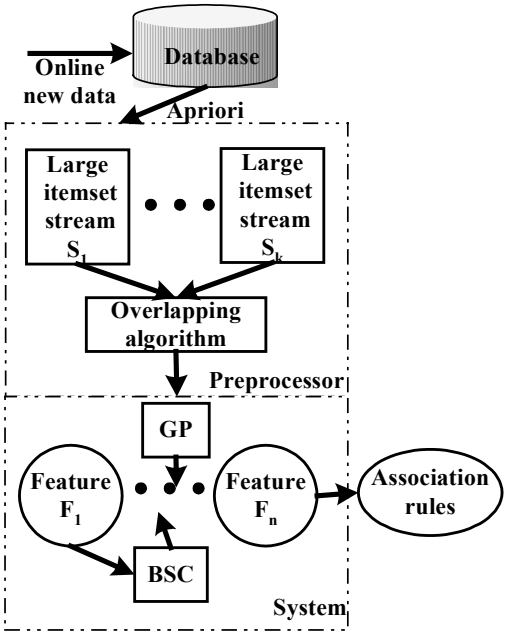


Figure 1. The architecture of finding the feature of association rules on huge database in multivariate time series

3. Modeling VL as stereochemistry structure

For the sake of representation of space complexity in a NP-Complete problem, we design a virtual lattice that has four varied forms: Lisp, tree, reading (3-dimensions), and vector, where form is a representation of problem solutions. These virtual lattices will own the resembling physical properties of polymers and activities of organisms. In this paper, a VL is an overlapping representation of a set of large item sets. It substantially represents a set of association rules on a certain dynamic database, too. Here, the overlapping algorithm ignores any 1-itemset. As shown in figure 2, these VLs take the 2-itemset of large itemsets 'I' as a kernel K by mean of overlapping algorithm. Next, the overlapping algorithm groups the others itemset except for 2-itemset into peripheral itemsets P that is a part of VL. The K➔P or P➔P links are connected after removing the redundant items from itemsets of each 'I'. Iteratively, these kernels of k-itemsets are separately formed for other VLs until overall k except for k=1 is processed. Since a kernel K of these VLs might appear onto P' linked by K' of another VL, so the virtual links of these K→P' or P→P' are similar to the structure of stereochemistry for polymers or organisms in the nature.

Stereochemistry is a branch of chemistry for analyzing the relationship of space among atoms of molecules. It is a basis of theory of a synthesis of polymers. The stereo structure is hold by hydrogen bond and van der Waals forces. As -mentioned above, the K of VLs simulates to be as a cell nucleus or an atomic nucleus. Single VL simulates to be as protoplasm or an organic molecule. The whole set of VLs simulate to be as a cell or polymer. These internal links K➔P and P➔P of VL simulate to be as a hydrogen bond. These external links K→P' and P→P' of VL simulate to be as a Van der Waals forces. In next section, a virtual lattice will be formally defined.

Specially, DNA is one natural organic polymer, too. The encoding of DNA can be applied to GP for evolutionary computation. In this study, The VLs simulate activities of owning a mechanism inheritance and evolution of genes to be as a cell if the handle of VL is an evolutionary process. However, the VLs simulate physical properties of owning a mechanism of kinetics of molecules to be as a polymer if the handle of VL is a synthesized process.

The overlapping grouping algorithm

```
Input: W (Weight of items), L (Large itemsets)
Output: VL (The lisp forms of VL)
Initialize empty sorted sets S order by K (Size of L)
Occurrence = 0
For each transaction
  Compute K
  If L exists in S then add 1 to Occurrence
  Add L to S
End For
While S has next L
  Create an empty vector V
  Set current = get next L as gamma node
  Set nucleon = current
  Add current to vector V
  While S has next L
    Add current – nucleon to vector V
    Set current = get next L
  End While
  Set n = Size of V
  For z=1 to n-1
    If V(z+1) contains V(z) Then
      If V(z+1).K – V(z).K = 1 then generate beta node
```

If V(z+1).K – V(z).K > 1 Then
Generate alpha node
Set V(z+1) = V(z+1) – V(z)
End If
End If
If V(z+1) not contains V(z) then generate delta node
End For
Generate VL according to W and get V from n-1 to 1
S removed the first element set
End While

3.1.1. Definition of a virtual lattice

To obtain the optimal solution for a problem, the problem must be encoded into a representation of individual for genetic algorithm at firstly step. In this paper, the problem is to process enormous database for the variant of prediction of association rules in multivariate time series effectively and efficiently. Speaking alternatively, we detect the distribution of association rules beforehand for a huge database. Therefore, the solution of problem is association rules. In general, the basic solution is a derivative of Apriori algorithm mostly. Our system utilizes Apriori algorithm to be as a preprocessor for searching a set large itemsets among some segments of transactions.

The items are arranged as shown in figure 3. It needs total 2^m-1 of item sets are evaluated. In the example, m=4, the dotted line represents a lattice path 1, 2, 3 and 0. It is called lattice subpath if a subset of lattice path exists. The definition of a virtual lattice is a set of lattice subpaths including some intermittent segments.

3.1.2. The data structure of a virtual lattice

To process with programming effectively, the following four kinds of data structure is used for reading easily and modeling explanatorily. For example, suppose there are items: 0,1,2,3,4,5, to be rearranged as 1,0,2,3,4,5 by the order of important weight. There are 5 large itemsets as flowing:

{1 0}, {1 0 2}, {10 4 5}, {1 0 2 3 4}, {1 0 2 3 4 5}

As shown in figure 4, a binary tree form is utilized as the data structure of an encoded individual with grammar base genetic programming [12] for the problem of association rules. The external nodes are arranged and attached to internal node with overlapping from left to right. The external nodes called items have a unique number and form some leaves of a tree. The design of overlapping arrangement is for saving the storage of genes. But the internal nodes are generated from left to right and then from top to bottom. However, the internal nodes are not overlapped. These encoded internal nodes α,β,γ,δ have been mathematically defined in section 3.1.1. The γ node stands for the recombination of these large itemsets, which cannot to be selected to execute during crossover of GP.

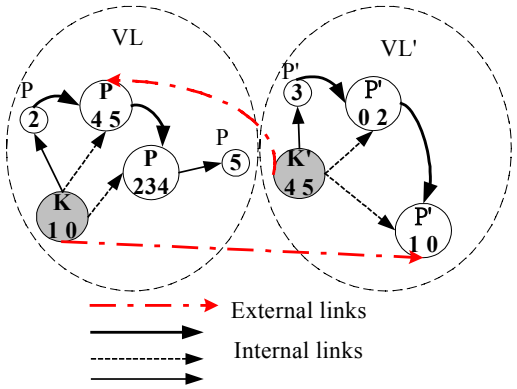


Figure 2. The diagram of simulated stereochemistry for virtual lattices

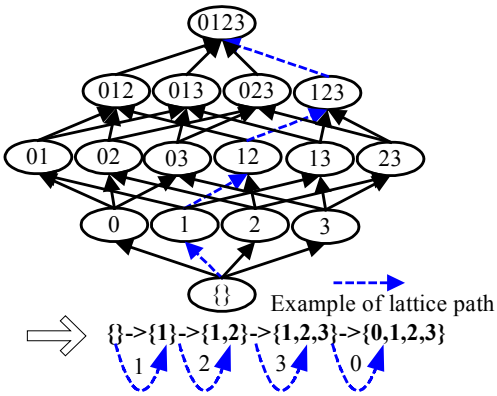


Figure 3. The diagram of itemsets use a lattice representation

However, for the sake of jumping outside a local maximal point, the mutation of GP might be still recombined. The β node expresses that a γ node plus a single item although it is not like a γ node tightly combined, but it is still a continuous arrangement of genes.

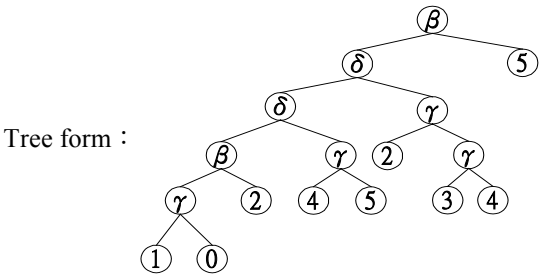


Figure 4. Example of a tree form for one virtual lattice

On the other hand, the α node will express a set of items that skips over two items above and forms another lattice subpath, so it may be loosely combined and conforms to a definition of lattice. However, The δ node expresses a lattice subpath that does not conform to a definition of lattice. There is no joined point between two subpaths, so

it conforms to the definition of a virtual lattice. This is a feature of polymers that no certain form such as random coil model. A lattice plus the property of δ node is called a virtual lattice.

Lisp form：

$$(\alpha\ (\delta\ (\delta\ (\beta\ (\gamma\ 4\ 5)3)(\gamma\ 0\ 2))(\gamma\ 1\ 0))(\gamma\ 2\ 3))$$

Figure 5. Example of a Lisp form for one virtual lattice

As shown in figure 5, an expression of Lisp language is helpful and is convenient to code as initial population of GP in programming. The most inner pair of parentheses is executed first; next executions from inside to outside, and then finish the whole evaluation of a Lisp expression ultimately.

As shown in figure 6, reading from the gray circle γ node along with the arrowed path to the end with δ node or null node. This case has three diverse virtual lattice subpaths: (1) {1 0 2}, (2){10 4 5}, and (3){1 0 2 3 4 5}. The dotted lines of lattice subpaths (2) and (3) express that they have the same kernel node but splitting into two subpaths by δ node. Since the two lattice subpaths depart from a joint of lattice subpath (1), it brings a crystal of lattice to a no certain form of virtual lattice. A degree of the phenomenon will affect physical property of VL. For programming, the data structure of internal operation of an overlapping algorithm is shown as below:

Vector form: {1 0}→{2}→{4 5}→{2 3 4}→{5}

Reading Form：

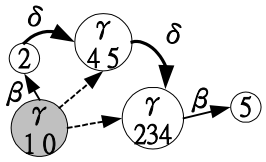


Figure 6. Example of a reading form for one virtual lattice

In contrast to the solid line in figure 6 and the internal nodes in figure 4, it clearly shows an order of internal nodes γ, β, δ, γ, δ, γ, γ, and β. The result is produced by a preorder search of tree within recursive programming.

As shown in figure 7, the example of overlapping grouping algorithm is presented by a set of test data.

4. A comparison of GP plus BCS and the other methods

Our approach outperforms other traditional exhaustive search on the problem of incremental mining. The reasons are addressed as below.

I. For the exhaustive algorithm, if its space complexity is $O(2^n)$, then the approach will certainly and quickly find an infeasible boundary for space complexity so that the exhaustive algorithm is unavailable while a space growth beyond the infeasible boundary. But there is no infeasible boundary for heuristic algorithm such as a genetic programming.

II. Since multiple search of GP will be facile to find out a group of sub-optimal solutions, an adaptive BCS model can stably converge on a tolerant region for an error of prediction. Hence, the sampling time and the number of large itemsets in unit time will substantially reduce if the most number of large itemsets are recognized on dynamic database with little change on the distribution of large item sets. So, the time complexity $O(m)$ of our algorithm is within the feasible boundary during a reasonable the number of large item sets.

III. Owing to that the exhaustive algorithm must search overall database for counting small itemsets to became a large itemset so that make incremental mining infeasible. Therefore, Our algorithm would be a feasible approach for solving the problem of incremental mining.

5. Conclusion and future work

We have presented functionality analysis for association rules based on the principle of biochemistry evidences, which we believe it is a useful and intuitive measure than other association rule's finding algorithm in multivariate time series.

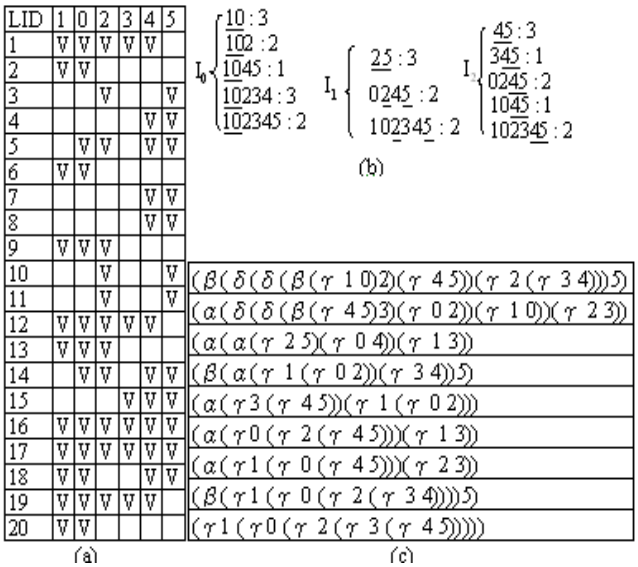


Figure 7. Examples of generating three groups by an overlapping grouping algorithm, (a) Many different sizes of LISs are combined as input, (b) List only 2-itemset kernel in this case, (c) Results of the algorithm written in Lisp as an initial population of GP for evolution.

Consequently, a new approach to implement incremental data mining for association rules for dynamic database is proposed in this work.

The experimental data is collecting currently Meanwhile, there are still some issues not been discussed in this paper yet, and it is very worthy to further study at the scope of data mining such as its meaning of α, β, γ, and δ node in biochemistry. Especially, it is critical to investigate a synthesizer, which can affect the growth of the BCS theorem. In the future, our research will concentrate on how to effectively synthesize a feature of domain knowledge in real world and show it is a useful and feasible predicted model.

References

[1] S. Brin, R. Motwani, J.D. Ullman, and S. Tsur, "Dynamic Itemset Counting and Implication Rules for Market Basket Data", *SIGMOD Record*, Volume 6, Number 2: New York, June 1997, pp. 255-264.

[2] Tucker, A. Swift, and S. Liu, "Variable grouping in multivariate time series via correlation", *Systems, Man and Cybernetics, Part B*, IEEE Transactions on, Volume: 31 Issue: 2, April 2001, pp. 235-245.

[3] R. Agrawal, T. Imilienski, and A. Swami, "Data base Mining: A Performance Perspective", *IEEE Transactions on Knowledge and Data Engineering*, IEEE , December 1993., pp. 914-925.

[4] R. Agrawal, T. Imilienski, and A. Swami, "Mining Association Rules between Sets of Items in Large Databases", *Proc. Of the ACM SIGMOD Int'l Conf. On Management of Data*, May 1993, pp. 207-216.

[5] R. Agrawal, K. Lin, S. Sawhney, and K. Shim, "Fast similarity search in the presence of noise, scaling and translation in timeseries databases", *In Proc. Of the Int'l Conf. On Very Large Data Bases (VLDB)*, 1995, pp. 490-501.

[6] R. Srikant and R. Agrawal, "Mining generalized association rules", *Proceedings of the 21st International Conference on Very Large Data Bases (VLDB'95)*, Zurich, Switzerland, 1995, pp. 407-419.

[7] J.S. Park, M.S. Chen, and P.S. Yu, "An effective hash-based algorithm for mining association rules", *In Proc. 1995 ACM-SIGMOD*, pp. 175-186.

[8] S. Brin, R. Motwani, and C. Silverstein, "Beyond market basket: Generalizing association rules to correlations", *In Proc. 1997 SIGMOD*, pp. 265-276.

[9] H. Toivonen, "Sampling large databases for association rules", *Proc. Of the Int'l Conf. On Very Large Data Bases (VLDB)*, 1996, pp. 134-145.

[10] R. Agrawal and S. Srikant, "Fast Algorithms for Mining Association Rules in Large Databases", *Proceedings of the 20th International Conference on Very Large Data Bases*, September 1994, pp. 487-499.

[11] M. Houtsma and A. Swami, "Set-Oriented Mining of Association Rules", *Technical Report RJ 9567*, IBM Almaden Research Laboratory, San Jose, CA, October 1993.

[12] M. L. Wong and K. S. Leung, *Data mining using grammar based genetic programming and applications*, Boston: Kluwer Academic, 2000.

A Force Field Model for Guided Cluster Discovery

C.H. Li
Department of Computer Science
Hong Kong Baptist University
Hong Kong

Abstract— **Cluster discovery is an essential part of many data mining applications. While the cluster discovery process is mainly unsupervised in nature, it can often be aided by a small amount of labeled data. A novel unified energy equation for clustering that incorporates both labeled data and unlabeled data is introduced. This formulation is posed as a force-field model integrating labeling constraint on labeled data and similarity information on unlabeled data for joint estimation. Experimental results show that good clusters can be identified using small amount of labeled data.**

I. Introduction

In machine learning for classification problems, there are two distinct approaches to learning or classifying data: the supervised learning and un-supervised learning. The supervised learning deals with problem where a set of data are labeled for training and another set of data would be used for testing. The un-supervised learning deals with problem where none of the labels of the data are available. Unsupervised clustering can be broadly classified into whether the clustering algorithm is hierarchical or non-hierarchical. Hierarchical methods often model the data to be clustered in the form of a tree, or a dendrogram [1]. The lowest level of the tree is usually each datum as a cluster. A dissimilarity measure is defined for merging clusters at a lower level to form a new cluster at a higher level in the tree. The hierarchical methods are often computationally intensive for large number of samples and is difficult to analyze if there is no logical hierarchical structure in the data.

Non-hierarchical methods divide the samples into a fixed number of groups using some measure of optimality. The most widely used measure is the minimization of the sum of squared distances from each sample to its cluster center. The k-means algorithm, also known as Forgy's method [2] or MacQueen [3] algorithm is a classical algorithm for non-hierarchical unsupervised clustering. However, the k-means algorithm tends to cluster data into even populations and rare abnormal samples in medical problems cannot be properly extracted as individual clusters.

In recent years, important data mining tasks have emerged with enormous volume of data. The labeling of a significant portions of the data for training is either infeasible or impossible. Sufficient labeled data for training are often unavailable in data mining, text categorization and web page classification. A number of approaches have been proposed to combine a set of labeled data with unlabeled data for improving the classification rate. The co-training approach has been proposed to solve the problem of web page classification where the web pages can be represented by two independent representations [4]. The drawback of this co-training approach is that not all data have two independent representations and the algorithm is thus not easy to be generalized. Subsequently, a similar co-training method is invented for combining labeled and unlabeled data by co-training with two learning algorithms [5]. Instead of using two representations of the data, this co-training algorithm uses two learning algorithms. The naive Bayes classifier and the EM algorithm have been combined for classifying text using labeled and unlabeled data [6]. A modified support vector machine and non-convex quadratic optimization approaches have been studied for optimizing semi-supervised learning [7].

II. Guided Cluster Discover and Classification

The guided cluster discovery is closely related to classification problem. In this section, we will look at the similarity and the differences between guided cluster discovery in data mining and the general pattern classification problem.

Suppose the classification task is to classify a set of data denoted by x_i $(i = 1, ..., m)$ into two classes denoted with labels $[A, B]$ respectively. The classification algorithm is to find a corresponding label $y_i \in [A, B]$. The set of all data in the dataset is denoted as x and the set of all labels y. The cardinality of both x and y is m.

In the classification problem, a fraction of data in the dataset are labeled and the remaining data are to be classified. The set containing all labeled data are denoted as x_L where for each element $x_i \in x_L$, the label y_i for that data is known. Similarly, the set of unlabeled data are denoted by x_U and the set of unknown labels are denoted by y_U. The usual non-intersecting requirement follows: $x_L \cup x_U = x$, $y_L \cup y_U = y$. Using this notation, the distinction between the data available in data mining and general machine learning is shown in Table I. In traditional classification, a large amount of labeled training data is usually available for training a mapping between the training data and the corresponding labels. In data mining, a large amount of unlabeled data is available and it is often costly or even impossible to assign labels to a significant portion of the unlabeled data for learning.

In order to fully utilize the unlabeled data, guided cluster discovery significantly improves classification accuracy by incorporating unlabeled data for training. In traditional classification, the training phase is carried out by constructing a mapping between the training samples pair though the labeled dataset $\{x_L, y_L\}$. In the estimation phase, the labels of the unknown testing data $x_i \in x_U$ will be obtained. This situation is generally applicable to different areas of pattern recognition and machine learning, especially for situations where the unknown testing data are obtained sequentially in time. However, in the case of data

TABLE I
Classification and Guided Cluster Discovery

	Classification	Guided Cluster Discovery
No. of labeled data $n(x_L)$	Large	Small(a few)
Labeled data	Well-defined	Ill-defined/To be discovered
No. of unlabeled data $n(x_U)$	Small	Large
Use of unlabeled Data x_U	Testing	Training/Analysis

mining, the unlabeled data x_U are already available for the learning algorithm and the only unknown is the set y_U. As demonstrated by the co-training method in classifying web pages and the color tracking with transductive learning method, the use of unlabeled data x_U can significantly increase the classification accuracy in machine learning tasks.

III. Probability Force Field For Semi-supervised Learning

The semi-supervised learning is a learning process that finds the set of labels y_U given the data x_U, x_L and y_L. Instead of a direct estimation on the actual labels y_i we estimate the probability of labels being a given label. For a two class classification problem with class labels $[A, B]$, we represent the probability of the data i taking the labels A as $P_i = P(y_i = A)$. For labeled data $x_i \in x_L$, if $y_i = A$, then $P_i = 1$, else if $y_i = B$, then $P_i = 0$. The probabilistic guided cluster discovery is to estimate the P_i for all i where $x_i \in x_U$.

The construction of the probabilistic force-field model depends on the following assumptions:

- Data vectors with small Euclidean distances between them will have similar probabilities
- Labeled data vectors has fixed probability of either 1 or 0
- The probability P_i of unlabeled data vectors freely distributes themselves to settle in a optimal configurations as defined by energy equation defined by the above two constraint.

The first assumption of spatial close data vector having similar probability can be modeled using attractive force between data vectors in high dimension vector spaces. The force between two data vectors i and j with inverse power law is given by,

$$F_{ij} = \frac{G}{r_{ij}^c} \tag{1}$$

where r_{ij} is the Euclidean distance between i-th vector and j-th vector, c is an integer constant, and G is a fixed positive constant defining the strength of attraction. For example, the gravitation law is an inverse square law with c equals to 2. Suppose that the data vector x_i is in m-dimensional space, we embed the probability into the data vector x_i to form a $m+1$-dimension vector $[x_i, \alpha P_i]$, where α is a constant balancing the scale of the probability and the data vector. The Euclidean distance between two extended vectors in the $m+1$ space is given by,

$$r_{ij} = \sqrt{|x_i - x_j|^2 + \alpha(P_i - P_j)^2} \tag{2}$$

where $|x_i - x_j|$ is the Euclidean distance in the m-dimension space and *alpha* is a positive constant balancing the scale of probability to the data vector space.

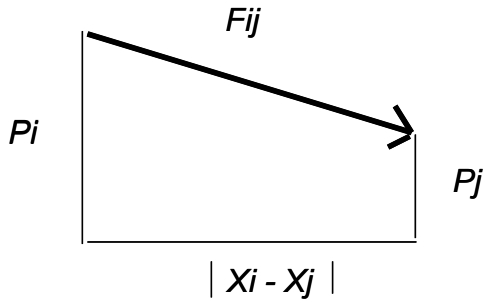


Fig. 1. Force vector between two probability vectors

The schematic representation of the embedded vector relationship is shown in Figure 1. The forces between probabilistic vector can then be written as

$$F_{ij} = \frac{G}{\sqrt{|x_i - x_j|^2 + \alpha(P_i - P_j)^2}}. \tag{3}$$

As the data vector x are fixed in spaces, the degree of freedom is along the freely distributable probability. Thus, the effective force on the probability vector is the component of the force along the probability axis

$$Fp_{ij} = \frac{G(P_i - P_j)}{\sqrt{|x_i - x_j|^2 + \alpha(P_i - P_j)^2}}. \tag{4}$$

Alternatively, a force-field energy approach can also be specified. In general, an attractive force can be equivalently represented by a force field energy equation where the energy experienced by a data point i is given by

$$U_i(P) = -\sum_j \frac{G'}{r_{ij}^{c-1}}, \tag{5}$$

where the dependence of P is effected through the dependence on r_{ij}. With P_i and P_j being small, r_{ij} will be minimized and the energy will be lowered. The estimated probability can then be solved by minimizing the energy of the system. The energy of the total system is given by

$$U(P) = -\sum_i \sum_j \frac{G'}{r_{ij}^{c-1}}. \tag{6}$$

An approximation of the above system can be readily obtained by the use of Markov assumptions [8]. Assuming that the interaction is localized by a fixed neighborhood, the above energy equation can be simplified to

$$U(P) = -\sum_i \sum_{j \in N_i} \frac{G'}{r_{ij}^{c-1}}. \tag{7}$$

where N_i is the neighbourhood of the data vector i. There are two choices of neighbourhood in high-dimensional space. First, we can define a hypercube with distance d centered at the data vector i. All data vectors inside this hypercube are elements in N_i. Alternatively, we can define N_i to be the set of k-th nearest neigbours of i. The use of nearest neighbours in constructing the Markov assumptions allows a fixed size neighbourhood for each data i and and a fixed computational $O(kn)$.

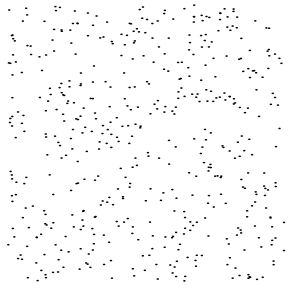


Fig. 2. Two clusters

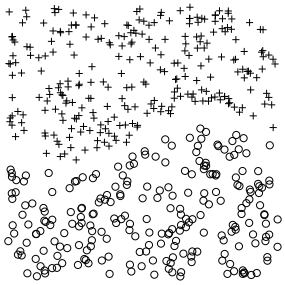


Fig. 3. Solution I

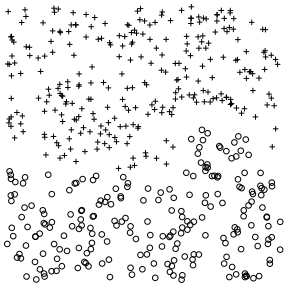


Fig. 6. Cluster Discovered by average link tree

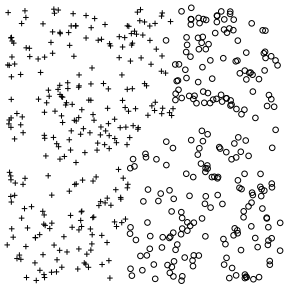


Fig. 7. Cluster Discovered by complete link tree

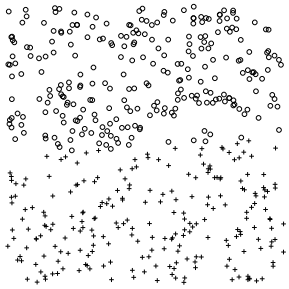


Fig. 4. Cluster Discovered by k-means algorithm

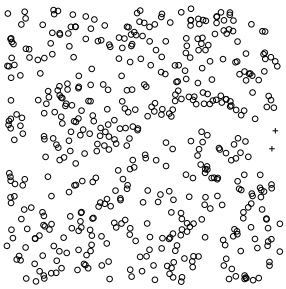


Fig. 5. Cluster Discovered by single link tree

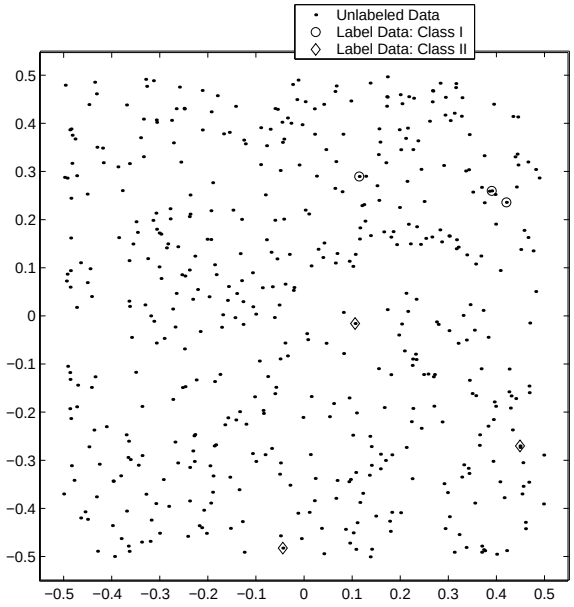


Fig. 8. Training Samples

IV. Results and Discussions

The guided cluster discovery is tested with the iris dataset and two synthetic datasets. Initially, the probability of known labeled samples are assigned to their values and the unknown labels are assigned a random values near 0.5. A gradient descent on the force field with inverse square law is used to update the probability for 1000 iterations. The 8-th nearest neighbourhood system is used in the two following experiments. The only unknown parameter in the experiments α is specified approximately by observing the ratio of the range of probability [0,1] to the ratio of the average ranges of data x.

A. Cluster Discovery on Non-linear Boundary

A synthetic dataset is used for testing the guided cluster discovery algorithm. Figure 2 shows the data for a two cluster clustering problem. The two clusters are seperated with a sinusoidal boundary. Figure 3 shows the ground truth of the dataset. Results of applying classical cluster discovery algorithms: the k-means algorithm, the hierarchical tree algorithms with single link, average link and complete link are shown in Figure 4, Figure 5, Figure 6, Figure 7 respectively. The clusters discovered by k-means and the average link tree are closer to the ground truth with some errors along the non-linear boundaries. The single link tree and the complete link tree has very poor performance for this dataset.

To test the performance of the guided cluster discovery algorithm on the synthetic data, three training samples are chosen from each classes. Figure 8 shows the training samples where the data with known labels are marked with circles and diamond respectively. With a small number of labeled training samples from each class, one can judge from the figures that inference based/decision surface based classifier will not be able to determine accurately the sinusoidal boundary between the cluster. Figure 9 shows the initial probability for the synthetic data. There are 500 data and the first three from each class are selected as training samples. The data with unknown labels are assigned with random probability in the range [0.45 0.55]. Figure 10 shows the initial estimated label, those data with probability above 0.5 is assigned to class I and those data with probability below 0.5 is assigned to class II. This shows that the labels are initially random, except at three pairs of training data.

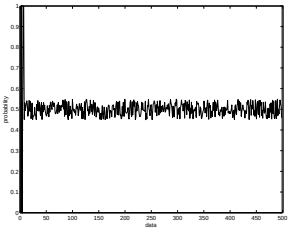


Fig. 9. Initial Probability

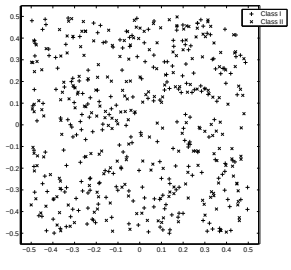


Fig. 10. initial estimated labels

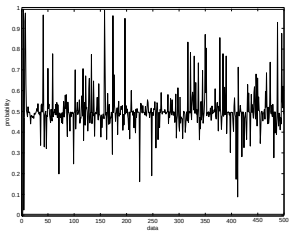


Fig. 11. Probability after 100 iterations

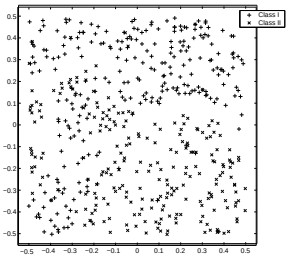


Fig. 12. Estimated labels after 100 iterations

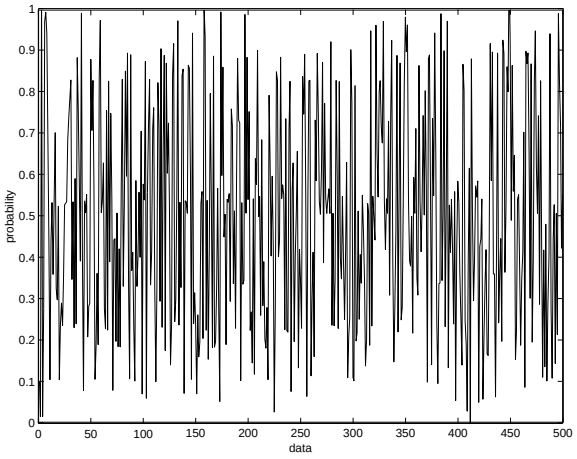


Fig. 13. Final Probability

The first step in force field based guided cluster discovery is the calculation of the 8-th nearest neighbour distance matrix from the data. The scale balancing constant α is chosen as 0.1. After updating the probability for 100 iterations, the probability for the data is shown in Figure 11. The probability is significantly modified after 100 iterations with some of the data point have confidence in having the label value. Those data point with probability close to 0.5 are data point whose label value is not certain. Figure 12 shows the estimated label at 100 iterations, those data with probability above 0.5 is assigned to class I and those data with probability below 0.5 is assigned to class II. There are some errors in the class labels, however those closer to the training samples are mostly correct. The final set of figures show the probability and estimated labels after 1000 iterations. Figure 13 shows that the probability have a wider range where more data are confidently determined after 1000 iterations. Figure 14 shows that the estimated labels after 1000 iterations is very accurate with only 3 errors in the middle left region.

B. Iris Dataset

The iris dataset is also used for testing the guided cluster discovery. The iris dataset consists of 150 samples of measurement of the iris plant. There are three species of iris in the dataset and each species has 50 samples. Typical approaches uses 100 samples for training and 50 untrained samples for testing. The iris dataset is well studied and results can be found in numerous literature [9].

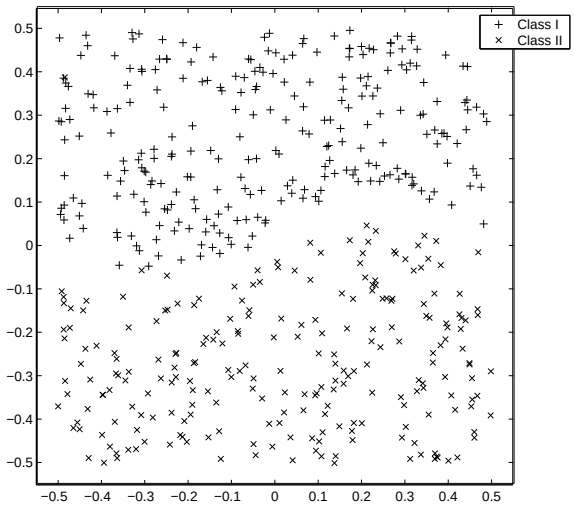


Fig. 14. Final estimated labels

In the first experiment in guided cluster discovery for iris dataset, we considered using only 2 labeled data. The use of such small amount of training data enables a 'what-if' scenario in data mining to be handled. Even if the true class is unknown, we can choose a few samples randomly and then evaluated the resultant classifications.

The first step in force field based guided cluster discovery is the calculation of the 8-th nearest neighbour distance matrix. After the calculation of the 8-th nearest neighbour distance, we observed that all the 50 samples from class I are nearest neighbours to members of its own class and thus these samples can be isolated without any further classification. This is also in good agreement with previous work in the iris dataset that the first class is linearly separable and can be classified trivially. Now we concentrate the classification of the second class and the third class. We pick the first sample in class II and first sample in class III as the labeled sample and denote this experiment as experiment (a). The initial probability before update is shown in Figure 15. The first sample is the labeled sample for class II and has a value of one, the 51 sample is the labeled sample for class III and has zero probability being class II. The updated probability distribution is shown in Figure 16. The scale constant α is set as 0.05. The probabilities of data from 1 to 50 belongs to class II, and their values are significantly due to the attractive force among each other and the attractive force from the labeled data. The forces of attraction can propagated through each other and different data feels a different level of attractive force according to their relative positions of the labeled data and other data vectors. The probabilities of data from 51 to 100 belongs to class III and their probabilities are significantly lower than those data that belongs to class II.

In the second experiment, experiment (b), we increase the number of training samples to 2 samples per class. The first two data from class II is used as samples for class II and plant 51 and 52 as samples from class III. The updated probability distribution is shown in Figure 17. The probabilities between the data in class II and class III are

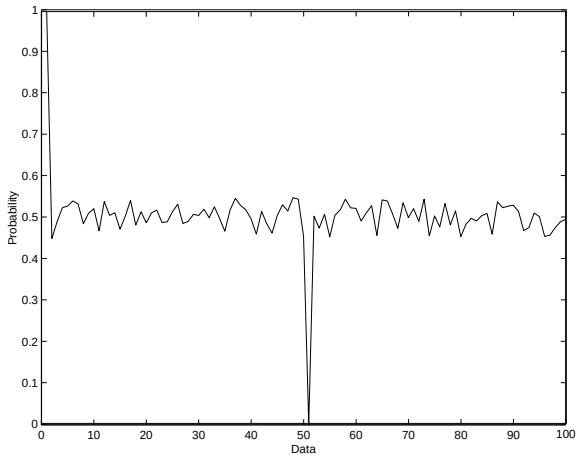


Fig. 15. Initial probability of class membership

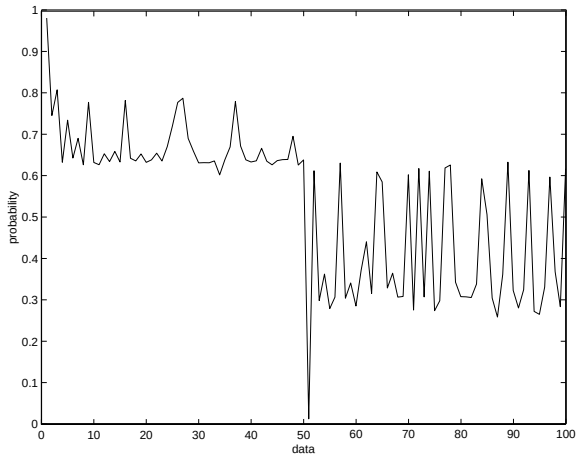


Fig. 16. Probability of class membership after transductive learning: Experiment (a)

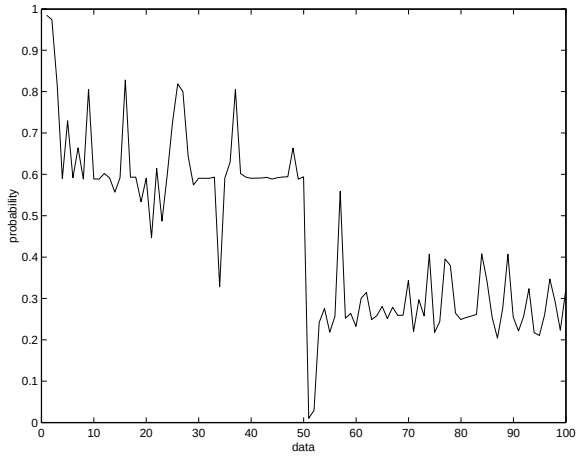


Fig. 17. Probability of class membership after transductive learning: Experiment (b)

TABLE II
NUMBER OF ERRORS AFTER GUIDED CLUSTER DISCOVERY ON IRIS DATASET

	errors in Class II	errors in Class III
Exp. (a)	0	15
Exp. (b)	3	1

much better seperated in this case. For the samples in the first 50 data, only three have probabilities below 0.5. For the samples from 51 to 100, there is only one data with probabilities above 0.5.

Table II shows the number of misclassifications of the two experiments in guided cluster discovery of the iris dataset. The true labels are taken as the class probabilities above 0.5. Similar results are obtained using other labeled samples as training data. In general, it is found that if the labeled samples represents typical features in the respective class, the learning performances would be acceptable. If a selected sample has feature vector that is similar to samples from the other class, the performances of the learning would be significantly lower.

The errors achieved by the guided cluster discovery algorithm is similar to classical classification algorithm. However, classical classification algorithm often requires up to 100 training samples for achieving this accuracy.

C. Gaussian Clusters

The third experiments deals with classification of three Gaussian Clusters. Figure 18 shows the ground truth of the three Gaussian clusters. The three Gaussians Clusters are of different variances and different sizes. To test the force-field method, a random points are selected from each cluster as a labeled data from each cluster. Figure 19 shows the training samples used for guided cluster discovery. The initial probabilities estimation is calculated differently from the previous two experiments. In this experiment, we use the nearest neighbour classifier for setting up the initial probabilities for guided cluster discovery. The result of the nearest neighbour classification on the dataset is shown in Figure 20. As there are only one random sample chosen as the labeled sample, the initial classification result has quite a large amount of errors. However, this initial classification result is utilized as the setting up the initial probabilities for transductive learning. The initial probabilities of the i-th data being class j, denoted as p_i^j, is assigned with the following rules:

- assign $p_i^j = 1$ if $y_i = j$ where $y_l \in y_L$,
- assign $p_i^j = 0$ if $y_i \neq j$ where $y_l \in y_L$,
- assign $p_i^j = 0.75$ if x_i is classified by the Nearest neighbour classifier to be class j, and
- assign $p_i^j = 0.25$ if x_i is classified by the Nearest neighbour classifier to be any class other than j.

The initial probabilities for the Class I p^1 is shown in Figure 21. As the data from 1 to 512 belongs to class I, those data with probabilities value 0.25 are data which are incorrectly classified by the nearest-neighbour classifier. The guided cluster discovery algorithm is applied to the prob-

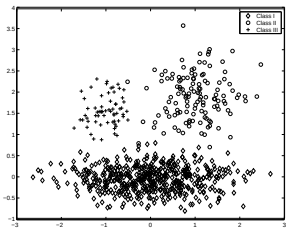


Fig. 18. Ground Truth of 3 Gaussian Clusters

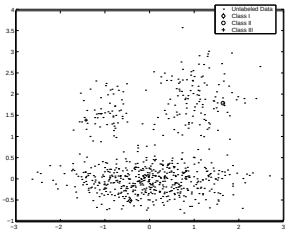


Fig. 19. Training Data of 3 Gaussian Clusters

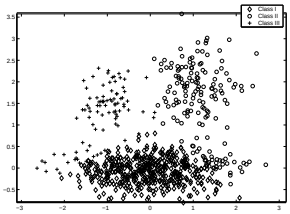


Fig. 20. Nearest Neighbour classification of 3 Gaussian Clusters

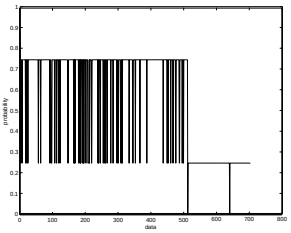


Fig. 21. Initial probabilities for Class I, p^1

abilities for 1000 iterations with α set as 0.1. Figure 22 shows the probabilities after the iterations. The probabilities of for data from 1 to 512 is raised significantly. The only data with values below 0.5 are located around number 310. From data 513 to 703, most of the data have probabilities below 0.5 and are thus correctly inferred as data not belonging to class I. There are only two data with probabilities above 0.3 in this portion. The estimation procedure can be repeated for estimating the other two class labels. The final estimated labels after calculating all probabilities of the three classes are shown in Figure 23. The estimated labels are in very good agreement with ground truth and visual assessment based on proximity criterion.

V. Conclusion

The guided cluster discovery problem is solved using attractive force-field formulations. The guided cluster discovery algorithm achieves integration of labeled information and spatial proximity information with force-field equations. Furthermore, the use of Markov assumptions allows a computationally feasible formulations to be developed. The guided cluster discovery approach excels in situations where a very small amount of labeled data and a large amount of unlabeled data is available for analysis. Experimental results shows that good clustering results can be obtained with only a few training samples as guidance and the spatial structure inherent in the problem can be readily utilized for improving the clustering process.

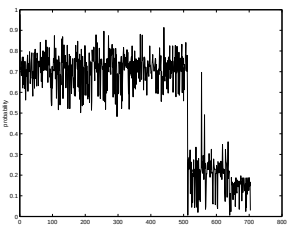


Fig. 22. Final probabilities for Class I, p^1

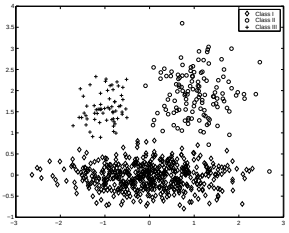


Fig. 23. Final Estimated Labels

References

[1] B.D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, Cambridge,UK, 1996.

[2] E. Forgy. Cluster analysis of multivariate data:efficiency vs. interpretablility of classifications. *Biometrics*, 21:768, 1965.

[3] J. MacQueen. On convergence of k-means and partitions with minimum average variance. *Ann. Math. Statist.*, 36:1084, 1965.

[4] A. Blum and Shuchi Chawla. Combining labeled and unlabeled data with co-training. In *The Eighteenth International Conference on Machine Learning*, 2001.

[5] S. Goldman and Y. Zhou. Enhancing supervised learning with unlabeled data. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[6] K. Nigam, A. McCallum, Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using em. *Machine Learning*, 34(1), 1999.

[7] T. S. Chiang and Y. Chow. Optimization approaches to semi-supervised learning. In M. C. Ferris, O. L. Mangasarian, and J. S. Pang, editors, *Applications and Algorithms of Complementarity*. Kluwer Academic Publishers, 2000.

[8] Ross Kindermann. *Markov random fields and their applications*. American Mathematical Society, Providence, R.I, 1980.

[9] M. Nadler and E. P. Smith. *Pattern Recognition Engineering*. Wiley-interscience, New York, 1993.

Feature Completion for Data Mining

T. Y. Lin

Department of Mathematics and Computer Science
San Jose State University,
San Jose, CA 95192
tylin@cs.sjsu.edu

ABSTRACT
Choosing a "correct" set of attributes is vital in data mining. Naturally one might asked: *Are there ways to examine all possible attributes (features)?* Via certain notion of isomorphism, mining high frequency patterns of a given relation can be reduced to mining in an associated machine oriented canonical model. In such a model all derived attributes can be determined – a positive answer to the question. It is important to observe that a canonical model represents an isomorphic class, hence semantically neutral. Roughly two relations are isomorphic, if one can be transformed to the other by renaming their attribute values.

1. INTRODUCTION

In traditional data mining, we search data patterns in a given set of attributes. The design or selection of such a set, in a typical database environment, is primarily for record keepings, *not* for the understanding of real world. Hence, it is highly possible that there are no meaningful patterns in the given set of attributes; see some mathematical example in Section 7. So certain new attributes, ***called derived attributes*** **(features),** which are transformed from the given set, may be needed. Natural questions are:

Q1: Could one examine all possible attributes features?
Q2: and to identify the ***desirable*** *ones?*

In this paper, we focus on Q1. At first glance, Solving Q1 seems an impossible task. Somewhat a surprise, we can enumerate them, that is, there is a positive answer.

2. MAIN IDEA

A relation will be viewed as a knowledge representation of a set of real world entities. First, we classify all such representations into isomorphic classes (Definition 2). Then, we observe that the high frequency patterns (large itemsets) can be generated by elementary operations from the patterns of a canonical member of an isomorphic class. Such a canonical member is called a *simplified relation*. So data mining is reduced to the search of patterns on a simplified relation (Theorem 1). A canonical model that represents the simplified relation is constructed. Based on this canonical model, all possible derived attributes (features) can be determined. This answers Q1 positively. It is important to note that the canonical model represents all isomorphic relations, so it is semantically neutral. Therefore the new derived attributes are also semantically neutral. The semantics of an individual relation, including the new derived attributes, have to be re-interpreted by the users. Finally, we include a section that explains an obvious conflicts.

3. A LITTLE THEORY

3.1. Basic Notions

Two measures, called the support and confidence, are used in mining association rules. In this paper, we will be interested only in the *high frequency patterns*, not necessary in the forms of rules. So the main concern is *the support* only. For definitive, we assert

an item is an attribute value,
an q-itemset is a sub-tuple of length q, and
the main focus is on *high frequency q-patterns(large q-itemsets)*

Let V be a set of real world entities, and $A = \{A^1, A^2, \ldots, A^n\}$ a set of attributes. As usual Dom(-) denote the domain. A bag relation K (we allow repeated rows) is regarded as a function (a knowledge representation)

$$K: V \rightarrow Dom(A^1) \times Dom(A^2) \times \ldots \times Dom(A^n),$$

which maps entities to tuples. The (function) image of K is a bag relation, and its graph (x, K(x) is the information table. By abuse of languages and notations, K may mean *the knowledge representation, information table (simply table), or the bag relation.* To avoid the notion of bags, we favor the term information table. An attribute can be interpreted as the composition of K and the projection P^i: $Dom(A^1) \times Dom(A^2) \times \ldots \times Dom(A^n) \rightarrow Dom(A^i)$

$$A^i: V \rightarrow Dom(A^i).$$

Example. Table 1 illustrates the knowledge representation:

$$K: V \rightarrow (S\#, STATUS, RANK, CITY).$$

Table 1. The Information Table K

V	K	(S#	STATUS	RANK	CITY)
v_1	→	(S_1	TWENTY	2	C_1)
v_2	→	(S_2	TEN	3	C_2)

v_3	→	(S_3	TEN	3	C_2)
v_4	→	(S_4	TEN	3	C_2)
v_5	→	(S_5	TEN	3	C_2)
v_6	→	(S_6	TEN	3	C_2)
v_7	→	(S_7	TWENTY	2	C_3)
v_8	→	(S_8	THIRTY	1	C_3)
v_9	→	(S_9	THIRTY	1	C_3)

The right hand side of the arrow, (S#, STATUS, RANK, CITY) is the image of K, a classical relation, and the left hand side is the set V of entities (independent variables). Mathematically, Table 1 is the graph, (v, K(v)) of K.

3.2. The Isomorphic Class of an Information Table

Definition 1. Attributes A^i and A^j are isomorphic iff there is a one-to-one and onto map, s: $Dom(A^i) \rightarrow Dom(A^j)$ such that $A^j(v)= s(A^i(v))$, $\forall\ v \in V$. s is called an isomorphism.

Intuitively, two columns are isomorphic iff one column turns into another one by properly renaming its attribute values.

Definition 2. Let K and H be two information tables with the same universe V. Let $A =\{A^1, A^2, \ldots, A^n\}$ and $B =\{B^1, B^2, \ldots, B^n\}$ be the attributes of K and H respectively. Then, K and H are said to be isomorphic if every A^i is isomorphic to some B^j, and vice versa. It is a strict isomorphism, if K and H have the same number of columns.

This isomorphism is reflexive, symmetric, and transitive, so it classifies all relation instances into equivalence classes; we call them isomorphic classes. Recall that the number of columns is called the degree of a relation.

Definition 3. H is a *simplified information table* of K, if H is isomorphic to K and has non-isomorphic attributes only.

Theorem 1. Let H be a simplified information table of K. The high frequency patterns (large itemsets) of K can be obtained from those of H by *elementary operations* that will be defined below.

We will consider the case, n = degree K - degree H =1, and there are two isomorphic attributes B and B' in K. Let the isomorphism be s: $Dom(B) \rightarrow Dom(B')$; we write b'=s(b). Let H be the new table in which B' has been removed.

Proposition 1. The high frequency patterns of K can be obtained from those of H by *elementary operations*, namely,

(1). If b is a large itemset in H, then b' and (b, b') are large in K.
(2) If (a. ., b, c. . .) is a large itemset in H, then (a. . , b', c. . .) and (a. . , b, b', c,. . .) are large in K.
(3) These are the only large itemsets in K.

The validity of this proposition is rather straightforward, we skip the proof, but illustrate by an example below.

Proposition 1 provides the critical step for Theorem 1

Example. Attributes, STATUS, RANKS are isomorphic. By removing RANK or STATUS from Table 1, we have Table 2 or Table 3 respectively.

Table 2. A Simplified Table H: RANK is removed.

V	H	(S#	STATUS	CITY)
v_1	→	(S_1	TWENTY	C_1)
v_2	→	(S_2	TEN	C_2)
v_3	→	(S_3	TEN	C_2)
v_4	→	(S_4	TEN	C_2)
v_5	→	(S_5	TEN	C_2)
v_6	→	(S_6	TEN	C_2)
v_7	→	(S_7	TWENTY	C_3)
v_8	→	(S_8	THIRTY	C_3)
v_9	→	(S_9	THIRTY	C_3)

Table 3. Another Simplified Table: STATUS is removed

V	H'	(S#	RANK	CITY)
v_1	→	(S_1	2	C_1)
v_2	→	(S_2	3	C_2)
v_3	→	(S_3	3	C_2)
v_4	→	(S_4	3	C_2)
v_5	→	(S_5	3	C_2)
v_6	→	(S_6	3	C_2)
v_7	→	(S_7	2	C_3)
v_8	→	(S_8	1	C_3)
v_9	→	(S_9	1	C_3)

Let us assume we require the support ≥ 4. It is easy to see that in H (Table 2):

the 1-itemsets, TEN, and C_2, are large,
the 2-itemset, (TEN, C_2), is large

By elementary operations, we can generate the large itemsets of K as follows:

the 1-itemsets, TEN, C_2 and 3, are large
the 2-itemsets, (TEN, 3), and (C_2 , 3), are large
the 3-itemset, (TEN, C_2, 3), is large

It is obvious that searching K is more expensive than H (degree K > degree H), so to find the patterns in K, we will find them on H, and then do the elementary operations.

In next subsection, we will find a *canonical model* for each isomorphic class; and mining data on such models.

3.3. Canonical Model – Unnamed Information Table

In Section 2, we observed that an attribute is a map, A^i: V $\rightarrow Dom(A^i)$. So one can consider an equivalence relation defined as follows:

$$v_1 \approx v_2 \text{ iff } A^i(v_1) = A^i(v_2)$$

The equivalence relation will be denoted by Q^i; note that each attribute value defines a granule (equivalence class). Hence A^i, as a mapping, can be factored through a projection

$$V \to V/Q^i.; v \to [v]_{Q^i}$$

and a naming map,

$$V/Q^i \to Dom(A^i); [v]_{Q^i}. \to NAME([v]_{Q^i}) = A^i(v)$$

where attribute value $A^i(v)$ is the name of a granule.

The attribute STATUS defines three granules by TEN, TWENTY and THIRTY; see Table 4. The 4 attributes in Table 1 define 4 equivalence relations (but only 3 distinct ones, Q^1, Q^2, Q^3). We will regard an attribute value as the name of a granule. They are summarized in Table 4.

Table 4. The 4 Partitions Induced from Table 1

line	Equiv. Classes	Attri value	Attribute	Equiv rel
1	*		S#	Identity
2	$\{v_1, v_4\}$	TWENTY	STATUS	Q^1
3	$\{v_2,v_3,v_4,v_5,v_6\}$	TEN		
4	$\{v_8,v_9\}$	THIRTY		
5	$\{v_1\}$	C_1	CITY	Q^2
6	$\{v_2,v_3,v_4,v_5,v_6\}$	C_2		
7	$\{v_7,v_8,v_9\}$	C_3		
8	$\{v_1, v_4\}$	1	RANK	Q^3
9	$\{v_2,v_3,v_4,v_5,v_6\}$	3		
10	$\{v_8,v_9\}$	1		

In Table 4, Line 1 illustrates the identity equivalence relation defined by S#, Line 2-3 is the equivalence relation Q^1 of STATUS, Line5-7, Q^2 of CITY, and Line 8-9, Q^3 of RANK. Note that RANK and STATUS are isomorphic and define the same equivalence relation ($Q^1 = Q^3$). Each attribute value, for example, say TWENTY, in STATUS is mapped to 2 in RANK by the isomorphism. Both TWENTY and 2 define the same equivalence class. This leads us to consider using granules as attribute values. Table 2 and 3 are transformed to the same Table 5, *the unnamed information table of K*.

Table 5. Unnamed Information Table of Table 1

V		Identity	**Q2**	**Q3**
v_1	$\to$	$(\{v_1\}$	$\{v_1,v_7\}$	$\{v_1\})$
v_2	$\to$	$(\{v_2\}$	$\{v_2,v_3,v_4,v_5,v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\})$
v_3	$\to$	$(\{v_3\}$	$\{v_2,v_3,v_4,v_5,v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\})$
v_4	$\to$	$(\{v_4\}$	$\{v_2,v_3,v_4,v_5,v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\})$
v_5	$\to$	$(\{v_5\}$	$\{v_2,v_3,v_4,v_5,v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\})$
v_6	$\to$	$(\{v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\}$	$\{v_2,v_3,v_4,v_5,v_6\})$
v_7	$\to$	$(\{v_7\}$	$\{v_1,v_7\}$	$\{v_7,v_8,v_9\})$
v_8	$\to$	$(\{v_8\}$	$\{v_8,v_9\}$	$\{v_7,v_8,v_9\})$
v_9	$\to$	$(\{v_9\}$	$\{v_8,v_9\}$	$\{v_7,v_8,v_9\})$

Observed that isomorphic attributes, STATUS and RANK, induce the same equivalence relation. Hence an unnamed information table is a simplified information table. We will denote the unnamed information table by UK. As the corollary of Theorem 1, we have

Theorem 2. High frequency patterns of K can be generated from the patterns of UK via elementary operations.

This implies that to find all high frequency patterns of K, we only need to find the patterns on UK. Previously, we have shown that it is extremely fast in finding patterns (large itemsets) from UK using granular computing [1].

The unnamed information table consists of V and Q, where $Q = \{Q^1, Q^2, \ldots, Q^n\}$ be the set of distinct equivalence relations induced by $A = \{A^1, A^2, \ldots, A^m\}$.

Definition 4. The pair (V, Q) is called the Granular Data Model (GDM) of the given information table K.

Pawlak called (V, Q) an approximation space, if Q has only one equivalence relation, and in general a knowledge base; knowledge base often has different meaning, we use GDM.

4. ATTRIBUTE(FEATURE) COMPLETION

4.1. Granular Data Model (GDM)

Data mining has been treated as a set of *added* operations on the classical data model. So data mining uses the same terms as database. However, their semantics are very different. The semantics of database are expressed in the schema and enforced by the database system. So all instances conform the expressed semantics. For example, the (intensional) function dependency is obeyed by every instance. In contrast, the semantics of data instance cannot impose the full semantics of the schema. Since data mining works on data instance, it is more beneficial to use Granular Data Model (GDM) to express the data model. From Table 5, we observed that

- The role of attributes is played by equivalence relations on V
- The attribute values are played by granules (equivalence classes) and attribute domains by the quotient sets.

From Section 8,

- Intension/extension functional dependencies are played by refinements (called knowledge dependency) of equivalence relations.

4.1. Attribute Transformations

Given an attribute Y and a subset B={B^1, B^2, …, B^k} of A, where each B^i is some A^{j_i}. Y is said to be an attribute transformation of B, if there exist a map

$$f: Dom(B^1) \times Dom(B^2) \times \ldots \times Dom(B^k) \rightarrow Dom(Y).$$

In other word, Y is functionally depended on B = {B^1, B^2, …, B^k}. The details of f are display in Table 6.

Table 6 Attribute (feature) Transformations

V	→	(B^1	B^2		B^k	Y)
v_1	→	(b^1_1	b^2_1	...	b^k_1	$f_1=f(b^1_1,b^2_1,...,b^k_1)$)
v_2	→	(b^1_2	b^2_2	...	b^k_2	$f_2=f(b^1_2,b^2_2,...,b^k_2)$)
v_3	→	(b^1_3	b^2_3	...	b^k_3	$f_3=f(b^1_3,b^2_3,...,b^k_3)$)
	→			...		...
v_i	→	(b^1_i	b^2_i		b^k_i	$f_i=f(b^1_i,b^2_i,...,b^k_i)$)
	→			...		...

Proposition 2. Let Y be the attribute transformed from B ={B^1, B^2, …, B^k}. Then the equivalence relation Y_E, induced by Y, is coarser than the equivalence relation $Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}$, where each Q^{j_i} is induced from some A^{j_i}. (See Section 8)

4.3. Derived Attributes in Unnamed Tables

Let A ={A^1, A^2, …, A^m} be the attributes, and Q={Q^1, Q^2, …, Q^n} be the induced *distinct* equivalence relations. The power set 2^A of A forms a lattice, in fact a Boolean algebra, where join and meet operations are the intersection and union respectively. Please note the *twist* from the common usage. Let Π(V) be the lattice of all equivalence relations on V, where join is the intersection of equivalence relations and meet is the "union;" where the "union" is the smallest coarsening of its components. Lee called Π(V) the partition lattice of V [3], and observed that any subset of A induces an equivalence relation on V, in fact, he stated that

Proposition 3..There is a map

$$\theta: 2^A \rightarrow \Pi(V),$$

that respects the meet, but not the join, operations.

The image Imθ is called the *relation lattice* by T. T. Lee; we will denote it by L(Q), recall that Q is the set of equivalence relations induced by A. Lee noted that

1. The join in L(Q) is different from that of Π(V).
2. So L(Q) is a subset, but not a sublattice, of Π(V).

T.T. Lee focus his study on L(Q), and his second comment is the point of our departure. Let L(Q*) denote *the smallest lattice containing all coarsening of* L(Q). We call L(Q*) the lattice-in-partitions; in [4], we call it generalized relation lattice. Recall that K is the given relation and UK is the unnamed relation. Now we will state the main result of this paper.

Main Theorem. L(Q*) consists of all possible derived attributes of the unnamed relation, UK, of K

Proof: (1) The forward direction: First observe that

$$V/Q^{j_1} \times V/Q^{j_2} \times \ldots \times V/Q^{j_k} = V/(Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}).$$

Let P ∈ L(Q*), that is, P is an equivalence relation coarser than some $Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}$. Then the refinement implies a map on their respective quotient sets,

$$g: V/(Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}).) \rightarrow V/P$$

In standard notations,

$$g: Dom(Q^{j_1}) \times Dom(Q^{j_2}) \ldots Dom(Q^{j_k}).) \rightarrow Dom(P)$$

In functional notation

$$P = g(Q^{j_1}, Q^{j_2}, \ldots, Q^{j_k}).$$

So g, as a map between attributes, is an attribute transformation. Hence P is a derived attribute.

(2) The reverse direction: Let P be a derived attribute of UK. That is, P is an equivalence relation and there is an attribute transformation

$$f: Dom(Q^{j_1}) \times Dom(Q^{j_2}) \ldots Dom(Q^{j_k}).) \rightarrow Dom(P)$$

In terms of the notations of unnamed information tables,

$$f: V/Q^{j_1} \times V/Q^{j_2} \times \ldots \times V/Q^{j_k} \rightarrow V/P$$

Observe that $V/Q^{j_1} \times V/Q^{j_2} \times \ldots \times V/Q^{j_k} = V/Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}$, so the existence of f implies that P is coarser than $Q^{j_1} \cap Q^{j_2} \cap \ldots \cap Q^{j_k}$. By definition P is an element in L(Q*). Q.E.D

All possible partitions of a finite set is finite, Π(V) is a finite set.

Corollary. L(Q*) is a finite sublattice of Π(V).

L(Q*) is illustrated in Table 7 and 8.

Table 7. The attribute (feature) completion of K

$E^1 \cap E^2$	E^1	E^2	E^3	E^4	E^5
$\{u_1,u_4\}$	$\{u_1,u_4\}$	$\{u_1,u_4\}$	$\{u_1,u_2,u_3,u_4\}$	$\{u_1,u_4,u_5\}$	$\{u_1,u_2,u_3,u_4,u_5\}$
$\{u_2,u_3\}$	$\{u_2,u_3,u_5\}$	$\{u_2,u_3\}$	$\{u_2,u_3\}$	$\{u_2,u_3\}$	$\{u_1,u_2,u_3,u_4,u_5\}$

$\{u_2,u_3\}$	$\{u_2,u_3,u_5\}$	$\{u_2,u_3\}$	$\{u_2,u_3\}$	$\{u_2,u_3\}$	$\{u_1,u_2,u_3,u_4,u_5\}$
$\{u_1,u_4\}$	$\{u_1,u_4\}$	$\{u_1,u_4\}$	$\{u_1,u_2,u_3,u_4\}$	$\{u_1,u_4,u_5\}$	$\{u_1,u_2,u_3,u_4,u_5\}$
$\{u_5\}$	$\{u_2,u_3,u_5\}$	$\{u_5\}$	$\{u_5\}$	$\{u_1,u_4,u_5\}$	$\{u_1,u_2,u_3,u_4,u_5\}$

Table 8. The lattice-in-Partitions;
The partial ordering of the lattice is read from left to right (E^5 is the largest element)

E^2	E^1	E^5
	E^3	
	E^4	

5. POST PROOF DISCUSSIONS

Readers may wander how could such an "unthinkable" result has such a simple proof. The essence is on the notion of isomorphism. It removes all the semantics of individual relation away from the underlying data structures that the mining is really relied on.

To convince the readers, let us walk through a seemingly contradicting example. Let us consider a numerical table of degree 2 (two columns). Each tuple represents a point in the Euclidean plane. So a relation K represents a set of n points on this plane.

Let us consider the case that the coordinate axes rotate slowly around the origin. Intuitively, the rotation will produce infinitely many derived attributes (each new location of X-axis is a new X-coordinate). How could L(Q*) be finite?

Let L1, L2, , , , Lk be the lines determined by these points. The number k is finite, in fact, at most ${}_nC_2$ (collinear points may reduce this number). For simplicity, let us assume initially the axes do not parallel to any of these lines (This is possible, since there are only finitely many of them). Under such assumption, no two points have either the same X or Y coordinates. If they do, then the line passing through them will be parallel either to X or Y axis.

While the axes rotating, as long as they do not parallel to any Li, the changing numerical tables are isomorphic to each other. The only times we may have a new non-isomorphic table is the time one of the axes parallel to a Li. So there are at most finitely many non-isomorphic classes of the rotating numerical tables. We hope this provides a convincing argument.

6. CONCLUSIONS

In this paper, we successfully enumerate all possible derived attributes of a given relation. The results seem striking; however, they are of theoretical nature. Even though L(Q*) contains a complete list of all attributes, the number is insurmountably large. The exhaustive search of association rules on all those attributes are beyond current reach. However, by combining the classical techniques of feature selections [8, 9], we may reach new applications. Classical feature selection has focused on the original set of attributes, now with our result, it seems suggest that the domain of selection should be extended to this complete set of attributes. We will report such research in near future.

7. MOTIVATIONAL EXAMPLE

Example. The six attributes of Table 9 consists of the coefficients of the quadratic equations:

$$Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0.$$

Table 9 has no high frequency patterns (support ≥ 2). However, if we set two new derived attributes

$$\Delta = (B^2 - 4AC), \text{ and } SIGN = \text{sign}(\Delta.),$$

then we have patterns on the SIGN attributes and we will get the "theorem" of the classifications of conic sections (see the last column of Table 10), where Δ is the discriminant and SIGN is the "signs" of Δ., positive, negative and zero.

It is clear that without Δ and SIGN, we will not be able to classifies the conic sections. This mathematical example indicates that *some transformations* are needed in some data mining [7].

Table 9. Relation of quadratic equations

Eq#	A	B	C	D	E	F
1	9	-4	-72	0	8	176
2	9	0	16	1	0	-144

Table 10. Derived Attributes Classification of quadratic equations

Δ	**SIGN**	Interpretations
2608	positive	Ellipse
-576	negative	Hyperbola

3	73	72	52	30	-40	-75	-10000	negative	Hyperbola
4	2	-72	23	-80	-60	-125	5000	positive	Ellipse
5	0	0	1	0	4	0	0	Zero	Parabola
6	2	4	2	12	-1	39	0	Zero	Parabola

8. EQUIVALENCE RELATIONS

Let P and Q be two equivalence relations on V. Each partitions V into granules (equivalence classes). If every Q-granule is a union of P-granules, then we say that Q is *knowledge depended* (KD) on P, Q is coarse than P, or P is finer than Q [11]. It is easy to verify that the intersection $P \cap Q$ of two equivalence relations is another equivalence relation whose partition consists of all intersections of P- and Q-granules. More generally, the intersection of all the equivalence relations is another equivalence relation.

In relational databases, a functional dependency occurs when the values of a tuple on the set of attributes uniquely determine the values of another set of attributes. Formally, let B and C be two subsets of A. An extensional function dependency EFD: $X \rightarrow Y$ if for very X-value there is a uniquely determined Y-values in the relation instance R. An intensional function dependency FD: $X \rightarrow Y$ exists on the relation scheme $\underline{R}$, if FD is satisfied by all relation instances R of the scheme $\underline{R}$. In database community, FD always refers to intensional FD. One should note that at any given moment, a relation instance may satisfy some family of extensional functional dependencies EFDs, however, the same family may not be satisfied by other relation instances. The family that is satisfied by all the relation instances is the intensional functional dependency FD. In this paper, we will be interested in the extensional functional dependency, so the notation "$X \rightarrow Y$" is an EFD. As pointed out earlier that attributes induce equivalence relations. An EFD can be interpreted as a knowledge dependency.

Proposition. An EFD, $X \rightarrow Y$, between two sets of attributes X and Y is equivalent to a knowledge dependency (refinements) of the equivalence relations induced by the attributes X and Y.

9. REFERENCES

[1] R. Agrawal, T. Imielinski, and A. Swami, "Mining Association Rules Between Sets of Items in Large Databases," in Proceeding of ACM-SIGMOD international Conference on Management of Data, pp. 207-216, Washington, DC, June, 1993

[2] Herbert B, Enderton, A mathamtical Introduction to logic, Academic Press, 1972.

[3] T. T. Lee, "Algebraic Theory of Relational Databases," The Bell System Technical Journal Vol 62, No 10, December, 1983, pp.3159-3204

[4] T. Y. Lin "The Lattice Structure of Database and Mining Multiple Level Rules." Presented in the Workshop on Data Mining and E-organizations, COMPSAC 2001, Chicago, Oct 8-12, 2001. it will appear as "Feature Transformations and Structure of Attributes" In: Data Mining and Knowledge Discovery: Theory, Tools, and Technology IV, Proceeding of SPIE's aeroSence 2002 1-5 April 2002 Orlando, FL.

[5] T. Y. Lin, ``Data Mining and Machine Oriented Modeling: A Granular Computing Approach," Journal of Applied Intelligence, Kluwer, Vol. 13, No 2, September/October,2000, pp.113-124.

[6] T. Y. Lin, "Data Mining: Granular Computing Approach." In: Methodologies for Knowledge Discovery and Data Mining, Lecture Notes in Artificial Intelligence 1574, Third Pacific-Asia Conference, Beijing, April 26-28, 1999, 24-33.

[7] T. Y. Lin and J. Tremba, "Attribute Transformations on Numerical Databases," Lecture Notes in Artificial Intelligence 1805, Terano, Liu, Chen (eds), PAKDD2000, Kyoto, Japan, April 18-20, 2000, 181-192.

[8] H. Liu and H. Motoda, ``Feature Transformaion and Subset Selection," IEEE Intelligent Systems, Vol. 13, No. 2, March/April, pp.26-28 (1998)

[9] H. Liu and H. Motoda (eds), Feature Extraction, Construction and Selection - A Data Mining Perspective, Kluwer Academic Pubihsers (1998).

[10] E. Louie and T. Y. Lin, "Finding Association Rules using Fast Bit Computation: Machine-Oriented Modeling," in: Foundations of Intelligent Systems, Z. Ras and S. Ohsuga (eds), Lecture Notes in Artificial Intelligence #1932, Springer-Verlag, 2000, pp. 486- 494. (12th International symposium on methodologies for Intelligent Systems, Charlotte, NC, Oct 11-14, 2000)

[11]. Z. Pawlak, Rough sets. Theoretical Aspects of Reasoning about Data, Kluwer Academic Publishers, 1991

Tsau Young Lin, more commonly T. Y. Lin, received his Ph. D from Yale University. Now he is a professor at the Department of Mathematics and Computer Science, San Jose State University. He has served as the (co-)chairs of conferences, and chief/associate editors, advisory, editorial /advisory/review board of journals. His interests include approximate retrievals, data mining, data security, intelligent control, Petri nets (as automata), and novel computing methodologies (fuzzy, granular and rough computing).

Granular Language and Its Deductive Reasoning[1]*

Qing Liu
Department of Computer Science & Engineering, Nanchang University, Nanchang 330029
E-mail: qliu@263.net

Abstract

In the paper discusses many real examples on information granules. To propose a deductive reasoning system corresponding to the fields according to various real examples. It can finish the deductive reasoning based on granular computing in the system. Objects of the reasoning are a binary pair with two elements. First element is an assertion and second is a semantic set corresponding to the assertion, so the reasoning is both in logic and in set theory. So-called logic means that the reasoning obeys the syntax in logical language; So called set theory means that the operations corresponding to semantic sets of logical formulas obey the methods in set theory, such that one may observe the evolution process of the formula operations. Hence, the reasoning is transparent and generalized. We define also the truth values and its computing of formulas (sentences) in granular language.

1. Introduction

Information granules are that a great complex information is divided into many blocks or many classes or many groups or crowds according to feature and properties of complex information in reasoning and making decision of people's. By granule we handle a great complex information, the process is called information granule. For example, information granules of car parking are that car parking is divided into several blocks according to properties or type or size or trademark of cars, a class car of having same properties or type or size or trademark is put in each block. Granules are important in many fields, such as interval analysis, data reasoning, the whole decomposition, rough set theory, evidence theory of DS, learning from examples, qualitative analysis theory, decision tree, semantic net, restrained programming, image cut and entities classification, etc..

Let IS=(U, A)be an information system, (a,v) or a_v is defined as a descriptor in IS, where $a\in$A is an attribute on set A of attributes, v is a value of attribute a with respect to object $x\in$U, namely $v=a(x)$. Thus (a,v) or a_v is used as an atom in Rough Logic[1]. We combine the (a,v) or a_v with ordinary logical connectives $\sim$, $\vee$, $\wedge$, $\rightarrow$ and $\leftrightarrow$, to have the well-formed formulas in rough logic. Let φ be the formula, $m(\varphi)_{IS}=\{x\in U: x \mid\approx_{IS} \varphi\}$ is called as a mean set of formula φ in IS, that is a set of the objects on U satisfying φ, where m is a symbol of mean function. Hence, $(\varphi, m(\varphi)_{IS})$ is an elementary granule in IS. Elementary granule of atomic formula a_v is written by$(a_v, m(a_v)_{IS})$. Obviously, $m(\varphi)_{IS}=\phi$ is considered φ to be false in IS; $m(\varphi)_{IS}=U$ is considered φ to be true in IS; When $\phi\subseteq m(\varphi)_{IS}\subseteq U$, φ is satisfiable in IS. If φ is undefinable on any subset $X\subseteq U$, then $m(\varphi)_{IS}$ is no observable in IS [2], then we may place $m(\varphi)_{IS}$ by the lower approximation$(B_*\varphi, B_*(m(\varphi)_{IS}))$ and upper approximation $(B^*\varphi, B^*(m(\varphi)_{IS}))$, they are also an elementary granule, where $B\subseteq A$ is a subset on A . If $m_*(\varphi)_{IS}=\{x\in U: x \mid\approx_{IS*} \varphi\}= B_*(m(\varphi)_{IS})$, where m_* is a symbol of rough lower semantic function of formula φ, then we call φ rough lower true in IS; If $m_*(\varphi)_{IS}=\{x\in U: x \mid\approx_{IS*} \varphi\}=\phi$, then we call φ rough lower false in IS; If $m^*(\varphi)_{IS}=\{x\in U: x \mid\approx_{IS}^* \varphi\}= B^*(m(\varphi)_{IS})$, where m^* is a symbol of rough upper semantic function of formula φ, then we call φ rough upper true in IS; If $m^*(\varphi)_{IS}=\{x\in U: x \mid\approx_{IS}^* \varphi\}=\phi$, then we call φ rough upper false in IS; $\phi\subset m_*(\varphi)_{IS}\subset B_*(m(\varphi)_{IS})\subset B^*(m(\varphi)_{IS})\subset U$, then φ is rough satisfiable in IS. Therefore the reasoning in the paper is the idea based on the information granules and granular computing.

2. Generalization of Granules [3,4]

2.1. Granules on Decision Algorithm

Decision algorithm is a set of decision rules, so elementary granule corresponding to decision algorithm is a set of elementary granules corresponding to decision rules. Let $G=\{g_1,\ldots,g_k\}$ is a set of elementary granules, where g_i is an elementary granule corresponding to a rule (φ_i,ψ_i), leading to the granule of set G from a rule (φ_i,ψ_i) being $(G, m(G))=\cup_{i=1}^{k}\{(g_i, m(g_i))\}$.

Decision Table 1 DT=(U,A∪{*d*})

*This study is support by SNSF（#60173054）and NSF（#9911027） in Jiangxi province in China .

A∪{*d*} / N	*a*	*b*	*c*	*d*
1	*n*(5)	*n*(4)	*n*(0)	*n*(1)
2	*n*(3)	*n*(4)	*n*(0)	*n*(0)
3	*n*(3)	*n*(4)	*n*(0)	*n*(2)
4	*n*(0)	*n*(2)	*n*(0)	*n*(1)
5	*n*(3)	*n*(2)	*n*(1)	*n*(2)
6	*n*(5)	*n*(2)	*n*(0)	*n*(1)

2.2. Granules on Decision Algorithm

Decision algorithm is a set of decision rules, so elementary granule corresponding to decision algorithm is a set of elementary granules corresponding to decision rules. Let $G=\{g_1,\ldots,g_k\}$ is a set of elementary granules, where g_i is an elementary granule corresponding to a rule (φ_i,ψ_i), leading to the granule of set G from a rule (φ_i,ψ_i) being $(G, m(G))=\cup_{i=1}^{k}\{(g_i, m(g_i))\}$.

2.3. Granule on Binary Relation

Let IS=(U,A) be an information system, *R* is a binary relation on set of granules, $G=\{g_1,\ldots,g_k\}$, $\forall g_i\in G$, g_i has a *R* relation with $g_j \in G$, written by $(g_i ,g_j) \in R$, the elementary granule corresponding to it is defined as

$$m(g_i,R)= \{m(g_j): (g_i ,g_j) \in R\}$$

For example, Let U be a nonempty universe used as objects with granules. V is a data set. $B\subseteq V\times U$ is called as a binary relation, $\forall p\in V$, $N(p)=\{u: p\ B\ u\}$ is a neighborhood of point *p*. Thus the Granule of point *P* with respect to binary relation B is defined as

$$m(p,B)=\{m(u): (p, u) \in B\}$$

Such as again, a decision rule $\varphi\rightarrow\psi$ in IS, where φ has an implication relation "$\rightarrow$" with ψ, then a new information granule is defined as

$$m(\varphi,\rightarrow)=\{m(\psi): (\varphi,\psi)\in\rightarrow\}$$

2.4. Granules on Graph

Let binary pair (G,E) be a graph, where $G=\{g_1,\ldots,g_k\}$ is a finite set of granules, so the granules of graph G is defined as $m(G)=\{m(g): g\in G\}$; $E\in G\times G$ is a binary relation on G, to call also the set of sides, so the granules of graph E is defined as $m(E)=\{(m(g),m(g')): g, g' \in G\}$, hence the granule of graph (G,E) is written by ((G,E), m((G,E)))=((G,E),(m(G),m(E))).

2.5. Granules on Graph

Let φ be a logical combination of the form $a_{v,}$ so φ is an ordinary formula in rough logic defined in IS [1], all sub-formulas of φ are denoted by $APP(\varphi)=\{\varphi_1,\ldots,\varphi_k\}$, then we define $m(\varphi)$ with $m(\varphi_i)$, namely

$$(\varphi,m(\varphi))=(APP_B(\varphi),\{m(\varphi_i)_{IS}: \varphi_i\in APP_B(\varphi)\})$$

If $\varphi_i\in APP_B(\varphi)$ and φ_i is no observable in given $X\subseteq U$, then φ_i is moved into $B_*\ \varphi_i$ and $B^*\ \varphi_i$ to be observable, that is, placing $(\varphi_i ,m(\varphi_i)_{IS})$ by $(B_*\ \varphi_i , B_*\ (m(\varphi_I)))$ and $(B^*\ \varphi_i ,B^*\ (m(\varphi_I)))$.

3. Granular Language

Definition 1 (syntax) Let us denote the language consisted of elementary granules by LIS, its syntax is defined as follows recursively:

(1) The descriptors of form $((a,v),m(a,v))$ or $(a_v ,m(a_v))$ having from attribute subset $B\subseteq A$ are the formula in L_{IS};

(2) The structure of form$(B_*\varphi, m(B_*\varphi))$ and $(B^*\varphi, m(B^*\varphi))$, where φ is the logical combination of form $((a,v),m(a,v))$ or $(a_v ,m(a_v))$ having from set A of attributes, then the structure φ is a formula in L_{IS};

(3) If φ and ψ are a formula in L_{IS}, then$(\sim\varphi, m(\sim\varphi))$, $(\varphi\wedge\psi,m(\varphi\wedge\psi))$ and $(\varphi\vee\psi,m(\varphi\vee\psi))$ are also a formula in L_{IS};

(4) The formulas defined via finite quotation (1)-(3) are considered in L_{IS}.

Definition 2 (semantics)The semantics of formulas in language L_{IS} defined in information system IS=(U, S) are defined recursively by

(1) $m(a_v)=\{x\in U: a(x)=v\in V\}$, where V is the set of attribute values, *m* is a symbol of semantic function of formulas;

(2) $(\varphi,m(\sim\varphi))=U-(\varphi,m(\varphi)_{IS})$;

(3) $(\varphi\wedge\psi,m(\varphi\wedge\psi))=(\varphi,m(\varphi))\cap(\psi,m(\psi))$;

(4) $(\varphi\vee\psi,m(\varphi\vee\psi))=(\varphi,m(\varphi))\cup(\psi,m(\psi))$.

The formulas with connectives $\rightarrow$ and $\leftrightarrow$ can substitute by ~ and $\vee$ or $\wedge$.

We note that an information granule is a binary pair consisting of a logical formula φ and its semantic set $m(\varphi)$, written by $(S(\varphi), m(\varphi))$formally, where $S(\varphi)$ is called granular syntax of φ, $m(\varphi)$ is called granular semantics of φ; For symbol simplicity, we use second element $m(\varphi)$ usually, and first element $S(\varphi)$ is only used as a logical combination. For example, a elementary conjunction of $\varphi\in L_{IS}$ is denoted by $CNF_B(x)$, where $B\subseteq A$ is a subset of attribute set A, $x\in U$. If let $B=\{a, b\}$, $\varphi=CNF_{IS}(x)=a_1\wedge b_1$, then the elementary Granule of φ is denoted by $(\varphi,m(\varphi))=(a_1\wedge b_1 ,m(a_1 \wedge b_1))=(a_1\wedge b_1 ,m(a_1)\cap m(b_1))$.

4. Deductive Reasoning Based on Granular Computing

After introduction granular language the above, we have following several results:

Proposition 1

(1) $\forall\varphi,\psi\in L_{IS}$, if $\varphi=_G\psi$, then $(B_*\varphi, B_*(m(\varphi)))=(B_*\psi, B_*(m(\psi)))$;

(2) $\forall\varphi,\psi\in L_{IS}$, if $\varphi=_G\psi$, then $(B^*\varphi, B^*(m(\varphi)))=(B^*\psi, B^*(m(\psi)))$;

(3) $\forall\varphi,\psi\in L_{IS}$, if $\varphi=_G\psi$, then $(B_*(\varphi\wedge\zeta), B_*(m(\varphi\wedge\zeta)))=(B_*(\psi\wedge\zeta), B_*(m(\psi\wedge\zeta)))$;

(4) $\forall\varphi,\psi\in L_{IS}$, if $\varphi=_G\psi$, then $(B^*(\varphi\vee\zeta), B^*(m(\varphi\vee\zeta)))=(B^*(\psi\vee\zeta), B^*(m(\psi\vee\zeta)))$;

(5) $\forall\varphi,\psi\in L_{IS}$, if $(\varphi,m(\varphi)_{IS})\subseteq_G(\psi,m(\psi)_{IS})$, then $(\sim\psi,m(\sim\psi)_{IS})\subseteq_G(\sim\varphi,m(\sim\varphi)_{IS})$;

(6) $\forall\varphi,\psi\in L_{IS}$, if φ is true or rough true in IS, then the elementary granule $(\varphi,m(\varphi)_{IS})$ is interpreted as $m(\varphi)_{IS}$ being universe U or $m_*(\varphi)_{IS}$ being $B_*(m(\varphi)_{IS})$ and $m^*(\varphi)_{IS}$ being $B^*(m(\varphi)_{IS})$. Rule MP(Modus Pones) holds in granular language L_{IS}, denoted by GMP, namely

$$\text{GMP: } \frac{(\varphi, m(\varphi)_{IS}) \wedge (\varphi, m(\varphi)_{IS})\subseteq_G(\psi, m(\psi)_{IS})}{(\psi, m(\psi)_{IS})}$$

where $(\psi, m(\psi)_{IS})$ is interpreted as $m(\psi)_{IS}$ being universe U or $m_*(\varphi)_{IS}$ being $B_*(m(\varphi)_{IS})$ and $m^*(\varphi)_{IS}$ being $B^*(m(\varphi)_{IS})$, here "$=_G$" is a equal symbol of formulas in granular language L_{IS} [6].

We prove (3) in proposition 1, the proof is following.

Supposing $(B_*(\varphi\wedge\zeta), B_*(m(\varphi\wedge\zeta)_{IS}))\neq(B_*(\psi\wedge\zeta), B_*(m(\psi\wedge\zeta)_{IS}))$, then $\exists x\in U$, equivalent class of x with respect to B is written by $B(x)\subseteq (m(\varphi\wedge\zeta)_{IS})$, but $B(x)\not\subset m(\psi\wedge\zeta)_{IS})$. By the semantic definition we have $B(x)\subseteq m(\varphi)_{IS}\cap m(\zeta)_{IS}$, so $B(x)\subseteq m(\varphi)_{IS}\wedge B(x)\subseteq m(\zeta)_{IS}$. From $B(x)\not\subset m(\psi\wedge\zeta)_{IS})$, to obtain $B(x)\not\subset m(\varphi)_{IS}\cap m(\zeta)_{IS}$, that is $B(x)\not\subset m(\psi)_{IS}\vee B(x)\not\subset m(\zeta)_{IS}$. If $B(x)\not\subset m(\psi)_{IS}$, then $x\notin B_*(m(\psi)_{IS})$, but $x\in B_*(m(\varphi)_{IS})$, hence $(B_*\psi,B_*(m(\psi)_{IS}))\neq(B_*\varphi,B_*(m(\varphi)_{IS}))$, by result (1) to see $\varphi\neq_G\psi$, it is contrary with supposing precondition. Therefore, result(3) has been proved. Similarly, we can prove other results in proposition 1 .

Let L and H be the rough lower and upper approximate operators respectively[2]. The semantic function of formulas functioned by two operators is defined as follows:

Definition 3 $\forall\varphi\in L_{IS}$, then $(L\varphi,m(L\varphi))_{IS}$ and $(H\varphi,m(H\varphi))_{IS}$ are called interior greatest definable granule and exterior smallest definable granule of φ respectively, thus having

$$m(L\varphi)_{IS}=\{x\in U: x\mid\approx L\varphi\}$$

and

$$m(H\varphi)_{IS}=\{x\in U: x\mid\approx H\varphi\}$$

By the definition above, we have the granules of formulas with operators L and H.

Proposition 2

(1) $(L\varphi,m(L\varphi))_{IS}=_G(L\psi,m(L\psi))_{IS}$ iff $(B_*L\varphi, B_*(m(L\varphi)_{IS}))=(B_*L\psi, B_*(m(L\psi)_{IS}))\wedge (B^*L\varphi, B^*(m(L\varphi)_{IS}))= (B^*L\psi, B^*(m(L\psi)_{IS}))$;

(2) $(H\varphi,m(H\varphi))_{IS}=_G(H\psi,m(H\psi))_{IS}$ iff $(B_*H\varphi, B_*(m(H\varphi)_{IS}))= (B_*H\psi, B_*(m(H\psi)_{IS}))\wedge (B^*H\varphi, B^*(m(H\varphi)_{IS})) = (B^*H\psi, B^*(m(H\psi)_{IS}))$.

Let L_{IS} be a granular language, the calculus of formulas (sentances) in the language are a granular computing. And each granule is a binary pair, first element is a logical formula in L_{IS}, second element is a semantic set corresponding to the formula. We call the pair a granule, because the binary pair is a no separable entirety consisting of both syntax and semantics. Hence, the binary pair is both in logic and in set theory. Such that we can use both logical method and set theory method in deductive reasoning or in other approximate reasoning. This is the superiority using information granules and granular computing. We can prove following theorems based on granular computing.

Theorems

(1) $(LL\varphi,m(LL\varphi)_{IS})=_G(HL\varphi,m(HL\varphi)_{IS})$;

(2) $(LH\varphi,m(LH\varphi)_{IS})=_G(H\varphi,m(H\varphi)_{IS})$;

(3) $(HH\varphi,m(HH\varphi)_{IS})=_G(LH\varphi,m(LH\varphi)_{IS})$;

(4) $(HH\varphi,m(HH\varphi)_{IS})=_G(H\varphi,m(H\varphi)_{IS})$.

Where "$=_G$" is a equal symbol of formulas in granular language. The procedure of deductive proving(2) is following:

Proof of (2): We need only essentially to prove $(LH\varphi,B_*(m(LH\varphi)_{IS}))=(H\varphi,B_*(m(H\varphi)_{IS}))$ and $(LH\varphi,B^*(m(LH\varphi)_{IS}))=(H\varphi,B^*(m(H\varphi)_{IS}))$.

The deductive proof of first equal form is following:

① $(L\sim\varphi,B_*(m(L\sim\varphi)_{IS}))\subseteq(\sim\varphi,B_*(m(\sim\varphi)_{IS}))$, Properties of rough[6].

② $(\varphi,B_*(m(\varphi)_{IS}))\subseteq(H\varphi,B_*(m(H\varphi)_{IS}))$, changing position in ① , L is dual with H.

③ $(L\varphi,B_*(m(L\varphi)_{IS}))\subseteq(HL\varphi,B_*(m(HL\varphi)_{IS}))$, Placing φ by $L\varphi$ in ②.

④ $(HL\varphi,B_*(m(HL\varphi)_{IS}))\subseteq(L\varphi,B_*(m(L\varphi)_{IS}))$, Properties of rough[6].

⑤ $(HL\varphi,B_*(m(HL\varphi)_{IS}))=(L\varphi,B_*(m(L\varphi)_{IS}))$, ③ and ④。

⑥ $(HLH\varphi,B_*(m(HLH\varphi)_{IS}))=(LH\varphi,B_*(m(LH\varphi)_{IS}))$, Placing φ by $H\varphi$ in ⑤.

⑦ $(LH\varphi,B_*(m(LH\varphi)_{IS}))=(HH\varphi,B_*(m(HH\varphi)_{IS}))$, Properties of rough[6] .

⑧ $(HLH\varphi,B_*(m(HLH\varphi)_{IS}))=(HH\varphi,B_*(m(HH\varphi)_{IS}))$, Three segment theory in ⑥ and ⑦.

⑨ $(HH\varphi,B_*(m(HH\varphi)_{IS}))=(H\varphi,B_*(m(H\varphi)_{IS}))$, Property of rough sets had been proved[6].

⑩ $(LH\varphi,B_*(m(LH\varphi)_{IS}))=(H\varphi,B_*(m(H\varphi)_{IS}))$, Three segment theory in ⑦ and ⑨.

Similarly, we can prove second equal form, thus theorems (2) had been proved. By similar methods we may prove other results in the theorems.

5. Conclusion

It discusses some basic concepts of information granules in the paper. From the list of many real examples of information granules, it may see that an information granule is essentially a binary pair. First element is an assertion and second element is the semantic set corresponding to the assertion. Hence, the binary pair is consisted of both logic and set, and the pair is no decomposable. Therefore, we call it a condensing entirety. Here we discuss a method of extracting relative pattern by tuning parameters from information granule. Namely the parameters in information granules are taken from various information source. These patterns can be consisted of the parameters. These patterns can be used as inference rules in approximate reasoning. Therefore, we call the reasoning establishing in basis of granular computing. By the list of related real examples we explain that the patterns and methods based on information granules and granular computing have a special superiority and fashion in reasoning., namely the objects used as reasoning are a binary pair consisting of both logic and set. So we may use both logical method and set theory method in reasoning. Further research is to develop new search for technology, extracting information structure pattern from different information source and background knowledge. This study will discovery more suitable and valuable inference rules by granular computing high-speed. We are finishing the research of granular language and its deductive reasoning.

10. References

[1] Q.Liu, S.H.Liu and F.Zheng, Rough Logic and Its Applications in Data Reduction, Journal of Software (in Chinese), Vol.12, No.3, 2001,3, 415-419.

[2] T.Y.Lin and Q.Liu, First-Order Rough Logic 1: Approximate Reasoning Via Rough Sets, Fundamenta Informaticae, Vol.27, No.2-3, Aug.1996, 137-154.

[3] A.Skowron, Toward Intelligent Systems: Calculi of Information Granules, Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC-2001)-Bulletin of International Rough Set Society Vol.5, No.1/2, May 20-22,2001,9-30.

[4] A. Skowron, J. Stepaniuk. and James. F. Peters, Extracting Patterns Using Information Granules, Proceedings of International Workshop on Rough Set Theory and Granular Computing (RSTGC-2001)-Bulletin of International Rough Set Society Vol.5, No.1/2, May 20-22,2001, 135-142.

[5] Q. Liu, Neighborhood Logic and Its Data Reasoning in Information Table of Neighborhood Values, Chinese Journal of Computers, Vol.24, No.4, 2001,4,405-410.

[6] Q. Liu, Rough Sets and Rough Reasoning (in Chinese), Published by Science Press, Beijing, 2001, 8.

Feature Selection, Extraction and Construction

Hiroshi Motoda

Inst. of Sci. & Indus. Res.
Osaka University
Ibaraki, Osaka, Japan 567-0047

Huan Liu

Dept. of CS & Eng.
Arizona State University
Tempe, AZ, USA 85287-5406

Abstract

Feature selection is a process that chooses a subset of features from the original features so that the feature space is optimally reduced according to a certain criterion. Feature extraction/construction is a process through which a set of new features is created. They are used either in isolation or in combination. All attempt to improve performance such as estimated accuracy, visualization and comprehensibility of learned knowledge. Basic approaches to these three are reviewed giving pointers to references for further studies.

1 Introduction

Researchers and practitioners realize that an important part of data mining is pre-processing in which data is processed before it is presented to any learning, discovering, or visualizing algorithm [23, 11]. Feature extraction, selection and construction is one effective approach to data reduction among others such as instance selection [24], data selection [25]. The goal of feature extraction, selection and construction is three fold: 1) reducing the amount of data; 2) focusing on the relevant data; and 3) improving the quality of data and hence the performance of data mining algorithms, such as learning time, predictive accuracy. There could be two main approaches. One is to rely on data mining algorithms and the other is to conduct preprocessing before data mining. It seems natural to let data mining algorithms deal with data directly as the ultimate goal of data mining is to find hidden patterns from data. Indeed, many data mining methods attempt to select, extract, or construct features, however, both theoretical analyses and experimental studies indicate that many algorithms scale poorly in domains with large numbers of irrelevant and/or redundant features [17].

The other approach is to preprocess the data so that it is made suitable for data mining. Feature selection, extraction and construction are normally tasks of preprocessing, and are independent of data mining. First, it can be done once and used for all subsequent data mining tasks. Second, it usually employs a less expensive evaluation measure than using a data mining algorithm. Hence, it can handle larger sized data than data mining can. Third, it often works off-line. Therefore, if necessary, many different algorithms can be tried. However, in addition to their being mainly preprocessing tasks, there are other commonalities among them: 1) they try to achieve the same goal for data reduction, 2) they require some criteria to make sure that the resulted data allows data mining algorithm to accomplish nothing less, if not more, and 3) their effectiveness has to be measured in multiple aspects such as reduced amounts of data, relevance of the reduced data, mostly, if possible, their direct impact on data mining algorithms. Feature selection (FS), extraction (FE) and construction (FC) can be used in combination. In many cases, feature construction expands the number of features with newly constructed ones that are more expressive but they may include useless features. Feature selection can help automatically reduce those excessive features.

2 Feature Selection

2.1 Concept

Feature selection is a process that chooses a subset of M features from the original set of N features ($M \leq N$), so that the feature space is optimally reduced according to a certain criterion [3, 5]. According

to [21], the role of feature selection in machine learning is 1) to reduce the dimensionality of feature space, 2) to speed up a learning algorithm, 3) to improve the predictive accuracy of a classification algorithm, and 4) to improve the comprehensibility of the learning results. Recent study about feature selection in unsupervised learning context shows that feature selection can also help to improve the performance of clustering algorithms with reduced feature space [31, 32, 7, 6, 14]. In general, feature selection is a search problem according to some evaluation criterion.

Feature subset generation One intuitive way is to generate subsets of features sequentially. If we start with an empty subset and gradually add one feature at a time, we adopt a scheme called *sequential forward selection*; if we start with a full set and remove one feature at a time, we have a scheme called *sequential backward selection*. We can also *randomly* generate a subset so that each possible subset (in total, 2^N, where N is the number of features) has an approximately equal chance to be generated. One extreme way is to *exhaustively* enumerate 2^N possible subsets.

Feature evaluation An optimal subset is always relative to a certain evaluation criterion (*i.e.* an optimal subset chosen using one evaluation criterion may not be the same as that using another evaluation criterion). Evaluation criteria can be broadly categorized into two groups based on their dependence on the learning algorithm applied on the selected feature subset. Typically, an independent criterion (i.e. filter) tries to evaluate the goodness of a feature or feature subset without the involvement of a learning algorithm in this process. Some of the independent criteria are distance measure, information measure, dependency measure, consistency measure [LM98b]. A dependent criterion (i.e. wrapper) tries to evaluate the goodness of a feature or feature subset by evaluating the performance of the learning algorithm applied on the selected subset. In other words, it is the same measure on the performance of the applied learning algorithm. For supervised learning, the primary goal of classification is to maximize predictive accuracy, therefore, predictive accuracy is generally accepted and widely used as the primary measure by researchers and practitioners. While for unsupervised learning, there exist a number of heuristic criteria for estimating the quality of clustering results, such as cluster compactness, scatter separability, and maximum likelihood. Recent reviews on developing dependent evaluation criteria for unsupervised feature selection based on these criteria can be found in [Tal99b, DB00, KSM00].

2.2 Algorithms

Many feature selection algorithms exist. Using the general model described earlier, we can regenerate these existing algorithms by having proper combinations for each component.

Exhaustive/complete approaches Focus [1, 2] applies an inconsistency measure and exhaustively evaluates subsets starting from subsets with one feature (i.e., sequential forward search); Branch-and-Bound [27, 30] evaluates estimated accuracy, and ABB [22] checks an inconsistency measure that is monotonic. Both start with a full feature set until the preset bound cannot be maintained.

Heuristic approaches SFS (sequential forward search) and SBS (sequential backward search) [30, 5, 3] can apply any of five measures. DTM [4] is the simplest version of a wrapper model - just learn a classifier once and use whatever features found in the classifier.

Nondeterministic approaches LVF [18] and LVW [19] randomly generate feature subsets but test them differently: LVF applies an inconsistency measure, LVW uses accuracy estimated by a classifier. Genetic Algorithms and Simulated Annealing are also used in feature selection [30, 13]. The former may produce multiple subsets, the latter produces a single subset.

Instance-based approaches Relief [15, 16] is a typical example for this category. There is no explicit procedure for feature subset generation. Many small data samples are sampled from the data. Features are weighted according to their roles in differentiating instances of different classes for a data sample. Features with higher weights can be selected.

3 Feature Extraction

3.1 Concepts

Feature extraction is a process that extracts a set of new features from the original features through some functional mapping [35]. Assuming there are n features (or attributes) $A_1, A_2, ..., A_n$, after feature extraction, we have another set of new features $B_1, B_2, ..., B_m (m < n), B_i = F_i(A_1, A_2, ..., A_n)$, and F_i is a mapping function. Intensive search is generally

required in finding good transformations. The goal of feature extraction is to search for a minimum set of new features via some transformation according to some performance measure. The major research issues can therefore be summarized as follows.

Performance Measure It investigates what is the most suitable in evaluating extracted features. For a task of classification, the data has class labels and predictive accuracy might be used to determine what is a set of extracted features. When it is of clustering, the data does not have class labels and one has to resort to other measures such as inter-cluster/intra-cluster similarity, variance among data, etc.

Transformation It studies ways of mapping original attributes to new features. Different mappings can be employed to extract features. In general, the mappings can be categorized into linear or nonlinear transformations. One could categorize transformations along two dimensions: linear and labeled, linear and non-labeled, nonlinear and labeled, nonlinear and non-labeled. Many data mining techniques can be used in transformation such as EM, k-Means, k-Medoids, Multi-layer Perceptrons, etc [12].

Number of new features It surveys methods that determine the minimum number of new features. With our objective to create a minimum set of new features, the real question is how many new features can ensure that "the true nature" of the data remains after transformation.

One can take advantage of data characteristics as a critical constraint in selecting performance measure, number of new features, and transformation. In addition to with/without class labels, data attributes can be of various types: continuous, nominal, binary, mixed. Feature extraction can find its many usages: dimensionality reduction for further processing [23], visualization [8], compound features used to booster some data mining algorithms [20].

3.2 Algorithms

The functional mapping can be realized in several ways. We present here two exemplar algorithms to illustrate how they treat different aspects of feature extraction.

A feedforward neural networks approach A single hidden layer multilayer perceptron can be used to to extract new features [29]. The basic idea is to use the hidden units as newly extracted features. The predictive accuracy is estimated and used as the performance measure. This entails that data should be labeled with classes. The transformation from input units to hidden units is non-linear. Two algorithms are designed to construct a network with the minimum number of hidden units and the minimum of connections between the input and hidden layers: the network construction algorithm parsimoniously adds one more hidden unit to improve predictive accuracy; and the network pruning algorithm generously removes redundant connections between the input and hidden layers if predictive accuracy does not deteriorate.

Principal Component Analysis PCA is a classic technique in which the original n attributes are replaced by another set of m new features that are formed from linear combinations of the original attributes. The basic idea is straightforward: to form an m-dimensional projection ($1 \leq m \leq n-1$) by those linear combinations that maximize the sample variance subject to being uncorrelated with all these already selected linear combinations. Performance measure is sample variance; the number of new features, m, is determined by the m principal components that capture the amount of variance subject to a pre-determined threshold; and the transformation is linear combination. PCA does not require that data be labeled with classes. The search for m principal components can be rephrased to finding m eigenvectors associated with the m largest eigenvalues of the covariance matrix of a data set [12].

4 Feature Construction

4.1 Concepts

Feature construction is a process that discovers missing information about the relationships between features and augments the space of features by inferring or creating additional features [26, 34, 23]. Assuming there are n features $A_1, A_2, ..., A_n$, after feature construction, we may have additional m features $A_{n+1}, A_{n+2}, ..., A_{n+m}$. All new constructed features are defined in terms of original features, as such, no inherently new informed is added through feature construction. Feature construction attempts to increase the expressive power of the original features. Usually, the dimensionality of the new feature set is expanded and is bigger than that of the original feature set. Intuitively, there could be exponentially many

combinations of original features, and not all combinations are necessary and useful. Feature construction aims to automatically transform the original representation space to a new one that can help better achieve data mining objectives: improved accuracy, easy comprehensibility, truthful clusters, revealing hidden patterns, etc. Therefore, the major research issues of feature construction are the following four.

How to construct new features Various approaches can be categorized into four groups: data-driven, hypothesis-driven, knowledge-based, and hybrid [34, 23]. The data-driven approach is to construct new features based on analysis of the available data by applying various operators. The hypothesis-driven approach is to construct new features based on the hypotheses generated previously. The knowledge-based is to construct new features applying existing knowledge and domain knowledge.

How to choose and design operators for feature construction There are many operators for combining features to form compound features [23]. Conjunction, disjunction and negation are commonly used constructive operators for nominal features. Other common operators are M-of-N and X-of-N [36], where M-of-N is true iff at least M out of N conditions are true, and X-of-N X iff X of N conditions are true; cartesian product [28] of two or more nominal features. For numerical features, simple algebraic operators such as equivalence, inequality, addition, subtraction, multiplication, division, maximum, minimum, average are often used to construct compound features.

How to use operators to construct new features efficiently It is impossible to exhaustively explore every possible operator. It is, thus, imperative to find intelligent methods that can avoid exhaustive search and heuristically try potentially useful operators. This line of research investigates the connections between data mining tasks, data characteristics, and operators that could be effective.

How to measure and select useful new features Not all constructed features are good ones. We have to be selective. One option is to handle the selection part by applying feature selection techniques to remove redundant and irrelevant features. When the number of features is very large, it is sensible to make decision while a new compound feature is generated to avoid too many features. This would require an effective measure to evaluate a new feature and provide an indicator. Researchers are investigating various measures that are not computationally expensive. Some examples are measures of consistency, distance as used in feature selection [21].

4.2 Algorithms

Feature construction can be realized in several ways. We show here two exemplar algorithms to illustrate how new features are constructed and built into an induction model. Many examples can be found in [23].

Greedy search for use in decision tree nodes A straightforward algorithm is to use a greedy search. In case of a decision tree induction, the algorithm generates at each decision node one new feature based on both original features and those already constructed and select the best one. To construct a new feature, the algorithm performs a greedy search in the instance space using a prespecified set of constructive operators. The search starts from an empty set. At each search step, it either adds one possible feature-value pair or deletes one possible feature-value pair in a systematic manner. An evaluation function that takes both class entropy and model complexity into account can be used. Optimal M-of-N and X-of-N can be found in this manner [36]. A variant of this which is useful for numeric attributes is to search for the best linear discriminant function [9] and its extension to functional trees [10].

Genetic algorithm for use in wrapper mode Genetic algorithms are adaptive search techniques for evolutional optimization. Each individual is evaluated based on its overall fitness to the given application domain. New individuals are constructed from their parents by two operators: mutation and crossover, as well as copy. An individual is represented by a variable-length nested list structure comprising a set of original and compound features. For continuous features, we can use a set of arithmetic operators such as +, , *, /. One good application is image classification (eye detection in pictures) [33] in which both feature construction and feature selection are interleaved and C4.5 is used to return a fitness value.

5 Conclusions

Feature extraction and construction are the variants of feature transformation through which a new

set of features is created. Feature construction often expands the feature space, whereas feature extraction usually reduces the feature space. Feature transformation and feature selection are not two totally independent issues. They can be viewed as two sides of the representation problem. We can consider features as a representation language. In some cases where this language contains more features than necessary, feature selection helps simplify the language; in other cases where this language is not sufficient to describe the problem, feature construction help enrich the language by constructing compound features. The use of feature selection, extraction and construction depends on the purpose for simpler concept description or for better data mining task performance.

Despite of recent advancement of feature selection, extraction and construction, much work is needed to unify this currently still diversified field. Many types of data exist in practice. Boolean, nominal, and numerical types are popular, but others like structural, relational, temporal should also receive our equal attention in data mining applications of real world problems. Feature selection, extraction and construction are key techniques in answering the pressing needs for data mining. These techniques can help reduce data for mining or learning tasks and enable those mining algorithms, which were unable to mine, to mine.

Acknowledgements

This work was partially supported by the grant-in-aid for scientific research on priority area "Active Mining" funded by the Japanese Ministry of Education, Culture, Sports, Science and Technology for H. Motoda and by the National Science Foundation under Grant No. IIS-0127815 for H. Liu.

References

[1] H. Almuallim and T. Dietterich, "Learning with Many Irrelevant Features", *Proc. of the Ninth National Conference on Artificial Intelligence*, pp. 547–552, 1991

[2] H. Almuallim and T. Dietterich, "Learning Boolean Concepts in the Presence of Many Irrelevant Features", *Artificial Intelligence* 69, 1-2, pp. 279–305, 1994.

[3] A.L. Blum and P. Langley, "Selection of Relevant Features and Examples in Machine Learning", *Artificial Intelligence*, 97, pp. 245–271, 1997.

[4] C. Cardie, "Using Decision Trees to Improve Case-Based Learning", *Proc of the Tenth International Conference on Machine Learning*, pp. 25–32, 1993.

[5] M. Dash and H. Liu, "Feature Selection Methods for Classifications", *Intelligent Data Analysis: An International Journal*, 1, 3, 1997. http://www-east.elsevier.com/ida/free.htm.

[6] M. Dash and H. Liu,"Feature Selection for Clustering", *Proc. of the Fourth Pacific Asia Conference on Knowledge Discovery and Data Mining, (PAKDD-2000)*, Springer Verlag, pp. 110–121, 2000.

[7] J. G. Dy and C. E. Brodley, "Feature Subset Selection and Order Identification for Unsupervised Learning. *Proc. of the Seventeenth International Conference on Machine Learning*, pp. 247–254, 2000.

[8] U. Fayyad, G.G. Grinstein, and A. Wierse, *Information Visualization in Data Mining and Knowledge Discovery*, Morgan Kaufmann Publishers, 2001.

[9] J. Gama and P. Brazdil, "Constructive Induction on Continuous Spaces", chapter 18, pp. 289–303. In [23], 1998. 2nd Printing, 2001.

[10] J. Gama, "Functional Trees", em Proc. of the Fourth International Conference on Discovery Science, pp. 58–73, 2001.

[11] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*, Morgan Kaufman, 2001.

[12] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, A Bradford Book The MIT press, 2001.

[13] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance", *IEEE Trans. on Pattern Analysis and Machine Intelligence* 19, 2, pp. 153–158, 1997.

[14] Y. Kim, W. Street, and F. Menczer, "Feature Selection for Unsupervised Learning via Evolutionary Search", *Proc. of the sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 365–369, 2000.

[15] K. Kira and L. Rendell, "The feature selection problem: Traditional methods and a new algorithm", *Proc. of the Tenth National Conference on Artificial Intelligence*, pp. 129–134, 1992.

[16] I. Kononenko, "Estimating attributes : Analysis and extension of RELIEF", *Proceedings of the European Conference on Machine Learning*, pp. 171–182, 1994.

[17] P. Langley, *Elements of Machine Learning*, Morgan Kaufmann, 1996.

[18] H. Liu, and R. Setiono, R, "A Probabilistic Approach to Feature Selection - A Filter Solution", *Proc. of the International Conference on Machine Learning (ICML-96)*, pp. 319–327, 1996.

[19] H. Liu and R. Setiono, "Feature Selection and Classification - A Probabilistic Wrapper Approach", *Proc. of the Ninth International Conference on Industrial and Engineering Applications of AI and ES*, pp. 419–424, 1996.

[20] H. Liu and R. Setiono, "Feature Transformation and Multivariate Decision Tree Induction", *Proc. of the First International Conference on Discovery Science (DS'98)*, Springer Verlag, pp. 279–290, 1998.

[21] H. Liu and H. Motoda, *Feature Selection for Knowledge Discovery Data Mining*, Boston: Kluwer Academic Publishers, 1998.

[22] H. Liu, H. Motoda and M. Dash, "A Monotonic Measure for Optimal Feature Selection", *Proc. of the European Conference on Machine Learning*, pp. 101–106, 1998

[23] H. Liu and H. Motoda, editors, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Boston: Kluwer Academic Publishers, 1998. 2nd Printing, 2001.

[24] H. Liu and H. Motoda, editors, *Instance Selection and Construction for Data Mining*, Boston: Kluwer Academic Publishers, 2001.

[25] H. Liu, H. Lu, and J. Yao, "Toward Multidatabase Mining: Identifying Relevant Databases", *IEEE Transactions on Knowledge and Data Engineering*, 13, 4, pp. 541–553, 2001.

[26] C.J. Matheus, "The Need for Constructive Induction", *Proc. of the Eighth International Workshop on Machine Learning*, pp. 173–177, 1991.

[27] P. Narendra and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection", *IEEE Trans. on Computer*, C-26, 9, pp. 917–922, 1977.

[28] M.J. Pazzani, "Constructive Induction of Cartesian Product Attributes", chapter 21, pp. 341–354. In [23], 1998. 2nd Printing, 2001.

[29] R. Setiono and H. Liu, "Feature Extraction via Neural Networks", chapter 12, pp. 191–204. In [23], 1998. 2nd Printing, 2001.

[30] W. Siedlecki and J. Sklansky, "On Automatic Feature Selection", *International Journal of Pattern Recognition and Artificial Intelligence*, 2, pp. 197–220, 1988

[31] L. Talavera,"Feature Selection as A Preprocessing Step for Hierarchical Clustering", *Proc. of the Sixteenth International Conference on Machine Learning*, pp. 389–397, 1999.

[32] L. Talavera, "Feature Selection as Retrospective Pruning in Hierarchical Clustering", *Proc. of the Third Symposium on Intelligent Data Analysis (IDA '99)*, pp. 75–86, 1999.

[33] H. Vafaie and K. De Jong, "Evolutionary Feature Space Transformation", pp. 307–323. In [23], 1998. 2nd Printing, 2001.

[34] J. Wnek and R.S. Michalski, "Hypothesis-Driven Constructive Induction in AQ17-HCI: A Method and Experiments", *Machine Learning*, 14, pp. 139–168, 1994.

[35] N. Wyse, R. Dubes, and A.K. Jain, "A critical evaluation of intrinsic dimensionality algorithms", In E.S. Gelsema and L.N. Kanal, editors, *Pattern Recognition in Practice*, pp. 415–425. Morgan Kaufmann Publishers, Inc., 1980.

[36] Z. Zheng, "A Comparison of Constructing Different Types of New Features for Decision Tree Learning", chapter 15, pp. 239 – 255. In [23], 1998. 2nd Printing, 2001.

Association Rules as Relative Modal Sentences Based on Conditional Probability

Tetsuya MURAI[1], Michinori NAKATA[2], and Yoshiharu SATO[1]

[1] Division of Systems and Information Engineering
Graduate School of Engineering, Hokkaido University
Kita 13, Nishi 8, Kita-ku, Sapporo 060-8628, JAPAN
E-mail: {murahiko, ysato}@main.eng.hokudai.ac.jp

[2] Faculty of Management and Information Sciences
Josai International University
1 Gumyo, Togane, Chiba 283-8555, JAPAN
E-mail: nakata@jiu.ac.jp

Abstract

Association rules are shown to be embedded as relative modal sentences into relative modal logic with conditional probability measure. For the purpose, a model for graded relative modal logic is formulated in a general way. Then, the model is actually constructed for a given database, from which association rules are mined.

1. Introduction

The authors have recently attempts to provide a framework of understanding logical meaning of *association rules* proposed by Agrawal et al.[1] in data mining. In our previous paper[8], we presented a point of view of association rules as conditionals in the sense of Chellas's conditional logic (cf. Chellas[2], p.268) and its extension. The logic differentiates conditional from material implication.

This paper shows that we can understand association rules as relative modal sentences based on Chellas's point of view (cf. Chellas[2], p.269) and fuzzy-measure-based semantics proposed by ourselves[5, 6, 7]. This form of association rules would be useful to consider them in the framework of rough set theory[9, 10] and granular computing[3, 4]

2. Association Rules

Let $\mathcal{I}$ be a finite set of *items*. Any subset in $\mathcal{I}$ is called an *itemset* X in $\mathcal{I}$, which can be a possible *transaction*. A *database* $\mathcal{D}$ is defined as a finite multiset of actually obtained transactions, thus $\mathcal{D} \subseteq 2^{\mathcal{I}}$.

For an itemset X, its *degree of support* $s(X)$ is defined by

$$s(X) \stackrel{\text{df}}{=} \frac{|\{T \in \mathcal{D} \mid X \subseteq T\}|}{|\mathcal{D}|},$$

where $|\cdot|$ is a size of a multiset.

Definition 1 (Agrawal et al.[1])
Given a set of items $\mathcal{I}$ and a database $\mathcal{D}$, an *association rule* is an implication of the form

$$X \Longrightarrow Y,$$

where X and Y are itemsets in $\mathcal{I}$ with $X \cap Y = \emptyset$. ∎

The following two indices are introduced:

Definition 2 (Agrawal et al.[1])

1. An association rule $r = (X \Longrightarrow Y)$ holds with *confidence* $c(r)$ $(0 \leq c(r) \leq 1)$ in $\mathcal{D}$ if and only if

$$c(r) = \frac{s(X \cup Y)}{s(X)}.$$

2. An association rule $r = (X \Longrightarrow Y)$ has a *degree of support* $s(r)$ $(0 \leq s(r) \leq 1)$ in $\mathcal{D}$ if and only if

$$s(r) = s(X \cup Y). \quad \blacksquare$$

In this paper, we will deal with the former index.

3. Models for Measure-Based Graded Relative Modality

Chellas[2] formulated a model for *conditional logic* where he remarked that a *conditional*

$$p \Rightarrow q$$

could be read as a form of *relative necessity*

$$[p]q,$$

which means that the necessity of what sentence q expresses relative to the condition given by sentence p (Chellas[2],p.269). In this paper, we extend his model using measure-based semantics formerly proposed by ourselves[5, 6, 7].

Given a countable set of *atomic sentences* $\mathcal{P}$, a *language* $\mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$ for graded relative modal logic is formed from $\mathcal{P}$ as the set of sentences closed under the usual propositional operators such as $\top$, $\bot$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$ as well as $[\,\cdot\,]_k$ (*graded relative necessity*) for $0 < k \leq 1$:

1. If $\mathsf{p} \in \mathcal{P}$ then $\mathsf{p} \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$.
2. $\top, \bot \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$.
3. If $p \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$ then $\neg p \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$.
4. If $p, q \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$ then $p \wedge q, p \vee q, p \rightarrow q, p \leftrightarrow q \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$,
5. If $(p, q \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$ and $0 < k \leq 1)$ then $[p]_k q \in \mathcal{L}_{\mathrm{GRML}}(\mathcal{P})$.

When $k = 1$, $[\,\cdot\,]$ is an abbreviation of $[\,\cdot\,]_1$.

Definition 3

A *measure-based graded RM-model* $\mathcal{M}_{\mathrm{GRML}}$ is defined as a structure

$$< W, \{m_w\}_{w \in W}, V >,$$

where

1. W is a non-empty set of (possible) worlds,
2. m_w is a conditional measure on W assigned at each world w in W,
3. V is a truth-assignment function, which gives either 1(true) or 0(false) to each atomic sentences at each world:

$$V : \mathcal{P} \times W \rightarrow \{0, 1\}. \quad \blacksquare$$

Define a relation $\models$ between a model, a world and an atomic sentence by

$$\mathcal{M}_{\mathrm{GRML}}, w \models \mathsf{p} \text{ iff } V(\mathsf{p}, w) = 1.$$

The relation $\models$ is extended for any compound sentence in the usual way, and, in particular, for graded relative modal sentences

$$\mathcal{M}_{\mathrm{GRML}}, w \models [p]_k q$$

$$\text{iff } m_w(\|q\|^{\mathcal{M}_{\mathrm{GRML}}} \mid \|p\|^{\mathcal{M}_{\mathrm{GRML}}}) \geq k,$$

where

$$\|p\|^{\mathcal{M}_{\mathrm{GRML}}} \stackrel{\mathrm{df}}{=} \{w \in W \mid \mathcal{M}_{\mathrm{GRML}}, w \models p\},$$

which is called the *truth set* or *proposition* of p in $\mathcal{M}_{\mathrm{GRML}}$.

When every world is assigned the same conditional measure, that is , $m_w = m_{w'}$ for every w, w' in W, the model is said to be *uniform*. The model is said to be *finite* if so is W.

In a finite uniform RM-model $\mathcal{M}_{\mathrm{GRML}}$, we may adopt the familiar definition of conditional probability:

Definition 4

Assume $\neg(p \leftrightarrow \bot)$, then

$$\mathcal{M}_{\mathrm{GRML}}, w \models [p]_k q$$

$$\text{iff} \quad Pr(\|p\|^{\mathcal{M}_{\mathrm{GRML}}} | \|q\|^{\mathcal{M}_{\mathrm{GRML}}})$$

$$= \frac{|\,\|p\|^{\mathcal{M}_{\mathrm{GRML}}} \cap \|q\|^{\mathcal{M}_{\mathrm{GRML}}}|}{|\,\|p\|^{\mathcal{M}_{\mathrm{GRML}}}|} \geq k. \quad \blacksquare$$

We have several soundness results based on fuzzy-measure-based semantics (cf. Murai et al.[5, 6, 7]) shown in Table 1.

4. Association Rules and Graded Relative Modality

Given a set of items $\mathcal{I}$ and a database $\mathcal{D} \subseteq 2^{\mathcal{I}}$, we construct a language $\mathcal{L}_{\mathrm{GRML}}(\mathcal{I})$, thus we regard any item as an atomic sentence.

Table 1: Soundness results of graded conditionals by conditional probability measures.

$0 < k \leq \frac{1}{2}$	$\frac{1}{2} < k < 1$	$k = 1$	Rules and Axiom schemata	
○	○	○	**RCEA.**	$\dfrac{p \leftrightarrow q}{[p]_k q \leftrightarrow [p]_k q}$
○	○	○	**RCEC.**	$\dfrac{q \leftrightarrow q'}{[p]_k q \leftrightarrow [p]_k q'}$
○	○	○	**RCM.**	$\dfrac{q \rightarrow q'}{[p]_k q \rightarrow [p]_k q'}$
○	○	○	**RCR.**	$\dfrac{(q \wedge q') \rightarrow r}{([p]_k q \wedge [p]_k q') \rightarrow [p]_k r}$
○	○	○	**RCN.**	$\dfrac{q}{[p]_k q}$
		○	**RCK.**	$\dfrac{(q_1 \wedge \cdots \wedge q_n) \rightarrow q}{([p]_k q_1 \wedge \cdots \wedge [p]_k q_n) \rightarrow [p]_k q}$
○	○	○	**I.**	$[p]_k p$
		○	**MP.**	$[p]_k q \rightarrow (p \rightarrow q)$
○	○	○	**CM.**	$[p]_k(q \wedge r) \rightarrow ([p]_k q \wedge [p]_k r)$
		○	**CC.**	$([p]_k q \wedge [p]_k r) \rightarrow [p]_k(q \wedge r)$
		○	**CR.**	$[p]_k(q \wedge r) \leftrightarrow ([p]_k q \wedge [p]_k r)$
○	○	○	**CN.**	$[p]_k \top$
○	○	○	**CP.**	$\neg([p]_k \bot)$
		○	**CK.**	$[p]_k(q \rightarrow r) \rightarrow ([p]_k q \rightarrow [p]_k r)$
	○	○	**CD.**	$\neg([p]_k q \wedge [p]_k \neg q)$
○			**CD$_{\mathrm{C}}$.**	$[p]_k q \vee [p]_k \neg q$

Definition 5

For a given database $\mathcal{D}$, its corresponding finite uniform graded RM-model $\mathcal{M}_{\mathrm{g}\mathcal{D}}$ is defined as a structure

$$< W_{\mathcal{D}}, Pr, V_{\mathcal{D}} >,$$

where

1. $W_{\mathcal{D}} = \mathcal{D}$,
2. Pr is the familiar conditional probability measure,
3. For every item x in $\mathcal{I}$ and every world (transaction) T in $W_{\mathcal{D}}$,

$$V_{\mathcal{D}}(x, T) = 1 \text{ iff } x \in T. \quad \blacksquare$$

Definition 6

For an association rule $r = (X \Longrightarrow Y)$, let

$$X = \{x_1, \cdots, x_m\},$$
$$Y = \{y_1, \cdots, y_n\}.$$

Then, two sentences p_X and p_Y are defined by

$$(1)\ p_X \stackrel{\mathrm{df}}{=} x_1 \wedge \cdots \wedge x_m,$$
$$(2)\ p_Y \stackrel{\mathrm{df}}{=} y_1 \wedge \cdots \wedge y_n. \quad \blacksquare$$

The conditional probability adopted in Definition 4 is nothing but the degree of confidence when it is applied to an association rule $r = (X \Longrightarrow Y)$. Then we can have the following theorem:

Theorem 7

Given a database $\mathcal{D}$ and its corresponding graded RM-model $\mathcal{M}_{\mathrm{g}\mathcal{D}}$, for an association rule $r = (X \Longrightarrow Y)$ with a positive degree of confidence $c(r)$,

$$c(r) \geq k \text{ iff } \mathcal{M}_{\mathrm{g}\mathcal{D}} \models [p_X]_k p_Y. \quad \blacksquare$$

When $c(r) = 1$ or 0, we have the following three lemmas.

Lemma 8

Given a database $\mathcal{D} \subseteq \mathcal{I}$, for an association rule $r = (X \Longrightarrow Y)$,

(1) $c(r) = 1$ iff $\forall T \in \mathcal{D}\ (X \subseteq T \Rightarrow Y \subseteq T)$,
(2) $c(r) = 0$ iff $\forall T \in \mathcal{D}\ (X \subseteq T \Rightarrow Y \not\subseteq T)$. ∎

Lemma 9

Given a database $\mathcal{D} \subseteq 2^{\mathcal{I}}$ and its graded RM-model $\mathcal{M}_{g\mathcal{D}}$, for any itemset $X \subseteq \mathcal{I}$ and any world (transaction) $T \in W_{\mathcal{D}}$,

$$X \subseteq T \text{ iff } T \in \|p_X\|^{\mathcal{M}_{g\mathcal{D}}}. \ ∎$$

Lemma 10

Given a database $\mathcal{D} \subseteq 2^{\mathcal{I}}$ and its graded RM-model $\mathcal{M}_{g\mathcal{D}}$, for an association rule $r = (X \Longrightarrow Y)$,

(1) $c(r) = 1$ iff $\|p_X\|^{\mathcal{M}_{g\mathcal{D}}} \subseteq \|p_Y\|^{\mathcal{M}_{g\mathcal{D}}}$,
(2) $c(r) = 0$ iff $\|p_X\|^{\mathcal{M}_{g\mathcal{D}}} \subseteq \|\neg p_Y\|^{\mathcal{M}_{g\mathcal{D}}}$. ∎

Thus we have the following theorem.

Theorem 11

Given a database $\mathcal{D}$ and its corresponding graded RM-model $\mathcal{M}_{g\mathcal{D}}$, for arbitrary association rule $r = (X \Longrightarrow Y)$,

$$\begin{aligned} c(r) = 1 &\quad \text{iff} \quad \mathcal{M}_{g\mathcal{D}} \models [p_X]p_Y, \\ 0 < c(r) < 1 &\quad \text{iff} \quad \mathcal{M}_{g\mathcal{D}} \models \neg[p_X]p_Y \wedge \neg[p_X]\neg p_Y, \\ c(r) = 0 &\quad \text{iff} \quad \mathcal{M}_{g\mathcal{D}} \models [p_X]\neg p_Y. \ ∎ \end{aligned}$$

5. Concluding Remarks

In this paper, we showed association rules could be regarded as relative modal sentences, which means that we no longer confine ourselves to the rules whose antecedent and consequent both have the form of conjunction.

The remained problem that we do not discuss in this paper is how to deal with the condition $X \cap Y = \emptyset$ of an association rule $X \Longrightarrow Y$. The condition is translated into GRML as p_X and p_Y cannot include the common atomic sentences, which means, when we make inference on association rules in GRML, we cannot use the fundamental formula **I**

$$[p]p.$$

By schema **MP** and modus ponens, formula **I** is found to be contained in the class of the law of identity

$$p \rightarrow p,$$

so it would be interesting to examine whether *intuitionistic logic* can be a base logic of GRML for association rules.

References

[1] Agrawal, R., Imielinski, T., Swami, A. (1993): Mining Association Rules between Sets of Items in Large Databases. *Proc. ACM SIGMOD Conf. on Management of Data*, 207–216.

[2] Chellas, B.F. (1980): *Modal Logic: An Introduction*. Cambridge Univ. Press, Cambridge.

[3] Lin,T.Y. (1998), Granular Computing on Binary Relation I. *L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 1: Methodology and Applications*, Physica-Verlag, pp.107-121.

[4] Lin,T.Y. (1998), Granular Computing on Binary Relation II. *L. Polkowski and A. Skowron (eds.), Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems*, Physica-Verlag, pp.122-140.

[5] Murai, T., Miyakoshi, M., Shimbo, M. (1993): Measure-Based Semantics for Modal Logic. *R.Lowen and M.Roubens (eds.), Fuzzy Logic: State of the Art*, Kluwer, Dordrecht, 395–405.

[6] Murai, T., Miyakoshi, M., Shimbo, M. (1994): Soundness and Completeness Theorems Between the Dempster-Shafer Theory and Logic of Belief. *Proc. 3rd FUZZ-IEEE (WCCI)*, 855–858.

[7] Murai, T., Miyakoshi, M., Shimbo, M. (1995) A Logical Foundation of Graded Modal Operators Defined by Fuzzy Measures. *Proc. 4th FUZZ-IEEE/2nd IFES*, 151–156.

[8] Murai, T., Nakata, M., Sato, Y. (2001) A Note on Conditional Logic and Association Rules. *T.Terano et al. (eds.)*, New Frontiers in Artificial Intelligence, LNAI 2253, Springer, 390–394.

[9] Pawlak, Z. (1982): Rough Sets. Int. J. Computer and Information Sciences, **11**, 341–356.

[10] Pawlak, Z. (1991): *Rough Sets: Theoretical Aspects of Reasoning about Data*. Kluwer, Dordrecht.

Interesting Association Rules and Multi-relational Association Rules

Jan Rauch,
EuroMISE – Kardio, Faculty of Informatics and Statistics,
University of Economics Prague, Czech Republic rauch@vse.cz

Abstract

Association rules $\varphi \approx \psi$ are introduced. The association rule $\varphi \approx \psi$ means that Boolean attributes φ and ψ are associated in the way given by the symbol $\approx$. This symbol is called 4ft quantifier. A condition concerning a four-fold contingency table of φ and ψ is associated to each 4ft quantifier. Various types of implication or equivalency of φ and ψ can be expressed. It is also possible to express relations corresponding to statistical hypotheses tests. Conditional association rules $\varphi \approx \psi / \chi$ are also introduced. Conditional association rule $\varphi \approx \psi / \chi$ means that when the condition χ is satisfied then the Boolean attributes φ and ψ are associated in the way given by 4ft quantifier $\approx$. The procedure 4ft-Miner mining for association rules $\varphi \approx \psi$ and $\varphi \approx \psi / \chi$ is described. Logical properties of association rules are further discussed. A definition of multi-relational association rules is suggested.

1. Introduction

The goal of this paper is to contribute to the discussion concerning definition of *valid novel, potentially useful, and ultimately understandable pattern.* Data mining is the process of identifying such patterns from data. We deal with association rules. We are not interesting in "classical" association rules of the form $X \rightarrow Y$ where X and Y are sets of items [1]. The intuitive meaning of $X \rightarrow Y$ is that transactions (e.g. supermarket baskets) containing set X of items tend to contain set Y of items. Two measures of intensity of association rule are used, *confidence* and *support*. The A-priori algorithm is a tool for mining association rules of this form.

The association rule is here understood as an expression $\varphi \approx \psi$ where φ and ψ are derived Boolean attributes. The intuitive meaning of association rule $\varphi \approx \psi$ is that Boolean attributes φ and ψ are associated in the way corresponding to the condition given by the symbol $\approx$. Symbol $\approx$ is called 4ft-quantifier. It denotes a condition concerning a four-fold contingency table of φ and ψ. Various types of implication or equivalency of φ and ψ can be expressed. It is also possible to express relations corresponding to statistical hypotheses tests (e.g. χ^2-test or Fisher's test).

We use the following ideas formulated in connection with GUHA method [2]:

- The Boolean attributes φ and ψ can be derived from columns of analysed data matrix There are clear syntactical rules describing how φ and ψ can be derived. These rules ensure that association rules $\varphi \approx \psi$ are *ultimately understandable patterns*.
- It is possible to define a very large set of *potentially useful* association rules by several parameters. We call them *interesting* association rules.
- There is a data mining procedure (GUHA procedure in the sense of [2]) input of which consists of the analysed data matrix and of a simply definition of the very large set of interesting (i.e. potentially useful) association rules. The mining procedure automatically generates each interesting association rule and verifies it in the analysed data matrix.
- Output of the mining procedure consists of all prime association rules. The association rule is prime if (i) it is interesting and valid in the analysed data matrix and (ii) it does not immediately follow from other more simple output association rules.
- There is software enabling us to find really *novel* association rules among the output *valid* association rules.

There are several implementations of these ideas, e.g. PC-GUHA [3]. The procedure 4ft-Miner mines for association rules of the form $\varphi \approx \psi$ and also for conditional association rules of the form $\varphi \approx \psi / \chi$. The procedure 4ft-Miner is a part of the academic software system for KDD; see http://lispminer.vse.cz/.

Association rules are introduced in section 2. Conditional association rules are briefly described in section 3. Possibilities of the definition of the set of interesting (i.e. potentially useful) association rules implemented in the 4ft-Miner are outlined in section 4. We also show that association rules can be understood as formulae of special logical calculus and that these logical calculi have practically important properties see section 5. The last goal of this paper is to outline a possibility to define multi-relational association rules see section 6. Multi-relational association rules can be also understood as formulae of special many sorted calculi see section 6.

2. Association rules

The association rule is the expression $\varphi \approx \psi$. It means that Boolean attributes φ and ψ are associated in the way corresponding to the 4ft-quantifier $\approx$. Association rule concerns the analysed data matrix. Boolean attributes φ and ψ correspond to {0,1} columns of analysed data matrix $\mathcal{M}$. They are derived from original columns A_1, …, A_K of $\mathcal{M}$ see Figure 1.

object	A_1	A_2	…	A_K	φ	ψ
o_1	$a_{1,1}$	$a_{1,2}$	…	$a_{1,K}$	1	0
…	…	…	…	…	…	…
o_n	$a_{n,1}$	$a_{n,2}$	…	$a_{n,K}$	1	1

Figure 1. – Analysed data matrix $\mathcal{M}$

We suppose that $a_{1,1}$ is the value of attribute A_1 for object o_1, $a_{n,K}$ is the value of the attribute A_K for object o_n etc. We denote the value of attribute A_i for object o_j as $A_i(o_j)$. Thus it is $A_i(o_j) = a_{j,i}$.

Boolean attributes φ and ψ are conjunctions of literals. *Literal* is a basic Boolean attributes or a negation of a basic Boolean attributes. *Basic Boolean attribute* has a form $A(\omega)$ where A is the column of data matrix $\mathcal{M}$ and ω is a subset of all possible values of column A. Basic Boolean attribute $A(\omega)$ is true for the object o_i if it is $A(o_i) \in \omega$. Possible values of the column A are called *categories*.

Let v_1, ..., v_u be categories of the column A_1. Then $A_1(v_1,v_3)$ is an example of the basic Boolean attribute. The basic Boolean attribute $A_1(v_1,v_3)$ is true for object o_1 if and only if it is $a_{1,1} = v_1$ or $a_{1,1} = v_3$.

Association rule $\varphi \approx \psi$ is verified on the basis of four-fold table of φ and ψ in the analysed data matrix $\mathcal{M}$. The four-fold table of φ and ψ in the data matrix $\mathcal{M}$ is the quadruple <a,b,c,d> see Table. 1. This quadruple is denoted as 4ft(φ, ψ, $\mathcal{M}$).

$\mathcal{M}$	ψ	$\neg\psi$	
φ	**a**	**b**	r
$\neg\varphi$	**c**	**d**	s
	k	l	n

Table 1. – Four-fold table 4ft(φ, ψ, $\mathcal{M}$) of φ, ψ in $\mathcal{M}$

Here *a* is the number of objects satisfying both φ and ψ, *b* is the number of objects satisfying φ and not satisfying ψ, *c* is the number of objects not satisfying φ and satisfying ψ, and *d* is the number of objects satisfying neither φ nor ψ. Further $r = a + b$ is the number of objects satisfying φ, similarly for *s*, *k*, and *l*, *n* is the number of all objects.

The association rule can be true or false in the given data matrix $\mathcal{M}$. The true-value of $\varphi \approx \psi$ in the data matrix $\mathcal{M}$ is denoted by Val($\varphi \approx \psi$,$\mathcal{M}$). If it is Val($\varphi \approx \psi$,$\mathcal{M}$) = *true* then the Boolean attributes φ and ψ are associated in the way corresponding to the 4ft-quantifier $\approx$ in the data matrix $\mathcal{M}$.

There is a condition concerning four-fold tables <a,b,c,d> associated to each 4ft quantifier $\approx$. This condition is understood as a {0,1}-function $\approx$(a,b,c,d). The association rule $\varphi \approx \psi$ is true in the data matrix $\mathcal{M}$ if it is $\approx$(a,b,c,d) = 1 where <a,b,c,d> = 4ft(φ, ψ, $\mathcal{M}$).

Various 4ft quantifiers are defined in [2] and [5], some examples follows:

- **Founded implication** $\Rightarrow_{p;Base}$ with parameters $0 < p \leq 1$ and $Base > 0$. The condition $\frac{a}{a+b} \geq p \wedge a \geq Base$ is associated to 4ft quantifier $\Rightarrow_{p;Base}$. Association rule $\varphi \Rightarrow_{p;Base} \psi$ can be interpreted as "*100p per cent of objects satisfying* φ *satisfy also* ψ" or "φ *implies* ψ *on the level 100p per cent*".
- **Lower critical implication** $\Rightarrow^{!}_{p;\alpha;Base}$ with parameters $0 < p \leq 1$, $Base > 0$ and $0 < \alpha \leq 0.5$. The condition $\sum_{i=a}^{a+b} \frac{(a+b)!}{i!(a+b-i)!} * p^i * (1-p)^{a+b-i} \leq \alpha \wedge a \geq Base$ is associated to 4ft quantifier $\Rightarrow^{!}_{p;\alpha;Base}$. Association rule $\varphi \Rightarrow^{!}_{p;\alpha;Base} \psi$ corresponds to a test (on the level α) of a null hypothesis H_0: $P(\varphi|\psi) \leq p$ against the alternative one H_1:

$P(\varphi|\psi) > p$. If association rule $\varphi \Rightarrow^{!}_{p;\alpha;Base} \psi$ is true in data matrix $\mathcal{M}$ then the alternative hypothesis is accepted.

- **Double founded implication** $\Leftrightarrow_{p;Base}$ with parameters $0 < p \leq 1$ and $Base > 0$. The condition $\frac{a}{a+b+c} \geq p \wedge a \geq Base$ is associated to 4ft quantifier $\Leftrightarrow_{p;Base}$. Association rule $\varphi \Leftrightarrow_{p;Base} \psi$ can be interpreted as "*100p per cent of objects satisfying* φ *or* ψ *satisfy both* φ *and* ψ" or "$\varphi \wedge \psi$ *implies* $\varphi \vee \psi$ *on the level 100p per cent*".
- **Founded equivalence** $\equiv_{p;Base}$ with parameters $0 < p \leq 1$ and $Base > 0$. The condition $\frac{a+d}{a+b+c+d} \geq p \wedge a \geq Base$ is associated to 4ft quantifier $\equiv_{p;Base}$. Association rule $\varphi \equiv_{p;Base} \psi$ can be interpreted as "*100p per cent of objects have the same value for* φ *and* ψ".
- **Fisher´s quantifier** $\sim_{\alpha,Base}$ with parameters $0 < \alpha \leq 0.5$ and $Base > 0$. The condition $ad > bc \wedge \sum_{i=a}^{\min(a+b,a+c)} \frac{r!s!k!!}{n!!(r-i)!(k-i)!(n-r-k-i)!} \leq \alpha \wedge a \geq Base$ is associated to 4ft quantifier $\sim_{\alpha,Base}$. Association rule $\varphi \sim_{\alpha,Base} \psi$ corresponds to a test (on the level α) of the null hypothesis of independence of φ and ψ against the alternative one of the positive dependence.

Let us give two remarks:

- The "classical" associational rule can be also understood as a 4ft quantifier $\rightarrow_{C;S}$ with the condition $\frac{a}{a+b} \geq C \wedge \frac{a}{a+b+c+d} \geq S$ associated to it. Here C is the support and S is the confidence.
- A further relation of two Boolean attributes is defined in [9]. It can be understood as a generalised quantifier $\approx^{E}_{\delta}$ with the condition $\frac{b}{a+b} < \delta \wedge \frac{c}{c+d} < \delta$ associated to it.

3. Conditional association rules

Procedure 4ft-Miner mines not only for association rules of the form $\varphi \approx \psi$ but also for *conditional association rules* of the form $\varphi \approx \psi / \chi$. Here φ, ψ and χ are conjunctions of literals. The intuitive meaning of association rule $\varphi \approx \psi / \chi$ is that Boolean attributes φ and ψ are associated in the way given by the 4ft-quantifier $\approx$ when the condition χ is satisfied.

Conditional association rule $\varphi \approx \psi / \chi$ is true in analysed data matrix M if association rule $\varphi \approx \psi$ is true in a data matrix $\mathcal{M} / \chi$. The data matrix $\mathcal{M} / \chi$ is data matrix consisting from all rows of data matrix $\mathcal{M}$ satisfying Boolean attribute χ. We suppose that there is at least one row of $\mathcal{M}$ satisfying χ.

4. Interesting Association Rules

The procedure 4ft-Miner (see http://lispminer.vse.cz/) mines for association rules Ant ≈ Suc and for conditional association rules Ant ≈ Suc / Cond. Here Ant, Suc and Cond are automatically generated conjunctions of literals. Ant is called antecedent, Suc is called succedent and Cond is called condition. An example of association rule is

$$A_1(v) \wedge A_4(u_1,u_2) \Rightarrow_{0.95;300} A_5(w).$$

Here A_1 is the attribute and v is one of its categories (i.e. possible values); similarly u_1 and u_2 are categories of the attribute A_4 and w is one of categories of the attribute A_5. Further $\Rightarrow_{0.95;300}$ is the quantifier of founded implication with parameters $p = 0.95$ and $Base = 300$.

The set of *interesting* association rules to be automatically generated and tested in the analysed data matrix is given by:

- Simple definition of all antecedents. It consists of:
  - a list of attributes from which literals of antecedent will be generated,
  - a simple definition of the set of all literals to be generated from each attribute,
  - minimal and maximal number of literals in antecedent.
- Analogous definition of all succedents.
- Analogous definition of all conditions (in the case of conditional association rules only).
- 4ft quantifier – there are 17 types of 4ft quantifiers.

Literal can be positive or negative. Positive literal is the expression $A(\omega)$, negative literal is the expression $\neg A(\omega)$. Here A is the attribute and ω is an own subset of the set of categories of the attribute A also section 2.

The set of all literals to be generated from a particular attribute is given by:

- A type of coefficient - there are five types of coefficients: subsets, intervals, left cuts, right cuts, and cuts.
- Minimal and maximal number of categories in coefficient.
- Positive/negative literal option:
  - only positive literals are generated,
  - only negative literals are generated,
  - both positive and negative literals are generated.

We show examples of literals with coefficients of particular types. We use an attribute A with the set of categories {1,2,3,4,5}. We suppose that only positive literals are generated. Examples of coefficients of particular types follow:

- **Subsets**: definition *subsets with 2-3 categories* defines literals A(1,2), A(1,3), A(1,4), A(1,5), A(2,3), …, A(3,4), ..., A(4,5), A(1,2,3), A(1,2,4), A(1,2,5), A(2,3,4), …, A(3,4,5).
- **Intervals**: definition *intervals with 2-3 categories* defines literals A(1,2), A(2,3), A(3,4), A(4,5), A(1,2,3), A(2,3,4) and A(3,4,5).
- **Left cuts**: definition *left cuts with maximally 3 categories* defines literals A(1), A(1,2) and A(1,2,3).
- **Right cuts**: definition *right cuts with maximally 4 categories* defines literals A(5), A(5,4), A(5,4,3) and A(5,4,3,2).
- **Cuts** means both left cuts and right cuts.

Output of the procedure 4ft-Miner consists of all prime association rules. The association rule is prime if both it is true in the analysed data matrix and it does not follow immediately from other more simple output association rules. The definition of prime association rule depends on properties 4ft-quantifier.

E.g. the definition of the prime association rule for the 4ft quantifier $\Rightarrow_{p;Base}$ must take into that if the association rule A(1) $\Rightarrow_{p;Base}$ B(1) is true then also the association rule A(1) $\Rightarrow_{p;Base}$ B(1,2) is true. Thus A(1) $\Rightarrow_{p;Base}$ B(1,2) immediately follow from the more simple association rule A(1) $\Rightarrow_{p;Base}$ B(1). The precise definition of prime association rules is out of the range of this paper.

The procedure 4ft-Miner does not use the A-priori algorithm. It uses the different algorithm based on representation of analysed data matrix by suitable strings of bits see [7]. It works very fast such that interactive work can be used in analysis of data matrices with about dozens thousands of rows. There is also additional software for filtering and sorting of found valid association rules.

5. Logical properties of association rules

The definition of the association rule can be done in a very precise way such that the association rules can be understood as formulae of special logical calculus.

Mathematical logic studies formal languages and formal data structures as their models. It is defined what does it mean that a sentence of formal language is true/false in a model. A very known example is first-order predicate calculus. There are lot of interesting results concerning universally valid formulas, deduction rules, an axiomatization, a decidability, etc. see e.g. [8].

Logical calculi formulae of which correspond to association rules were defined and studied in [2], [4], [5] and [6]. These logical calculi can be understood as modifications of classical predicate calculi. They differ from the classical predicate calculi in two features: (i) only finite models are allowed and (ii) 4ft-quantifiers are used.

Important results concerning correct deduction rules of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ were achieved. The deduction rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ is correct if the following is satisfied for each data matrix $\mathcal{M}$: If the association rule $\varphi \approx \psi$ is true in $\mathcal{M}$ then also the association rule $\varphi' \approx \psi'$ is true in $\mathcal{M}$.

An example of very simple correct deduction rule is the deduction rule $\frac{\varphi \Rightarrow_{p,Base} \psi}{\varphi \Rightarrow_{p,Base} \psi \vee \chi}$. It says: If $\varphi \Rightarrow_{p;Base} \psi$ is true in $\mathcal{M}$ then also $\varphi \Rightarrow_{p;Base} \psi \vee \chi$ is true in $\mathcal{M}$.

The correct deduction rules of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ are used in the procedure 4ft-Miner in two ways:

- If it is known that the association rule $\varphi \approx \psi$ is true then it is not necessary to generate and test the association rules $\varphi' \approx \psi'$ because of they are sure true.
- If it is known that $\varphi \approx \psi$ is true then it is not necessary to put into the output the association rule

φ' ≈ ψ' because of it is sure true. This approach but requires only **transparent** deduction rules.

More detailed description of application of correct deduction rules used in the procedure 4ft-Miner is out of range of this paper. Let us remark that deduction rules not only of the form $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ are used.

Various classes of 4ft-quantifiers can be defined e.g. implication 4ft-quantifiers, double implication 4ft-quantifiers and equivalency 4ft-quantifiers.

For all the 4ft-quantifiers used in the procedure 4ft-Miner there is a relatively simple condition equivalent to the fact that deduction rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ is correct [5]. This condition depends on the class the 4ft-quantifier ≈ belongs to. The condition concerns a propositional formula Φ(φ, ψ, φ', ψ') derived from Boolean attributes φ, ψ, φ', ψ'. Deduction rule $\frac{\varphi \approx \psi}{\varphi' \approx \psi'}$ is correct if and only if the formula Φ(φ, ψ , φ', ψ') is a tautology of the propositional calculus. A more detailed description of properties of deduction rules concerning association rules is out of range of this paper.

6. Multi-relational association rules

We outline a definition of multi-relational association rules. We use two simple data matrices (i.e. relations): $\mathcal{M}$ (see Figure 1) and Transactions (see Figure 2).

transaction	client	Amount	Bank	Type
t_1	o_1	5 000	A	X
t_2	o_1	7 000	Z	Z
…	o_1	…	…	…
t_{100}	o_1	8 000	B	Y
…	…	…	…	…
t_T	o_n	6 000	A	W

Figure 2. – Data matrix Transactions

We suppose that rows of data matrix $\mathcal{M}$ correspond to clients of a bank and that rows of data matrix Transactions correspond to bank transactions concerning particular clients. The client number o_1 has 100 transactions t_1,…, t_{100}.

The data matrices $\mathcal{M}$ and Transactions are related by the attribute Client. The attribute Client can be understood as a function from data matrix Transaction to data matrix $\mathcal{M}$. It assigns a row from data matrix $\mathcal{M}$ to each row of data matrix Transactions.

Each transaction is described by 3 attributes. The attribute Amount gives the amount of transferred money. The attribute Bank describes the bank from/to the transaction goes; there are 26 particular banks A,…, Z. The attribute Type describes the type of transaction; there are 6 types of transactions.

Various attributes concerning particular clients can be derived from transactions. An example is the attribute Average[Client;Amount;Bank(A)]. The value of the attribute Average[Client;Amount;Bank(A)] for the client o_1 is the average of values of the attribute Amount for all transactions concerning the client o_1 (i.e. transactions t_1, … t_{100}) such that the value of the attribute Bank is A. Analogously for the other clients.

We can define four new categories of the attribute Average[Client;Amount;Bank(A)]: *small*, *average*, *high* and *very high*. They can be defined e.g. by intervals <0,1 000), <1000, 5000), <5000, 10000), <10 000, ∞) respectively. Then the attribute Average[Client;Amount;Bank(A)] can be used in the definitions of the sets of interesting association rules see section 4.

The attribute Average[Client;Amount;Bank(A)] is defined using two data matrices (i.e. relations) $\mathcal{M}$ and Transactions. Thus the e.g. association rule

$$\text{Average[Client;Amount;Bank(A)](small)} \Rightarrow_{0.95;300} A_5\,(w)$$

concerning data matrix $\mathcal{M}$ can be understood as *a multi-relational association rule*.

There are 26 banks A, B, …, Z. Thus we can define further 25 analogous attributes Average[Client; Amount; Bank(B)] ,…, Average[Client; Amount; Bank(Z)]. The set of all 26 such defined attributes can be coded by Average[Client; Amount; Bank(?)]. This set is called a *family of attributes defined by the function*.

There are various ways how to define further families of attributes. We can use more attributes instead of the attribute Bank. An example is the family

Average[Client; Amount; Bank(?),TYPE(?)].

We can also use further functions e.g. Sum, Min etc. to define families of attributes

Sum[Client; Amount; Bank(?)],

Min[Client; Amount; TYPE(?), Bank(?)] etc.

We can also use 4ft-quantifier to define family of attributes. We show only very simple example. We start with the conditional association rule

$$\text{Bank(A)} \Rightarrow_{0.8;10} \text{Type(Y)} / \text{Client}(o_1)$$

concerning the data matrix Transactions. It is true if the association rule Bank(A) $\Rightarrow_{0.8;10}$ Type(Y) is true in the

Sampling Theories for Rule Discovery Based on Generality and Accuracy: the Worst Case and a Distribution-Based Case

Einoshin Suzuki
Electrical and Computer Engineering,
Yokohama National University
79-5, Tokiwadai, Hodogaya, Yokohama, 240-8501, Japan

Abstract

In this paper, we overview our endeavor for sampling theories of rule discovery based on generality and accuracy. Especially, we show our results for the worst case and for a distribution-based case. Through examples for conjunction-rule discovery, we show that the former certificates "safe discovery", while the latter discovers reliable rules.

1 Introduction

Although rule discovery has been extensively studied in data mining as one of its most important discovery method, its theoretical analyses are surprisingly rare. Several exceptions include Agrawal et al.'s analysis of association rule discovery [1], our analysis of a discovered rule based on simultaneous reliability evaluation [13], and our PAGA (Probably Approximately General and Accurate) discovery, a worst-case analysis of rule discovery [14]. While the first one is based on generality of a discovered rule, the latter two consider both generality and accuracy. This difference favors our studies since they exploit more information to give a detailed analysis.

Our simultaneous reliability evaluation gives the exact condition for discovering a rule which is sufficiently general and accurate under a user-specified degree reliability. On the other hand, our PAGA discovery gives a sample complexity, which represents the required number of examples, for the same objective. These studies allow us to deepen our undestandings toward a foundation of data mining.

2 PAGA Discovery

2.1 Rule

Let a data set contain m examples each of which is expressed by b discrete attributes and a class attribute. Typically rule discovery assumes no specific class attribute unlike classification. However, for the sake of formalization, we consider a rule which predicts a specific class attribute to be true.

Let a value v assignment $A = v$ to an attribute A be an atom. Here a continuous attribute is assumed to be discretized by a method such as those in [2] in advance. In this paper, we regard a given data set as a result of sampling with replacement from a true data set. We call the probability of examples each of which satisfies a propositional logical formula f the true probability $\Pr(f)$ of f. Similarly, an estimated probability which is obtained from a given data set for $\Pr(f)$ is represented by $\widehat{\Pr}(f)$. Note that $\widehat{\Pr}(f)$ can be calculated by the Laplace estimate or simply by the ratio of examples which satisfy f in the data set. We employ the latter method in this paper.

A rule r is represented as follows with a premise y which represents a propositional formula of atoms, and a conclusion x which represents a true assignment to the class attribute.

$$r : y \rightarrow x$$

An intuitive interpretation of r is that many examples satisfy y and those examples are likely to satisfy x with high probability. We define $\Pr(y)$ and $\Pr(x|y)$ as the generality and the accuracy of r respectively. Similarly, we call $\widehat{\Pr}(y)$ and $\widehat{\Pr}(x|y)$ the estimated generality and the estimated accuracy of r respectively.

A probabilistic if-then rule [11] is defined as follows, where y_i represents a single atom.

$$y_1 \wedge y_2 \wedge \cdots \wedge y_K \rightarrow x$$

In [13], we call a probabilistic if-then rule a conjunction rule, and this paper follows this paraphrasing.

Since a premise of a conjunction rule is represented by a combination of atoms, the number $|R|$ of possible conjunction rules is typically huge. The following gives $|R|$, where a data set contains b attributes and each of these attributes can have one of a values.

$$|R| = (a+1)^b - 1 \tag{1}$$

2.2 Discovery Problem

In this paper, the objective of a user is to obtain, with high probability $1-\delta$, a rule of which generality and accuracy are no smaller than $1-\zeta$ and $1-\epsilon$ respectively. Typically multiple rules are obtained in rule discovery, but we restrict ourselves to single-rule discovery for the sake of analysis.

$$\begin{aligned}&\text{Objective : Find } y \rightarrow x \text{ which satisfies}\\ &\Pr\left[\Pr(y) \geq 1-\zeta,\ \Pr(x|y) \geq 1-\epsilon\right] \geq 1-\delta \\ &\text{where } \zeta,\ \epsilon,\ \delta\ >\ 0\end{aligned} \tag{2}$$

A discovery algorithm to be analyzed obtains a rule of which generality and accuracy are no smaller than user-given thresholds θ_{S} and θ_{F} respectively. As stated above, since a given data set is a result of sampling from a true data set, the user employs thresholds $\theta_{\mathrm{S}} \neq 1-\zeta$, $\theta_{\mathrm{F}} \neq 1-\epsilon$ in applying the algorithm.

$$\begin{aligned}&\text{Algorithm : Find } y \rightarrow x \text{ which satisfies}\\ &\widehat{\Pr}(y) \geq \theta_{\mathrm{S}},\ \widehat{\Pr}(x|y) \geq \theta_{\mathrm{F}}\end{aligned} \tag{3}$$

An interesting problem here is to bound the required number m of examples to accomplish (2) under (3). This problem can be named as PAGA (Probably Approximately General and Accurate) discovery after the well-known PAC (Probably Approximately Correct) learning [6, 9].

3 Sample Complexity Based on PAGA Discovery

Here, we assume that we avoid finding a bad rule. This condition can be considered as important in several domains where reliability represents a crucial concern.

3.1 Theoretical Analysis

First we introduce preliminaries which are needed in subsequent analyses. If the domain of a probabilistic variable X is $\{0, 1, \cdots, m\}$ and the probability distribution of the variable is represented as follows, X is said to follow a binary distribution [4].

$$\begin{aligned}\Pr(X=k) &= \mathrm{B}(k; m, p)\\ &= \left(\begin{array}{c} m \\ k \end{array} \right) p^k (1-p)^{m-k}\end{aligned} \tag{4}$$

where p represents a constant $0 < p < 1$ and $k = 0, 1, \cdots, m$. The Chernoff bound states that the following holds for an arbitrary constant $a > p$ [1].

$$\Pr(X > am) < \exp[-2m(a-p)^2] \tag{5}$$

From (2), a bad rule $r_{\mathrm{b}} : y \rightarrow x$ satisfies

$$\Pr(y) < 1-\zeta \text{ or } \Pr(x|y) < 1-\epsilon. \tag{6}$$

Since we assume that we avoid finding a bad rule, the employed thresholds for generality and accuracy are relatively large. This assumption together with (2) and (3) necessitate

$$\theta_{\mathrm{S}} > 1-\zeta \text{ and } \theta_{\mathrm{F}} > 1-\epsilon. \tag{7}$$

From (6) and (7),

$$\theta_{\mathrm{S}} > \Pr(y) \text{ or } \theta_{\mathrm{F}} > \Pr(x|y). \tag{8}$$

Since $r_{\mathrm{b}} : y \rightarrow x$ is discovered,

$$\widehat{\Pr}(y) \geq \theta_{\mathrm{S}} \text{ and } \widehat{\Pr}(x|y) \geq \theta_{\mathrm{F}}. \tag{9}$$

Let the number of examples in the given data set be m. If and only if y and xy are satisfied by at least $\lceil m\theta_{\mathrm{S}} \rceil$ and $\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil$ examples respectively in the data set, r_{b} happens to be discovered. Since each of the numbers of examples which satisfy y and xy follows a binary distribution,

$$\begin{aligned}&\Pr \quad (r_{\mathrm{b}} \text{ discovered})\\ &\leq \mathrm{MAX}\left[\sum_{k=\lceil m\theta_{\mathrm{S}} \rceil}^{m} \mathrm{B}(k; m, \Pr(y)), \right.\\ &\qquad \left. \sum_{k=\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}} \rceil}^{m\widehat{\Pr}(y)} \mathrm{B}(k; m\widehat{\Pr}(y), \Pr(x|y)) \right]\\ &< \mathrm{MAX}\left\{ \exp\left[-2m \left(\frac{\lceil m\theta_{\mathrm{S}} \rceil}{m} - \Pr(y) \right)^2 \right], \right.\end{aligned} \tag{10}$$

$$\exp\left[-2m\widehat{\Pr}(y)\left(\frac{\lceil m\widehat{\Pr}(y)\theta_{\mathrm{F}}\rceil}{m\widehat{\Pr}(y)}-\Pr(x|y)\right)^2\right]\Bigg\} \tag{11}$$

$$< \mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right],\right.$$
$$\left.\exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\}. \tag{12}$$

Note that, in (10), we consider separately the case in which a bad rule $r_{\mathrm{b}1}$ in terms of generality is discovered and the case in which a bad rule $r_{\mathrm{b}2}$ in terms of accuracy is discovered. The first and second terms correspond to the left inequality and the right inequality of (6) respectively. Since $\Pr(r_{\mathrm{b}1})$ and $\Pr(r_{\mathrm{b}2})$ are unknown, we bound $\Pr(r_{\mathrm{b}}$ discovered) by $\mathrm{MAX}[\Pr(r_{\mathrm{b}1}$ discovered$), \Pr(r_{\mathrm{b}2}$discovered$)]$. In (11), the Chernoff bound (5) is employed from (8). Finally in (12), we employ (6) and the left inequality of (9).

Let the set of all possible rules and the set of all bad rules be R and R_{b} respectively, and let the cardinality of a set S be $|S|$. The probability of discovering a bad rule satisfies the following inequalities.

$$\Pr\quad(R_{\mathrm{b}}\text{ contains a discovered rule})$$
$$< |R_{\mathrm{b}}|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right],\right.$$
$$\left.\exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \tag{13}$$
$$\leq |R|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right],\right.$$
$$\left.\exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \tag{14}$$

Note that we allow to count multiple times the cases in which several bad rules satisfy the discovery condition in (13), and (14) uses $|R| \geq |R_{\mathrm{b}}|$. Our objective (2) requires the following with respect to a sufficiently small δ.

$$|R|\mathrm{MAX}\left\{\exp\left[-2m(\theta_{\mathrm{S}}-1+\zeta)^2\right],\right.$$
$$\left.\exp\left[-2m\theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2\right]\right\} \leq \delta \tag{15}$$

We obtain a sample complexity for discovery in which finding a bad rule is avoided with a high probability.

$$m \geq \frac{\ln\left(\frac{|R|}{\delta}\right)}{2\mathrm{MIN}\left[\left(\theta_{\mathrm{S}}-1+\zeta\right)^2, \theta_{\mathrm{S}}\left(\theta_{\mathrm{F}}-1+\epsilon\right)^2\right]} \tag{16}$$

The sample complexity of (16), which certificates "safe discovery", holds true even if we assume that we avoid overlooking a good rule [14].

The above inequality describes influence of each parameter to the sample complexity quantitatively. As we have seen in section 2, $|R|$ is typically large and is thus important even if its influence is tolerated by a logarithmic function. The second most important factors are $\theta_{\mathrm{S}}-1+\zeta$ and $\theta_{\mathrm{F}}-1+\epsilon$. Since they influence the lower bound of m by the inverse of their squares, they can be problematic when they are small. Since each of these terms represents the difference of a threshold and the user-expected value, $\theta_{\mathrm{S}}-1+\zeta$ and $\theta_{\mathrm{F}}-1+\epsilon$ can be named as the margin of generality and the margin of accuracy respectively.

3.2 Case of Conjunction Rule Discovery

From (1) and (16), the sample complexity is given as follows if we restrict the discovered rule to a conjunction rule.

$$m \geq \frac{\ln\left[(a+1)^b-1\right]+\ln\left(\frac{1}{\delta}\right)}{2\mathrm{MIN}\left[\left(\theta_{\mathrm{S}}-1+\zeta\right)^2, \theta_{\mathrm{S}}\left(\theta_{\mathrm{F}}-1+\epsilon\right)^2\right]} \tag{17}$$

Note that settling $a=1$ gives the case of association rule discovery.

Firstly, $\ln(1/\delta)$ can be typically ignored when $\delta =$ 0.01 - 0.05 from $\ln[(a+1)^b-1] \gg \ln(1/\delta)$, thus the lower bound of m is approximately proportional to b. Secondly, since the number a of possible values for an attribute only affects the right-hand side through a logarithmic function, a is typically not so important as b and margins of generality and accuracy. We show, in figure 1, a plot of the sample complexity against $\mathrm{MIN}[(\theta_{\mathrm{S}}-1+\zeta)^2, \theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2]$ for $b=10^2, 10^3, 10^4$, where we settled $a=2$ and $\delta=0.05$.

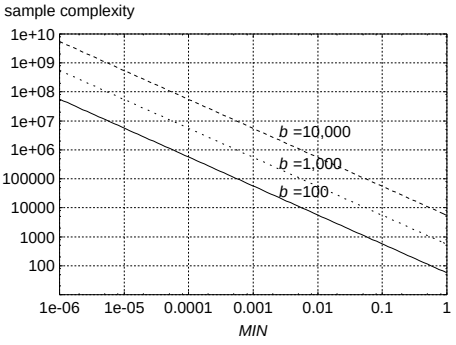


Figure 1: Sample complexity for conjunction rule discovery without finding a bad rule. In the figure, MIN represents $\mathrm{MIN}[(\theta_{\mathrm{S}}-1+\zeta)^2, \theta_{\mathrm{S}}(\theta_{\mathrm{F}}-1+\epsilon)^2]$.

The figure shows that the required number of examples for safe discovery can be prohibitively large for

small margins. It should be anyway noted that our analyses in this paper correspond to the worst case, and the required number of examples in a real discovery problem can be much smaller.

4 Simultaneous Reliability Evaluation of a Discovered Rule

4.1 Theoretical Analysis

Simultaneous reliability evaluation of a discovered rule [13] also deals with the case of sampling m examples from a true data set in rule discovery as in section 3. Its objective is also represented by (2).

Let $\overline{x}$ represent the negation of x. This analysis fixes m and employs neither θ_{S} nor θ_{F}. We assume that $(m\Pr(xy), m\Pr(\overline{x}y))$ follows a two-dimensional normal distribution, and obtain the exact condition for accomplishing the objective analytically. This is a different framework from section 3: we used a discovery algorithm with fixed thresholds θ_{S}, θ_{F} in (3) and bounded the number m of sampled examples. The problem dealt here can be reduced to the problem of deriving and analyzing two tangent lines of an ellipse, and applying Lagrange's multiplier method gives the following analytical solutions.

$$\left(1-\beta(\delta)\sqrt{\frac{1-\widehat{\Pr}(y)}{m\widehat{\Pr}(y)}}\right)\widehat{\Pr}(y) \geq 1-\zeta \tag{18}$$

$$\left(1-\beta(\delta)\sqrt{\frac{\widehat{\Pr}(\overline{x},y)}{\widehat{\Pr}(x,y)\{(m+\beta(\delta)^2)\widehat{\Pr}(y)-\beta(\delta)^2\}}}\right)$$
$$\widehat{\Pr}(x|y) \geq 1-\epsilon \tag{19}$$

Here $\beta(\delta)$ represents a positive constant which defines the size of a $1-\delta$ confidence region i.e. the ellipse for $(m\Pr(xy), m\Pr(\overline{x}y))$, and can be obtained by a simple numerical integration [13]. Note that (18) and (19) represent conditions for generality and accuracy respectively. Each of them states that the corresponding estimated probability multiplied by a coefficient which is related to the size of the confidence region is no smaller than the corresponding user-expected value ($1-\zeta$ or $1-\epsilon$).

4.2 Application to Data Sets

Since this study assumes a specific distribution to the simultaneous occurrence of random variables, it does not fall in the category of worst-case analysis. Rather, it can reduce the number of discovered rules. The proposed method was tested with data sets from several domains, including 21 benchmark data sets [7] in the machine learning community.

We show experimental results, where the parameters were set to $\delta = 0.05$, $\theta_{\mathrm{S}} = 0.1$ and $\theta_{\mathrm{F}} = 0.9$. In the experiments, a continuous attribute was discretized in advance by [3], and a premise contains at most three atoms. Figure 2 shows the ratio of the number of discovered rules between three approaches each of which considers reliability and the approach without reliability evaluation. A bullet ($\bullet$), a times ($\times$) and a triangle ($\triangle$) correspond to our approach, the approach based on the Chernoff bound and an approach based on the normal approximations of the binomial distributions respectively. In the figure, a bar chart represents the number of rules discovered by the approach without reliability evaluation in a logarithmic scale.

From figure 2, we note that the proposed approach reduces a considerable number of rules in many data sets compared with the other approaches. The proposed method, since it evaluates more information than the other methods, reduces a larger number of rules or at least the same number of rules. We have also shown that the proposed method is more time-efficient than the alternatives by experiments since it can prune unreliable rules using accuracy [13].

5 Discussions on Related Topics

5.1 PAC Learning

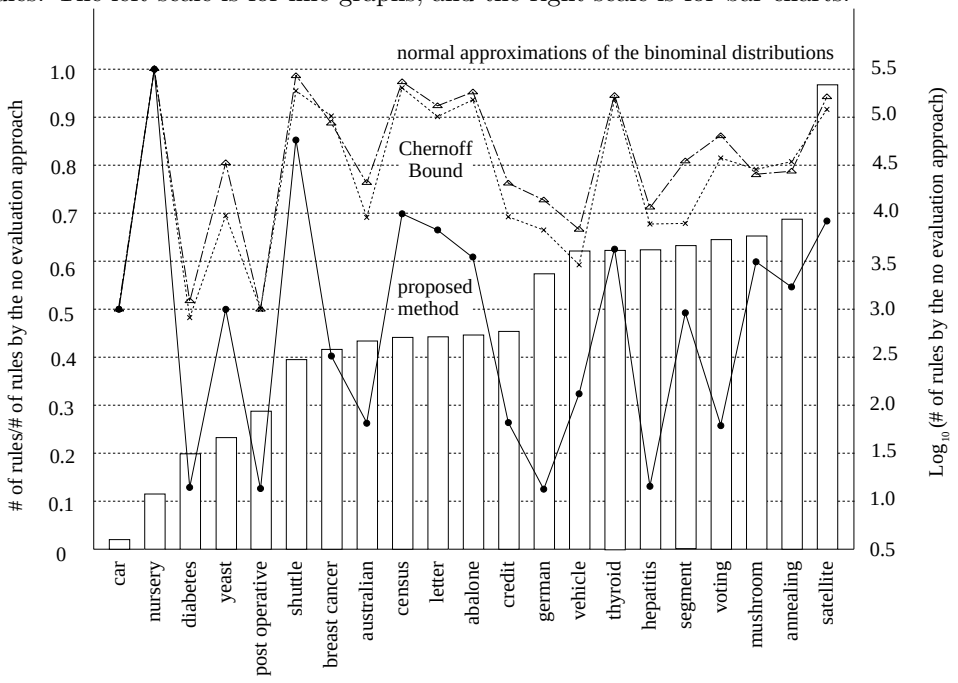
PAC learning represents a worst-case analysis for classification. Our results in section 3.1 can be considered as an extension to its preliminary version [9].

Compared to our PAGA discovery, [9] ignores the case of learning a classifier with low generality and the case of learning a classifier which is inconsistent to the training examples. In this case, application of the Chernoff bound can be skipped, and for a bad classifier h_{b}, we obtain $\Pr(h_{\mathrm{b}}\text{ learned}) = (1-\epsilon)^m$. In [9], a lower bound of the required number of examples m is given by the following, where H represents a set of all classifiers.

$$m \geq \frac{\ln\left(\frac{|H|}{\delta}\right)}{\epsilon} \tag{20}$$

Note that (20) resembles to (16): it only ignores generality ($\theta_{\mathrm{S}} = 1$ and no ζ), assumes $\theta_{\mathrm{F}} = 1$, and omits

Figure 2: Performance of 3 rule-discovery methods each of which considers reliability with respect to the number of discovered rules. The left scale is for line graphs, and the right scale is for bar charts.



the squares in ϵ^2 and 2 in the denominator. The last omissions are due to skipping the Chernoff bound.

5.2 Multiple Comparison

Jensen and Cohen's multiple comparison [5] proposes a prudent view of classification and is related to our PAGA discovery. Its essential point can be stated as a probabilistic explanation that the more candidates of classifiers are inspected in a learning algorithm, the smaller accuracy is exhibited by the obtained classifier. The multiple comparison provides a comprehensive unified view of several studies including overfitting [10] and oversearching [8], and [5] also proposes several realistic measures.

Since this study deals with classification as PAC learning, it ignores generality. This corresponds to considering only the second term in (10). Since [5] considers the case of $\theta_{\mathrm{F}} < 1$, it provides a more realistic framework to learning than [9]. The multiple comparison differs from PAGA discovery in that it directly calculates, based on a binary distribution without using the Chernoff bound, the probability for a bad classifier to satisfy at least $\lceil m\theta_{\mathrm{F}} \rceil$ examples. Moreover, they calculate exactly the probability that no bad classifier is learned while we, in (13), allow counting multiples times the cases in which more than one bad rules satisfy the discovery condition. Let the set of all bad classifiers be H_{b}, the probability in [5] is given by the following.

$$\Pr(H_{\mathrm{b}} \text{ contains a learned classifier}) = 1 - [1 - \Pr(h_{\mathrm{b}} \text{ learned})]^{|H|} \quad (21)$$

Pursuing strictness in calculation can be considered as a double-edged sword. Jensen and Cohen give no analytical solutions to the required number of examples for successful learning. We attribute this reason to the fact that resolving (21) for m is relatively difficult. We have employed several approximations in our PAGA discovery, and these were necessary to bound m analytically.

5.3 Theoretical Analysis of Association Rule Discovery

Analyses of association rule discovery [1] include the number of examples satisfied by an itemset in a sampled data set. This analysis is highly related to our study in that both of the two deal with the case of sampling m examples from a true data set in rule discovery.

The analysis provides a specification of the Chernoff bound (5), where X is regarded as $m\widehat{\Pr}(f)$ for an itemset f. It first regards the right-hand side $\exp[-2m(a-p)^2]$ as the upper bound of the probability for $\widehat{\Pr}(f)$ to deviate at least $a-p$ from its value p $(=\Pr(f))$ in the true data set. Next, it gives several examples of values for $a-p$ and δ in $\exp[-2m(a-p)^2]=\delta$, and represents the corresponding values of m in a table.

The discovery algorithm employed in [1] first obtains, by an algorithm called Apriori, a set of imtemsets f each of which satisfies $\widehat{\Pr}(f) \geq \theta_{\mathrm{S}}$. Then, it generates a set of association rules from this set. One of the motivations of the above analysis was to reduce the run-time of Apriori by the use of a sampled data set. Due to this motivation, [1] ignores accuracy unlike our studies. Moreover, since it considers a single association rule, the study fails to relate the size of a discovery problem to the number of examples needed for successful discovery unlike our PAGA discovery. Superiority of our simultaneous reliability evaluation was demonstrated empirically in section 4.2.

6 Conclusions

This paper has presented our previous studies for sampling in rule discovery. They represent PAGA (Probably Approximately General and Accurate) discovery, a worst-case analysis of rule discovery [14], and simultaneous reliability evaluation [13], a distribution-based analysis.

The latter was successfully applied to rule-pair discovery, in which pairs of a strong rule and its exception rule are discovered [12]. We plan to extend PAGA discovery for this problem.

References

[1] R. Agrawal, H. Mannila, R. Srikant, H. Toivonen, and A. I. Verkamo: Fast Discovery of Association Rules, *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, AAAI/MIT Press, Menlo Park, Calif. (1996).

[2] J. Dougherty, R. Kohavi, and M. Sahami: Supervised and Unsupervised Discretization of Continuous Features, *Proc. Twelfth Int'l Conf. on Machine Learning (ICML)*, pp. 194-202 (1995).

[3] U. M. Fayyad and K. B. Irani : Multi-Interval Discretization of Continuous-Valued Attributes for Classification Learning, *Proc. Thirteenth Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1022–1027 (1993).

[4] W. Feller: *An Introduction to Probability Theory and Its Applications*, Wiley, New York (1957).

[5] D. D. Jensen and P. R. Cohen: "Multiple Comparisons in Induction Algorithms", *Machine Learning*, Vol. 38, No. 3, pp. 309–338 (2000).

[6] M. J. Kearns and U. V. Vazirani: *An Introduction to Computational Learning Theory*. MIT Press, Cambridge, Mass. (1994).

[7] C. J. Merz and P. M. Murphy: UCI Repository of Machine Learning Databases, *http:// www.ics.uci.edu/ ~mlearn/MLRepository. html*, Univ. of California, Dept. of Information and Computer Sci. (1994).

[8] J. R. Quinlan and R. Cameron-Jones: "Oversearching and Layered Search in Empirical Learning", *Proc. Fourteenth Int'l Joint Conf. on Artificial Intelligence (IJCAI)*, pp. 1019–1024 (1995).

[9] S. Russel and P. Norvig: *Artificial Intelligence, a Modern Approach*, pp. 552–558, Prentice Hall, Upper Saddle River, N. J. (1995).

[10] C. Schaffer: "Overfitting Avoidance as Bias", *Machine Learning*, Vol. 10, No. 2, pp. 153–178 (1993).

[11] P. Smyth and R. M. Goodman: "An Information Theoretic Approach to Rule Induction from Databases", *IEEE Trans. Knowledge and Data Eng.*, Vol. 4, No. 4, pp. 301–316 (1992).

[12] E. Suzuki: "Autonomous Discovery of Reliable Exception Rules", *Proc. Third Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 259–262 (1997).

[13] E. Suzuki: "Simultaneous Reliability Evaluation of Generality and Accuracy for Rule Discovery in Databases", *Proc. Fourth Int'l Conf. on Knowledge Discovery and Data Mining (KDD)*, pp. 339–343 (1998).

[14] E. Suzuki: "Worst-Case Analysis of Rule Discovery", *Discovery Science, LNAI 2226 (DS)*, pp. 365–377, Springer (2001). (Erratum: http:// www.slab.dnj.ynu.ac.jp/erratumds2001.pdf.)

Rule Induction, Rough Sets and Matroid Theory

Shusaku Tsumoto
Department of Medicine Informatics,
Shimane Medical University, School of Medicine,
Enya-cho Izumo City, Shimane 693-8501 Japan

Abstract - Rule induction methods, such as C4.5 and AQ has been introduced to extract rule-based knowledge from databases. In data mining context, these methods have discovered knowledge interesting for experts. However, since no formal approach has been proposed to treat these methods in a common framework. In this paper, we introduce matroid theory and rough sets to construct a common framework for empirical machine learning methods which induce the combination of attribute-value pairs from databases. Combination of the concepts of rough sets and matroid theory gives us an excellent set-theoretical framework and enables us to understand the differences and the similarities between these methods from the viewpoint of partitions of the universe.

Keywords— **Keywords: Rough Sets, Matroid Theory, Rule Induction, Data Mining**

I. Introduction

In order to acquire knowledge from databases, there have been proposed several methods of inductive learning, such as ID3 family[2], [7] and AQ family[1], [4], [5]. These methods are applied to discover meaningful knowledge from large database, whose experimental shows they are very useful to find simple rules. However, since no formal approach has been proposed to compare these methods, they are compared by using real-world databases[1], [2], [5], [7], such as medical databases. These results suggest some differences between these methods. However, since sometimes these differences may depend on applied domains, general discussion has been left unsolved.

In this paper, we introduce matroid theory[11], [12], [13] and rough sets[6] to construct a common framework for empirical machine learning methods which induce knowledge from attribute-value pattern databases. Combination of the concepts of rough sets and matroid theory gives us an excellent set-theoretical framework and enables us to understand the differences of these methods clearly. Using this framework, we formulate the following three types of rules induction methods: covering, reduction and (recursive) partitioning.

The paper is organized as follows: section Section 2 presents fundamentals of rough sets. Section 3 gives a rule induction algorithm based on rough sets. Section 4 introduces matroid theory. Section 5 and 6 discusses matroids in rule induction methods and partitioning. Finally, Section 7 concludes this paper.

This research is based on the former work on AQ matroid, Pawlak matroid and ID3 greedoid[10] in which the partitions of universe and information granules were not focused on. In this paper, several results are discussed from these two viewpoints.

Correspondence: Email: tsumoto@computer.org; WWW: http://www.shimane-med.ac.jp/med_info/tsumoto/index.htm; Telephone: +81 853 20 2172; FAX: +81 853 20 2170

II. Rough Sets

In the subsequent sections, we adopt the following notations, which is introduced by Skowron and Grzymala-Busse[8]. Let U denote a nonempty, finite set called the universe and A denote a nonempty, finite set of attributes, i.e., $a : U \rightarrow V_a$ for $a \in A$, where V_a is called the domain of a, respectively.Then, a decision table is defined as an information system, $A = (U, A \cup \{d\})$. The atomic formulas over $B \subseteq A \cup \{d\}$ and V are expressions of the form $[a = v]$, called descriptors over B, where $a \in B$ and $v \in V_a$. The set $F(B, V)$ of formulas over B is the least set containing all atomic formulas over B and closed with respect to disjunction, conjunction and negation. For each $f \in F(B, V)$, f_A denote the meaning of f in A, i.e., the set of all objects in U with property f, defined inductively as follows.

(1) If f is of the form $[a = v]$ then, $f_A = \{s \in U | a(s) = v\}$
(2) $(f \wedge g)_A = f_A \cap g_A$; $(f \vee g)_A = f_A \vee g_A$; $(\neg f)_A = U - f_a$

By the use of this framework, classification accuracy and coverage, or true positive rate is defined as follows.

Definition 1:
Let R and D denote a formula in $F(B, V)$ and a set of objects which belong to a decision d. Classification accuracy and coverage(true positive rate) for $R \rightarrow d$ is

defined as:

$$\begin{aligned}\alpha_R(D) &= \frac{|R_A \cap D|}{|R_A|}(= P(D|R)), \text{ and}\\ \kappa_R(D) &= \frac{|R_A \cap D|}{|D|}(= P(R|D)),\end{aligned}$$

where $|A|$ denotes the cardinality of a set A, $\alpha_R(D)$ denotes a classification accuracy of R as to classification of D, and $\kappa_R(D)$ denotes a coverage, or a true positive rate of R to D, respectively. □

Finally, we define partial order of equivalence as follows:

Definition 2: Let R_i and R_j be the formulae in $F(B,V)$ and let $A(R_i)$ denote a set whose elements are attribute-value pairs of the form $[a = v]$ included in R_i. If $A(R_i) \subseteq A(R_j)$, then we define this partial order as:

$$R_i \preceq R_j.$$

A. Rough Set Approximation

Rough set consists of two important components for rule induction. One is approximation and the other one is reduct.

Concerning approximation, Pawlak proposed upper and lower approximation of the target concept as follows.

Definition 3: Let $\{R_i\}$ and D denote a formula in $F(B,V)$ and a set of objects which belong to a decision d. If $\{R_i\}$ forms a partition of the universe U ($\forall i, j, R_i \cap R_j = \phi, U = \cup_i R_i$), the equivalence class U/R is defined as $\{R_i\}$. Then, the lower and upper approximation of D, denoted by $\underline{R}(D)$ and $\overline{R}(D)$ are defined as:

$$\begin{aligned}\underline{R}(D) &= \bigcup\{Y \in U/R : Y \subseteq D\}\\ \overline{R}(D) &= \bigcup\{Y \in U/R : Y \cap D \neq \phi\}\end{aligned}$$

□

If we use accuracy and coverage, the above definitions can be rewritten as:

$$\begin{aligned}\underline{R}(D) &= \bigcup\{Y \in U/R : \alpha_Y(D) = 1.0\}\\ \overline{R}(D) &= \bigcup\{Y \in U/R : \kappa_Y(D) > 0\},\end{aligned}$$

where $\kappa_{Y_0}(D) = 1.0$ for $Y_0 = \vee Y_i$ s.t. $Y_i \in U/R$ and $\kappa_Y(D) > 0$.

B. Reducts

Based on the concepts of rough sets, Pawlak[6] introduces *Reduction of Knowledge*, which is a method to examine the independencies of the attributes iteratively and extract the minimum indispensable part of equivalence relations. Here we only mention about the definition of *consistent rules* and their knowledge reduction. For further details, readers could refer to [6].

Definition 4: Let R be a formula in $F(B,V)$ and D be a set of samples which belongs to a target concept. $R \Rightarrow D$ is called a *consistent rule* when R_A holds the following relation:

$$Posi_R(D) = R_A \subseteq D,$$

where $Posi_{R_j}(D)$ denotes the lower approximation of D in terms of R. □

Definition 5: Let R_0 be equal to $R \wedge f$, where f is a formula in $F(B,V)$. If f is satisfied with the following equation:

$$Posi_{R_0}(D) = Posi_{R \wedge f}(D) = Posi_R(D),$$

then we say that f is *dispensable* in R_0, and can be deleted from R_0. □

Intuitively, reduction procedure removes redundant variables which do not contribute to classification of a class. It is notable that classical Pawlak's method is an exhaustive search for decision rules. Thus, strictly, it is not a kind of the greedy algorithm. However, in order to suppress a computational power needed for computation, heuristic searches are unavoidable.

III. Rule Induction

The simplest rule induction algorithm is to induce probabilistic rules, which is defined below.

$$R \stackrel{\alpha,\kappa}{\rightarrow} d \ s.t. \ \alpha_R(D) > \delta_\alpha, \ and \ \kappa_R(D) > \delta_\kappa,$$

where δ_α, δ_κ are thresholds for accuracy and coverage, respectively. From this definition, a rule induction algorithm is described as Figure 1.

This algorithm is a kind of greedy algorithm which finds independent variables in terms of rough sets. We will discuss this characteristic later.

IV. Matroid Theory

A. Definition of Matroids

Matroid theory abstracts the important characteristics of matrix theory and graph theory, firstly developed by

```
procedure Induction of Classification Rules;
  var
    i : integer;    M, L_i : List;
  begin
    L_1 := L_er;
/* L_er: List of Elementary Relations */
    i := 1;    M := {};
    for i := 1 to n do
    /* n: Total number of attributes */
      begin
        while ( L_i ≠ {} ) do
          begin
            Sort L_i w.r.t. the value of coverage;
            Select one pair R = ∧[a_i = v_j] from L_i,
            which have the largest value on coverage;
            L_i := L_i − {R};
            if (κ_R(D) ≥ δ_κ)
              then do
                    if (α_R(D) ≥ δ_α)
                      then  do S_ir := S_ir + {R};
          /* Include R as Classification Rule */
                    M := M + {R};
          end
        L_{i+1} := (A list of the whole combination of
                the conjunction formulae in M);
      end
  end {Induction of Classification Rules };
```

Fig. 1. An Algorithm for Classification Rules

Whitney[13] in the thirties of this century. The advantages of introducing matroid theory is the following: 1) Since matroid theory abstracts graphical structure, this shows the characteristics of formal structure in graph clearly. 2) Since a matroid is defined by the axioms of independent sets, it makes the definition of independent structure clear. 3) Duality is one of the most important structure in matroid theory, which enables us to treat relations between dependency and independency rigorously. 4) The greedy algorithm is one of the algorithms for acquiring an optimal base of a matroid. Since this algorithm is studied in detail, well-established results can be applied to our problem.

Although there are many interesting and attractive characteristics of matroid theory, we only discuss about duality, and the greedy algorithm, both of which are enough for our algebraic specification. For further information on matroid theory, readers might refer to [11].

First, we begin with the definition of a matroid. A matroid is defined as an independent space which satisfies the following axioms:

Definition 6: The pair $M(E, \mathcal{J})$ is called a matroid, if
1) E is a finite set,
2) $\emptyset \in \mathcal{J} \subset 2^E$,
3) $X \in \mathcal{J}, Y \subset X \Rightarrow Y \in \mathcal{J}$,
4) $X, Y \in \mathcal{J}, card(X) = card(Y) + 1 \Rightarrow (\exists a \in X - Y)(Y \cup \{a\}) \in \mathcal{J}$,

where $\mathcal{J}$ is called **independent sets**, and where a set X whose cardinality is maximum in $\mathcal{J}$ is called a **base**.

If $X \in \mathcal{J}$, it is called **independent**, otherwise X is called **dependent**. □

From the above definition, it is easy to prove that independence sets are equivalent to the power sets of a base. For example, let us consider a case when a base β is equal to $\{a_1, a_2, a_3\}$ in the space spanned by a set $\{a_1, a_2, a_3, a_4, a_5\}$[1].

Then, E and $\mathcal{J}$ is equal to $\{a_1, a_2, a_3, a_4, a_5\}$ and 2^β, respectively. Thus, $\{a_1, a_2\}$ is independent, although $\{a_1, a_2, a_4\}$ is dependent.

B. Duality

Another important characteristic is duality. While this concept was firstly introduced in graph theory, a deeper understanding of the notion of the duality in graph theory can be obtained by examining matroid structure. Definition of duality is as follows:

Definition 7: If $M = (E, \mathcal{J})$, is a matroid with a set of bases β, then the matroid with a set of elements E, and a set of bases $\beta^* = \{E - B | B \in \beta\}$ is termed the **dual** of M and is denoted by M^*. □

From this definition, it can be easily shown that $(M^*)^* = M$, and M and thus M^* are referred to a **dual matroid pair**. And we have the following theorem:

Theorem 1: If M is a matroid, then M^* is a matroid. □

C. Greedy Algorithm

Since it is important to calculate a base of a matroid in practice, several methods are proposed. In these methods, we focus on the greedy algorithm. This algorithm can be formulated as follows:

Definition 8: Let B be a variable to store the calculated base of a matroid, and E denote the whole set of attributes. We define the Greedy Algorithm to calculate a base of a matroid as follows:

1. $B \leftarrow \phi$.
2. Calculate "priority queue" Q using weight function of

[1]This type of a matroid is called a *simple matroid*.

E.
3. If B is a base of $M(E, \mathcal{J})$ then stop. Else go to 4.
4. $e \leftarrow first(Q)$, which has a minimum weight in Q.
5. If $B \cup \{e\} \in \mathcal{J}$ then $B \leftarrow B \cup \{e\}$. goto 2. □

This algorithm searches one solution which is optimal in terms of one weight function. Note that a matroid may have many bases. The bases derived by the greedy algorithm are optimal to some **predefined** weight function. Hence if we cannot derive a suitable weight function we cannot get such an optimal base. In the following, we assume that we can define a good weight function for the greedy algorithm. For example, we can use *information gain* as defined in [2], [7] for such function. When information gain is used as a weight function, the greedy algorithm with this weight function gives a solution optimal to apparent accuracy. since this gain is closely related with apparent accuracy or apparent accuracy. In other words, the solution is optimal to apparent rate, that is, in the language of statistics, the algorithm calculates the best class allocation of training samples. Under this assumption, this algorithm has the following characteristics:

Theorem 2: The complexity of the greedy algorithm is

$$\mathcal{O}(mf(\rho(M)) + m \log m),$$

where $\rho(M)$ is equal to a rank of matroid M, m is equal to the number of the elements in the matroid, $|E|$, f represents a function of computational complexity of an independent test, which is the procedure to test whether the obtained set is independent, and is called *independent test oracle.* □

Theorem 3: The optimal solution is derived by this algorithm if and only if a subset of the attributes satisfies the axioms of the matroid. □

D. Unions and Intersections of Matroids

Since matroid theory is based on set-theoretical framework, we can define unions and intersections of matroids [2].

First, we define the union of matroids as follows.

Definition 9: Let $M_1, M_2, \cdots, M_m$ be matroids on S. Let

$$\mathcal{J} = \{X : X = X_1 \cup X_2 \cup \cdots \cup X_m; X_i \in \mathcal{J}(M_i)(1 \leq i \leq m)\}.$$

Then J is the collection of independent sets of a matroid on S, which satisfies the axioms of independent sets. We refer to the matroid M whose independent sets is equal to $\mathcal{J}$ as:

$$M = M_1 \vee M_2 \vee \cdots \vee M_m,$$

called the union of matroids. □

Therefore, it guarantees that the problem can be decomposed into disjoint subproblems and that the total solution can be obtained by taking unions of the solutions of sub-problems.

E. Matroid in Rule Induction

Under the above assumption, we can constitute a matroid for AQ method as follows:

Theorem 4: Let B denote the base of a matroid such that $B_A = D_k \subseteq D$. If we define an independent set $\mathcal{J}(D_k)$ as $\{A(R_j)\}$ which satisfies the following conditions:

1) $R \preceq B$,
2) $B_A \subseteq R_A$,
3) $\forall R_i$ s.t. $R_i \prec R \preceq B$, $B_A \subseteq R_A \subset R_{iA}$,

where the equality holds only if $R(j)_A = B$. Then this set satisfies the definition of a matroid. We call this type of matroid, $M(E, \mathcal{J}(D_k))$, a *induction matroid.* □

The first condition means that a base is a maximal independent set and each relation forms a subset of this base. And the second condition is the characteristic which satisfies all of these equivalence relations. Finally, the third condition denotes the relationship between the equivalence relations: Any relation R_i which forms a subset of $A(R_j)$ must satisfy $R_{jA} \subset R_{iA}$. Note that these conditions reflects the conditional part of AQ algorithm.

We can also derive the intersection of three bases, which corresponds to the core of reducts. Furthermore, since the independent test depends on the calculus of indiscernible sets, is less than $\mathcal{O}(\rho(M) * n^2)$ where n denotes a sample size, or the number of learning examples, the computational complexity is given as follows:

Theorem 5: Assume that we do not use constructive generalization. Then the complexity of AQ algorithm is less than

$$\mathcal{O}(mn^2\rho(M)) + m \log m),$$

where $\rho(M)$ is equal to a rank of matroid M, m is equal to the number of the elements in the matroid, $|E|$. □

Hence the computational complexity of rule induction method depends mainly on the number of the elements of a matroid, since it increases exponentially as the number of the attribute-value pairs grows large.

[2]Unfortunately, intersections of matroids do not always satisfy the axioms of a matroid in general [11]. However, in this paper, we deal with only special class of a matroid, called *simple matroids*, whose intersections always satisfies the axioms of a matroid.

F. Matroids in Calculation of Reducts

On the other hand, since $\rho(M)$ is the number of independent variables, $m - \rho(M)$ is equal to the number of dependent variables. From the concepts of the matroid theory, if we define an dependent set $\mathcal{I}$ as shown below, then $M(E, \mathcal{I})$ satisfies the condition of the dual matroid of $M(E, \mathcal{J})$.

Theorem 6: Let B denote the base of a matroid such that $B_A = D_k$. If we define an independent set $\mathcal{I}(D_k)$ as $\{A(R_j)\}$ which satisfies the following conditions:

1) $B \prec R$,
2) $B_A = R_A$,
3) $\forall R_i$ s.t. $B \prec R_i \preceq R$, $D_k = R_A = R_{iA} = B_A$,

then $M(E, \mathcal{I}(D_k))$ is a dual matroid of $M(E, \mathcal{J}(D_k))$, and we call $M(E, \mathcal{I}(D_k))$ a *reduction matroid.* □

The first condition means that a base is a maximal independent set and each relation forms a superset of this base. And the second condition is the characteristic which satisfies all of these equivalence relations. Finally, the third condition denotes the relationship between the equivalence relations: Any relation R_i which forms a subset of $A(R_j)$ must satisfy $R_{iA} \subset R_{jA}$. Note that these conditions reflects the conditional part of reduction method. As shown above, the algorithm of reduct computation is formally equivalent to the algorithm for the dual matroid of induction matroid, and its computational complexity is less than $\mathcal{O}((p-\rho(M))*(n^2+2n)+m\log m)$. Hence, we get the following theorem.

Theorem 7: The complexity of the reduction method is less than

$$\mathcal{O}(mn^2(p - \rho(M))) + m\log m),$$

where p is a total number of attributes, $\rho(M)$ is equal to a rank of matroid M, and m is equal to the number of the elements in the matroid, $|E|$. □

From these consideration, if $\rho(M)$ is small, induction algorithm performs better than reduction one under our assumption.

V. Partitioning

Induction of decision trees, such as CART[2] and ID3[7] is another inductive learning method based on the ordering of variables using information entropy measure or other similar measures. This method splits training samples into smaller ones in a top-down manner until it cannot split the samples, and then prunes the overfitting leaves. The main characteristics of the bases derived by ID3 are the following. First, the attribute-value pairs are totally ordered, and in each branch, which corresponds to each base for D_j, subsets of each branch have to preserve this order. For example,let a base be composed of binary attributes, say, $\{a, b, c\}$, in which ID3 algorithm chooses these attributes from the left to the right. Then the allowable subsets are: $\{a\}$, $\{a, b\}$, and $\{a, b, c\}$. Second, each base have the common attribute at least in the first element. For example, if one base is composed of $\{a, b, c\}$, then another base is like $\{a, b, \bar{c}\}$, or $\{\bar{a}, d, c\}$, where $\bar{a}$ denotes a complement of a.

These characterics can be captured by a partition matroid or a greedoid. First, a partition matroid is defined as:

Definition 10: Let π be a partition that separates E into m disjoint blocks E_1, $\cdots$, E_m a and let $d_i (i = 1, \cdots, m)$ be m given nonnegative integers. Then, for any E, π and d_i, $M = (E, J)$ is a *partition matroid* if $J = \{I | I \subseteq E, |I \cap E_i| \leq d_i, i = 1, \cdots, m\}$.

In a decision tree, the horizontal section of each level gives a partition of the universe. Thus,

Theorem 8: Let $\{E_i\}$ $(i = 1, \cdots, m)$ denote the partition corresponding to a given level h of decision tree (h is equal to the number of attriubes used in the formulae.) Then, the partition $\{F_i\}$ whose level is less than h gives a partition matroid with:

$$\forall F_i, \exists E_j, |F_i \cap E_j| = |E_j|$$

□

However, this definition cannot capture the characteristics of a set of attributes in decision trees shown above. So, from the viewpoint of attributes, greedoid is much better. Although these global constraints, especially the first one, decreases the search space spanned by attribute-value pairs, those makes a family of subsets lose the characteristics of a matroid. In fact, a set of the subsets derived by ID3 method does not satisfy the axiom of a matroid. It satisfies the axiom of a greedoid[12], which is a weaker form of a matroid, defined as follows.

Definition 11: The pair $M(E, \mathcal{J})$ is called a greedoid, if

1) E is a finite set,
2) $\emptyset \in \mathcal{F} \subset 2^E$,
3) $X \in \mathcal{F}$, there is an $x \in X$ such that $X - x \in \mathcal{F}$,
4) $X, Y \in \mathcal{F}, card(X) = card(Y) + 1 \Rightarrow (\exists a \in X - Y)(Y \cup \{a\}) \in \mathcal{F}$.

If $X \in \mathcal{J}$, it is called **feasible**, otherwise X is called **infeasible**. □

Note that the third condition becomes a weaker form, which allows for the total ordering of elements. Because of this weakness, some important characteristics of matroids, such as duality, are no longer preserved. Using the above formulation, the search space for ID3 is defined as a ordered greedoid in the following.

Definition 12: Let B denote the base of a matroid such that $B_A = D_k$. If we define a feasible set $\mathcal{K}(D_k)$ as: $\{A(R_j)\}$ which satisfies the following conditions:

1) $R_j \preceq B$,
2) $B_A \subseteq R_{j\,A}$,
3) $\forall R_i$ s.t. $R_i \preceq R_j \preceq B$, $D_j = B_A \subseteq R_{j\,A} \subset R_{i\,A}$,

where the equality holds only if $R_j = B$, and if we demand that the each $K(D_k)$ should satisfy the following conditions:

(1) for all R_i and R_l, $R_i \preceq R_l$ or $R_l \preceq R_j$ holds ,
(2) $\forall\, \mathcal{K}(D_q)$ and $\mathcal{K}(D_p)$, For all $R_j \in \mathcal{K}(D_q)$ and $R_i \in \mathcal{K}(D_p)$, if $R_{i\,A} \cap R_{j\,A} \neq \phi$, then $R_j \preceq R_i$,

then this set satisfies the definition of a greedoid. We call this type of greedoid, $G(E, \mathcal{K}(D_k))$, a *recursive partitioning greedoid.* □

Note that each D_k has exactly one feasible set $\mathcal{K}(D_k)$.

Therefore the whole ID3 algorithm is equivalent to the greedy algorithm for acquiring a set of bases of ID3 greedoid, denoted by $\{\mathcal{K}(D_k)\}$.

A. Computational Complexity of Decision Tree

As shown above, Decision tree algorithm is also the greedy algorithm for deriving a base of a greedoid. However, the main feature of this algorithm is that two constraints to independent sets are given. This reduces the search space of independent sets, because the sets which satisfy the above two constraints are not so many. Here, we obtain the following theorem:

Theorem 9: The complexity of decision tree algorithm is less than

$$\mathcal{O}(mn^2\rho(M)) + m\log m),$$

where $\rho(M)$ is equal to a rank of greedoid M,m is equal to the number of the elements in the greedoid, $|F|$. □

The difference in computational complexity between AQ and ID3 is the value of m.

VI. Conclusion

In this paper, we combine the concepts of matroid theory with those of rough sets, which gives us an excellent framework and enables us to understand the differences between rule induction, calculation of reducts and recursive partitoning more clearly.

Acknowledgements

This research was supported by the Grant-in-Aid for Scientific Research on Priority Areas(B) (No.759) "Implementation of Active Mining in the Era of Information Flood" by the Ministry of Education, Science, Culture, Science and Technology of Japan.

References

[1] Bergadano, F., Matwin, S., Michalski, R.S. and Zhang, J. Learning Two-Tiered Descriptions of Flexible Concepts: The POSEIDON System, *Machine Learning*, **8**, 5-43, 1992.

[2] Breiman, L., Freidman, J., Olshen, R. and Stone, C. *Classification And Regression Trees.* Belmont, CA: Wadsworth International Group, 1984.

[3] Grzymala-Busse, J.W. LERS- A system for Learning From Examples based on Rough Sets, in: Slowinski, R.(ed) Intelligent Decision Support. Handbook of Application and Advances of the Rough Set Theory, Kluwer Academic Publishers, 1992, Dordrecht, pp.3-18.

[4] Michalski, R.S. A Theory and Methodology of Machine Learning. Michalski, R.S., Carbonell, J.G. and Mitchell, T.M., *Machine Learning - An Artificial Intelligence Approach*, 83-134, Morgan Kaufmann, CA, 1983.

[5] Michalski, R.S., et al. The Multi-Purpose Incremental Learning System AQ15 and its Testing Application to Three Medical Domains, *Proc. of AAAI-86*, 1041-1045, Morgan Kaufmann, CA, 1986.

[6] Pawlak, Z. *Rough Sets*, Kluwer Academic Publishers, Dordrecht, 1991.

[7] Quinlan JR: *C4.5 - Programs for Machine Learning.* Morgan Kaufmann, Palo Alto CA, 1993.

[8] Skowron, A. and Grzymala-Busse, J. From rough set theory to evidence theory. In: Yager, R., Fedrizzi, M. and Kacprzyk, J.(eds.) *Advances in the Dempster-Shafer Theory of Evidence*, pp.193-236, John Wiley & Sons, New York, 1994.

[9] Skowron, Rauser,C.M. The Discerniblity Matrix and Functions in Information Systems, in: Slowinski, R.(ed) Intelligent Decision Support. Handbook of Application and Advances of the Rough Set Theory, Kluwer Academic Publishers, 1992, Dordrecht, pp.331-362.

[10] Tsumoto, S. and Tanaka, H. A Common Framework of Empirical Learning Methods based on Rough Sets and Matroid Theory. *Fundamentae Informaticae*, pp. 273-288, 1996.

[11] Welsh, D.J.A. *Matroid Theory*, Academic Press, London, 1976.

[12] White, N.(ed.) *Matroid Applications*, Cambridge University Press, 1991.

[13] Whitney, H. On the abstract properties of linear dependence, *Am. J. Math.*, **57**, 509-533, 1935.

[14] Ziarko, W. The Discovery, Analysis, and Representation of Data Dependencies in Databases, in:*Knowledge Discovery in Database*, Morgan Kaufmann, 1991.

[15] Ziarko, W. Variable Precision Rough Set Model, *Journal of Computer and System Sciences*, **46**, 39-59, 1993.

Sampling in Data Mining
(A Summary)

Trong Wu
Department of Computer Science
Southern Illinois University Edwardsville
Edwardsville, Illinois 62026-1656, U. S. A.
e-mail:twu@siue.edu; phone:618-650-2393

Abstract

This paper provides a tutorial about the numerous methods which have been developed and used for sampling in statistics. The main focus of this paper is to provide basic definitions or procedures for each sampling technique, to determine the sample sizes required by each of the various techniques, and to give some important statistical features, particularly the sample mean and variance, that can characterize the properties of the population and help us to make decisions. Most of the material included in this paper is abstracted out from standard graduate texts and rewritten for tutorial purposes. The paper is not only intended as a tutorial of sampling techniques, but also as an introduction into the basic elements of sampling theory.

1. Introduction

Data mining is a set of methods or procedures for extracting and processing previously unknown, incomprehensible, and un-actionable information from a large database to make certain critical business decisions. Data mining is often used in the knowledge discovery process to distinguish previously unknown relationships and patterns within a data set. Specially, it is applied to a large database and used to make important decisions.

Data mining techniques have been widely used in the business, industry, government agencies, and other organizations for sampling and processing data, and making decisions. In business, a marketing company may use them to develop a model to determine how many customers will respond to telephone solicitation based on previous information found in their database. In industry, a manufacturer may analyze a set of sensor data to isolate conditions that lead to termination of a non-profitable production. In a government agency, law enforcement agents can sift through the records of financial transactions looking for patterns that can indicate money laundering, drug smuggling, or other criminal activity. A nonprofit organization can use its previous donation records to predict the incoming year's fund raising.

The goals of data mining are to create a model, a set of executable codes that can be used to score a database, to perform some classification and estimation for prediction. In addition, we will obtain a more complete understanding by uncovering patterns and relationships for descriptive purposes. In order to achieve this properly, efficiently, and accurately, we need to select data correctly, effectively, and exactly. Therefore, various sampling techniques techniques should be used. In order to do these correctly, sampling techniques should be applied properly.

This paper addresses various sampling techniques that are commonly used in data mining. We will begin by generally introducing sampling techniques including the purpose and procedures of each. Next, we will review two basic sampling techniques, simple random sampling and sampling proportions and percentages. After we study estimations of sampling sizes, we then investigate more advanced sampling techniques such as stratified random sampling, systematic sampling, and cluster sampling.

2. Sampling Techniques

The importance of sampling and data mining is widely recognized and has been very widely adopted with a long history of application in business, industry, government agencies, and nonprofit organizations.

In the last thirty years the most important feature in sampling has been the rapid increase in the number and types of surveys taken by sampling. For example, the Statistical Office of the United Nations publishes reports from time to time on "Sample Surveys of Current Interest" conducted by member countries. The 1968 report lists surveys from 46 countries. Many of these surveys seek information of obvious importance to national planning on topics such as agricultural production and the use of land, percentage of unemployment and the potential size of the labor force, industrial production, retail and wholesale prices, health status of people, and family incomes and expenditures. But more specialized inquiries can also be found over time such as: annual leave arrangements, causes of divorce, rural debt and investment, household water consumption, holiday spending, age structure of cows, and job vacancies for various countries.

2.1 Advantages of sampling

Sampling has become to play a prominent part in national decennial census. In the United States a 5% sampling was

introduced in the 1940 National Census by asking persons or about one person from 1240. Surveys used to provide facts bearing on sales and advertising policy in market research may employ samples of only a few thousand. For the same reason, the data can be collected and summarized more quickly with a sample than with a complete count. This is a vital consideration when the information is urgently needed.

Sampling, when properly conducted, can provide numerous advantages over a complete census. However, in the most cases, it is necessary to study a number of samples of information that we collect. The advantages of sampling as compared with complete census are given as follows:

Lower cost
If information is extracted from a small fraction of all census data, the cost will be much smaller than the cost of conducting of entire complete census. In the U.S., the most important recurrent survey, the decennial census, usually samples one person out of 1,240. In business, a marketing research firm could use samples sizes of only a few thousand.

Higher speed
Using sampling techniques, data can be collected and analyzed much faster than with a complete census. This is a particularly necessary consideration when the information is urgently needed.

Wider scope
In many cases, a complete census is not possible to implement. The remaining selection lies between obtaining the information by sampling or not at all. Thus, sampling surveys have more scope and flexibility regarding the types of information that can be obtained. On the other hand, if accurate information is desired for many subdivisions of the population, the sample size needed to do the job is sometimes so large that a complete enumeration offers the best solution.

Better accuracy
Because personnel of higher quality can be employed and given intensive training and because more careful supervision of the field work and processing of results becomes feasible when the volume of work is reduced, a sample may produce more accurate results than the results of a complete enumeration.

Sampling procedures differ greatly in their complexity. To take a sample from 10,000 records, neatly arranged and numbered in a file, is an easy task. It is another matter to sample international refugee community residences with many different spoken languages and dialects, which are very suspicious of an inquisitive stranger. This makes the sampling survey very difficult. The principal steps in a conducting a sample is discussed in the following headings.

2.2 Objectives of the sampling

A precise statement of the objectives is greatly helpful for sampling. Otherwise, it is easy to forget the objectives in a complex survey when engaged in the details of planning work. It can cause the decision making at variance with the original objectives. Other objective includes:

Population to be sampled
The word population is used to denote the aggregate from which the sample is chosen, usually stored in a file in a given database. The definition of the population may present no problem; for example, when a sampling a batch of electronic parts in order to estimate the average lifespan of the parts. In sampling a population of factories, on the other hand, one needs a definition for a factory, and borderline cases may arise. The definition must be usable in practice: in particular, in data mining issues that will determine which database and files should be sampled without any hesitation. The file to be sampled should coincide with the population about which we want information. Sometimes, for reasons of practicability or convenience, the sampled population is more restricted than the target population. Therefore, any supplementary information from some other files that can be gathered about the nature of the differences between sampled and target population may be helpful.

Data to be collected
In general, collecting data can be done in two ways: one sampling from data file of a database and the other from the returning questionnaires of a survey sampling. In both cases, it is necessary to verify that all the data are relevant to the purposes of the sampling and that no essential data are omitted. In a survey sampling, if a conductor asks too many and overlong questions that will lower the quality of the answers to important as well as unimportant questions.

Degree of precision desired
Sometime the results of data mining and sample surveys are subject to some uncertainty because only part of the file and population has been measured and because of errors of measurement. This uncertainty can be reduced by taking larger samples and by using superior instruments of measurement. To accomplish this, it usually costs more money and time. Therefore, the specification of the degree of precision that we want in the final results is a crucial step.

The purpose of this paper is to give a briefly discuss various sampling techniques that may be suitable for use in data mining. We will begin with simple random sampling that generates a sample of n units out of N such that every one in the sample has an equal chance of being drawn. We then will study sampling proportions and percentages, estimations of sample size, stratified random sampling, systematic sampling, and finally we will study cluster

sampling; it use the available information that could achieve a greater precision.

3. Simple Random Sampling

The simple random sampling is a method of selecting n units out of the N such that each of the C_n^N distinct samples has an equal chance of being drawn. In the actual practice a simple random sample is drawn unit by unit. The units in the population are numbered from 1 to N. A series of random numbers between 1 and N is then generated by means of a computer program called random number generator that produces such a list. At any draw the process used must give an equal chance of selection to any number in the population not already drawn. The units that associate these n numbers constitute the sample. Hence all C_n^N distinct samples have an equal chance of being selected by this method.

$$\frac{n}{N}\frac{(n-1)}{(N-1)}\cdots\frac{1}{(N-n+1)}=\frac{n!(N-n)!}{N!}=\frac{1}{C_n^N}$$

4. Sampling Proportions and Percentages

In many cases, we wish to estimate the total number of a proportion or the percentage of units in the population that have some special properties of our interested. Many of the results from survey and censuses are interested in these measures. Consider a population consists of two groups G and G' or defect and non-defect D and D', respectively. Let A be the number of units in G and let a be the number of units in the sample. Proportions of units in population and sample are $P = A/N$ and $p = a / n$, respectively.

The sample estimate of P is p, and the sample estimate of A is Np. Parameters of *binomial* distribution is often used to estimates a and p. For a finite population, the exact distribution for this category data objects is a hyper-geometric distribution.

5. The Estimation of Sampling Size

In a sampling plan, it is necessary to determine the size of the sample. This decision is very important. If the sample is too large that may wastes the resources; if the sample is too small that will decrease the accuracy of information. To estimate the sample size, we may consider the criteria in selection of sample size and then start to process then to do the sampling.

(1) There must have a clear statement that states the expectation of the sample. The statement should include the desired precisions of some parameters.

(2) An equation that characterize the sample size n and the desired precision of certain parameters. The equation may contain some other parameters of the population.

5.1 Examples

(1) A simple example
A public school teacher is preparing a study of percentage of students taking breakfast before come to school in his school district. In this case, it is feasible that the teacher may take a simple random sample for his study. How large should the sample be? If the percentage of having breakfast is corrected within ±5%, it is feasible for his study. In this situation, if the sample shows 48% to have breakfast, the percentage for the whole school district is sure to lie between 43% and 53%.

$$V(p)\approx\frac{PQ}{n},\qquad \sigma(p)\approx\sqrt{PQ/n}$$

If we assume that $\sigma(p) = 5/2$, then $n = 4PQ/25$. For any value of P between 40 and 60, the product PQ is about 2400, the n = 384. For more conservative we may take n = 400.

(2) The formula for n in sampling for proportions
Consider units are classified into two classes, G and G', some maximum error d in estimated proportion p of units in class G has been assigned, and there is a small risk probability α that the actual error is larger than d, thus we have the probability

$$P_r(|p-P|\geq d)=\alpha$$

For simple random sampling and p is taken as normally distributed. From the last section we have

$$\sigma_p=\sqrt{\frac{N-n}{N-1}}\sqrt{\frac{PQ}{n}}$$

Hence, the formula that connects n with the desired degree of precision is

$$d=t\sqrt{\frac{N-n}{N-1}}\sqrt{\frac{PQ}{n}},$$

where t is the abscissa of the normal curve that cuts off the α at the tail end. Solve for n and we have

$$n=\frac{\frac{t^2PQ}{d^2}}{1+\frac{1}{N}\left(\frac{t^2PQ}{d^2}-1\right)}$$

For practical use, we may take p of P substituted in

the above formula. If N is large, let

$$n_0 = \frac{t^2 pq}{d^2} = \frac{pq}{V}$$

and this is called initial estimate. Hence,

$$n = \frac{n_0}{1+(n_0-1)/N} \approx \frac{n_0}{1+(n_0/N)}$$

Consider a similar example as in (1), let

$$d = 0.05, \quad p = 0.5, \quad \alpha = 0.05, \quad t = 2$$

$$n_0 = \frac{4 \times 0.5 \times 0.5}{0 - 0.0025} = 400$$

If the population of students in the school distract is 4000, then

$$n = \frac{n_0}{1+(n_0-1)/N} = \frac{400}{1+399/4000} \approx 364$$

(3) The formula for n in with continuous data In the continuous case, we want to control the relative error r in the estimated population total or mean. For a simple random sample having mean $\bar{x}$, we wish

$$P_r\left(\left|\frac{\bar{x}-\bar{X}}{\bar{X}}\right| \geq r\right) = P_r\left(\left|\bar{x}-\bar{X}\right| \geq r\bar{X}\right) = \alpha$$

to be a small probability. We assume that $\bar{x}$ is normally distributed, from Section 3, we have

$$V(\bar{x}) = E(\bar{x}-\bar{X})^2 = \frac{S^2(N-n)}{nN}, \text{ or}$$

$$\sigma_{\bar{x}} = \sqrt{\frac{N-n}{n}}\frac{S}{\sqrt{n}}$$

Thus,

$$r\bar{X} = t\sigma_{\bar{x}} = t\sqrt{\frac{N-n}{N}}\frac{S}{\sqrt{n}}$$

Hence, solving for n, we have

$$n = \left(\frac{tS}{r\bar{X}}\right)^2 \Bigg/ \left[1+\frac{1}{N}\left(\frac{tS}{r\bar{X}}\right)^2\right]$$

For the first approximation we take

$$n_0 = \left(\frac{tS}{r\bar{X}}\right)^2$$

Hence,

$$n = \frac{n_0}{1+(n_0/N)}.$$

Example: Evergreen Nursery is producing southern pines for sale. It is often to estimate its healthy products in the late winter in order to accept orders. The data were obtained from a bed of southern pine seedlings 1 foot wide and 400 feet long. The sampling unit was 1 foot of the length of the bed, so that N = 400. By completion the observation of the bed, it is found that $\bar{X}$ = 19, S^2 = 86.3. With simple random sampling, how many units must be taken to estimate $\bar{X}$ within 10% apart from a chance of 1 in 20? By the formulas above , we have

$$n_0 = \frac{t^2 S^2}{r^2 \bar{X}} = \frac{4 \times 86.3}{1.9^2} \approx 96$$

Hence,

$$n = \frac{96}{1+96/400} \approx 78.$$

6. Stratified Random Sampling

A stratified sampling of population size of N units is first divided it into subpopulations of $N_1, N_2, ..., N_L$ units respectively. These subpopulations are non-overlapping, and together they compose of the whole population, so that

$$N_1 + N_2 + \cdots + N_L = N$$

The subpopulations are called strata and each of these subpopulations is called a stratum. To obtain the full benefit from stratification, the values of the N_i must be known. When the strata have been determined, a sample is drawn from each stratum, the drawings being made independently in different strata. The sample sizes within each stratum are denoted by $n_1, n_2, ..., n_L$, respectively. If a simple random sample is taken in each stratum, the whole procedure is as stratified random sampling. Stratified random sampling is a common sampling technique. There are many reasons for this; the principal ones are the following.

(1) If the precision of known data are wanted for certain subdivisions of the population, it is advisable to treat each subdivision as a population in its own right.

(2) For some administrative convenience, it may dictate the use of stratification; for example, the agency

conducting the survey may have regional offices, each of which can supervise the survey for a part of its own population.

(3) Sampling problems may differ from different parts of the population. In exampling human populations, people living in institutions (e.g., dormitory, hotels, hospitals, prisons) are often placed in a different stratum from people living in ordinary homes because a different approach to the sampling is appropriate for the two situations. In sampling businesses we may possess a list of the large firms, which are placed in a separate stratum from ordinary smaller firms. Some type of area sampling may have to be used for the smaller firms.

(4) Stratified random sampling may produce a better precision in the estimates of characteristics of the whole population. It may possible to divide the whole heterogeneous population into subpopulations, each of which is internally homogeneous.

7. Systematic Sampling

This method of sampling is quite different from a simple random sampling. We may consider the N units in the population are numbered 1 to N in some order. To select a sample of n units, we take a unit at random from the first k units and every k-th unit thereafter. For instance, if k is 20 and if the first unit drawn is number 7, the subsequent units are numbers 27, 47, 67, and so on. The selection of the first unit determines the whole sample. This type is called an every k-th systematic sample. Actually, a systematic sample is a simple random sample of one cluster unit from a population of k cluster units. The advantages of this method over simple random sampling are as follows.

1. It is easier to draw a sample without mistakes and substantial saving in time.

2. The systematic sampling seems likely to be more precise than simple random sampling. In effect, it stratifies the population into n strata, the sample consists of the first k units, the second k units, and so on.

   We might therefore expect the systematic sample to be about as precise as the corresponding stratified random sample with one unit per stratum. The difference is that with the systematic sample the units occur at the same relative position in the stratum.

3. The systematic sample is spread more evenly over the population, and this fact has sometimes made systematic sampling considerably more precise than stratified random sampling. Sometimes, we may take the sample at the center element of each stratum. We consider that that the data at the central location is often better representing its group that a random selected one.

We like to use the example from Cochran's Sampling Techniques, third edition pages 210-211, to compare the variances of the three sampling techniques, simple random sampling, stratified sampling, and systematic sampling. The data are from a small population with steady rising trend with $N = 40$, $k = 10$, and $n = 4$.

Systematic sample numbers

Strata	1	2	3	4	5	6	7	8	9	10	Means
1	0	1	1	2	5	4	7	7	8	6	4.1
2	6	8	9	10	13	12	15	16	16	17	12.1
3	18	19	20	20	24	23	25	28	29	27	23.3
4	26	30	31	31	33	32	35	37	38	38	33.1
Total	50	58	61	63	75	71	82	88	91	88	72.7

Analysis Variances	df	ss	ms
Between rows	3	4828.3	
Within strata	36	485.5	$13.49 = S^2_{wst}$
Total	39	5313.8	$136.25 = S^2$

$$V_{sy} = \frac{1}{k}\sum_{i=1}^{k}(\bar{x}_{i.} - \bar{X})^2 = \frac{1}{n^2 k}\sum_{i=1}^{k}\left(n\bar{x}_{i.} - n\bar{X}\right)^2$$

$$= \frac{1}{160}\left[(50)^2 + (58)^2 + \ldots + (88)^2 - \frac{(727)^2}{10}\right] = 11.63$$

$$V_{ran} = \left(\frac{N-n}{N}\right)\frac{S^2}{n} = \frac{9}{10}\frac{136.25}{4} = 30.66$$

$$V_{st} = \left(\frac{N-n}{N}\right)\frac{S^2_{wst}}{n} = \frac{9}{10}\frac{13.49}{4} = 3.04$$

These results indicate that both systematic random sampling and stratified sample are much more effective than simple random sampling.

8. Cluster Sampling

The cluster sampling is to study some of numerous methods of sample selection and estimation that have been produced for cluster units of unequal size. An Example of Sampling with probability proportional to size is given.

In a cluster sampling, usually to select the units with probabilities proportional to their sizes M_i, this method can be illustrated by the following.

Unit	Size M_i	$\sum M_i$	Range
1	10	10	1-10
2	5	15	11-15
3	13	28	16-28
4	20	48	29-48

To select a unit is to draw a random number between 1 and 48. If a random number 20 is drawn, it falls in the unit 3, which covers from numbers 16 to 28, inclusive. With this method, the probability that any unit is selected is proportional to its unit size.

9. Conclusion

In this tutorial paper, we have briefly introduced various sampling techniques that are often used in the data mining for processing data objects and making business decisions. Our focus was to provide basic definitions or procedures for each sampling technique, to determine the sample sizes of various techniques, and to give some important statistical features, particularly the sample mean and variance that can characterize the properties of the population.

Statistical sampling theory is probably one of the oldest branches of statistics. In the last century, there has been much progress in its application and theory. In the early twentieth century, much new theory was developed and applied in the survey of social, economic, , and agricultural data for conducting statistical analysis and making decisions. In the second half of the twentieth century, survey sampling techniques have been used to solve quality control problems and in clinical trial studies of new medicines. During the last ten years, sampling techniques have been widely used in the data mining and bioinformatics areas.

Due to the page limitations, we are not able to cover all currently available sampling techniques for this tutorial. For those interested audiences, please reference some of the standard graduate level textbooks on sampling theory.

Bibliographies

Cochran, W. G., *Sampling Techniques*, 3rd ed., John Wiley & Sons. New York, 1977.

Curtiss, J. H., *Lectures on the Theory of Industrial Sampling*, New York University, Institute of mathematical Sciences, 1955.

Deming, W. E. Some Theory of Sampling Theory, John Wiley & Sons. New York, 1950.

Govindarajulu, Z., *Elements of Sampling Theory & Methods*, Prentice Hall, 1999.

Jessen, R. J., *Statistical Survey Techniques,* John Wiley & Sons, New York, 1978.

Neimark, E. D. & Estes, W. K., *Stimulus Sampling Theory*, Holden-Day, 1967.

Pitard, F. F., PierreGy's Sampling Theory & Sampling Practice: V.1, CRC Press, 1989.

Raj, Des, *Sampling Theory*, McGraw-Hill, 1968.

Sukhatme Pandurang Vasudeo, *Sampling Theory of surveys with Applications*, Iowa State Univ. Press, 1984.

Tsypkin, Y. Z., *Sampling Systems Theory and its Applications*, Macmillan, 1964.

Wetherill, G. B., *Sampling Inspection and Quality Control,* 2nd ed., Chapman and Hall, 1977.

Yamanc, T., *Elementary Sampling Theory*, Prentic –Hall, 1967.

Granular Computing as a Basis for Consistent Classification Problems

Y.Y. Yao, J.T. Yao
Department of Computer Science, University of Regina
Regina, Saskatchewan, Canada S4S 0A2
E-mail: {yyao, jtyao}@cs.uregina.ca

Abstract

Within a granular computing model of data mining, we reformulate the consistent classification problems. The granulation structures are partitions of a universe. A solution to a consistent classification problem is a definable partition. Such a solution can be obtained by searching a particular partition lattice. The new formulation enables us to precisely and concisely define many notions, and to present a more general framework for classification.

1 Introduction

As a recently renewed research topic, granular computing (GrC) is an umbrella term to cover any theories, methodologies, techniques, and tools that make use of granules (i.e., subsets of a universe) in problem solving [7, 9, 10]. Formal concept analysis may be considered as a concrete model of granular computing. It deals with the characterization of a concept by a unit of thoughts consisting of two parts, the intension and extension of the concept [1, 6]. The intension of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities which are instances of the concept.

Recently, a granular computing model for knowledge discovery and data mining is proposed by combining results from formal concept analysis and granular computing [8]. Each granule is viewed as the extension of a certain concept and a description of the granule is an intension of the concept. Knowledge discovery and data mining, especially rule mining, can be viewed as a process of forming concepts and finding relationships between concepts, in terms of their intensions and extensions.

The objective of this paper is to apply the granular computing model for the study of the consistent classification problems with respect to partitions of a universe. We re-express, precisely and concisely, many notions based on partition lattice. Our reformulation of the consistent classification problems not only provides a formal treatment of the problem, but also brings more insights into the solution of the problem.

2 Formal Concept Analysis, Granular Computing, and Data Mining

A close connection between formal concept analysis, granular computing, and data mining can be established by focusing on their two fundamental and related tasks, namely, concept formation and concept relationship identification [8].

In the study of formal concepts, every concept is understood as a unit of thoughts that consists of two parts, the intension and extension of the concept [1, 6]. The intension (comprehension) of a concept consists of all properties or attributes that are valid for all those objects to which the concept applies. The extension of a concept is the set of objects or entities which are instances of the concept. All objects in the extension have the same properties that characterize the concept. In other words, the intension of a concept is an abstract description of common features or properties shared by elements in the extension, and the extension consists of concrete examples of the concept. A concept is thus described jointly by its intension and extension.

Basic ingredients of granular computing are subsets, classes, and clusters of a universe [7, 10]. There are many fundamental issues in granular computing, such as granulation of the universe, description of granules, relationships between granules, and computing with granules. Granulation of a universe involves the decomposition of the universe into parts, or the grouping of individual elements into classes, based on available information and knowledge. The construction of granules may be viewed as a process of identifying extensions of concepts. Similarly, finding a description of a granule may be viewed as searching for intension of a concept. Demri and Orlowska referred to such processes as the learning of extensions of concepts and learning of intensions of concepts, respectively [1]. The relationships between concepts, such as sub-concepts, disjoint and

overlap concepts, and partial sub-concepts, can be inferred from intensions and extensions,

It may be argued that some of the main tasks of knowledge discovery and data mining are concept formation and concept relationship identification [2, 8, 9]. The results from formal concept analysis and granular computing can be immediately applied.

In a recently proposed granular computing model of data mining, it is assumed that information about a set of objects is given by an information table [8]. That is, an object is represented by a set of attribute-value pairs. A logic language is defined for the information table. The semantics of the language is given in the Tarski's style through the notions of a model and satisfiability. The model is an information table. An object satisfies a formula if the object has the properties as specified by the formula. Thus, the intension of a concept given by a formula of the language, and extension is given the set of objects satisfying the formula. This formulation enables us to study formal concepts in a logic setting in terms of intensions and also in a set-theoretic setting in terms of extensions.

In the following sections, we will demonstrate the application of the granular computing model for the study of a specific data mining problem known as the consistent classification problems.

3 Partition Lattices of Information Tables

In this section, we study the structures of several families of partitions.

3.1 Partition lattice

A partition provides a simple granulated view of a universe.

Definition 1 *A partition of a set U is a collection of nonempty, and pairwise disjoint subset of U whose union is U. The subsets in a partition are called blocks.*

When U is a finite set, a partition $\pi = \{X_i \mid 1 \leq i \leq m\}$ of U consists of a finite number of blocks. In this case, the conditions for partitions can be simply stated by:

$$\begin{aligned} &\text{(i).} && \text{each } X_i \text{ is nonempty,} \\ &\text{(ii).} && \text{for all } i \neq j, X_i \cap X_j = \emptyset, \\ &\text{(iii).} && \bigcup\{X_i \mid 1 \leq i \leq m\} = U. \end{aligned}$$

There is a one-to-one correspondence between partitions of U and equivalence relations (i.e., reflexive, symmetric, and transitive relations) on U. Each equivalence class of the equivalence relation is a block of the corresponding partition. In this paper, we will use partitions and equivalence relations, and blocks and equivalence classes interchangeably.

Definition 2 *A partition π_1 is refinement of another partition π_2, or equivalently, π_2 is a coarsening of π_1, denoted by $\pi_1 \preceq \pi_2$, if every block of π_1 is contained in some block of π_2.*

The refinement relation is a partial ordering of the set of all partitions. Given two partitions π_1 and π_2, their meet, $\pi_1 \wedge \pi_2$, is the largest partition that is a refinement of both π_1 and π_2, their join, $\pi_1 \vee \pi_2$, is the smallest partition that is a coarsening of both π_1 and π_2. An equivalence classes of the meet are all nonempty intersections of an equivalence class from π_1 and an equivalence class from π_2, equivalence classes of the join are the smallest subsets which are exactly a union of equivalence classes from π_1 and π_2. Under these operations, the poset is a lattice called the partition lattice, denoted by $\Pi(U)$.

3.2 Information tables

An information table provides a convenient way to describe a finite set of objects called the universe by a finite set of attributes [4, 9].

Definition 3 *An information table is the following tuple:*

$$S = (U, At, \mathcal{L}, \{V_a \mid a \in At\}, \{I_a \mid a \in At\}),$$

where

U is a finite nonempty set of objects,
At is a finite nonempty set of attributes,
$\mathcal{L}$ is a language defined using attributes in At,
V_a is a nonempty set of values for $a \in At$,
$I_a : U \rightarrow V_a$ is an information function.

Each information function I_a is a total function that maps an object of U to exactly one value in V_a.

An information table represents all available information and knowledge. That is, objects are only perceived, observed, or measured by using a finite number of properties. We can easily extend the information function I_a to subsets of attribues. For a subset $A \subseteq At$, the value of an object x over A is denoted by $I_A(x)$.

Definition 4 *In the language $\mathcal{L}$, an atomic formula is given by $a = v$, where $a \in At$ and $v \in V_a$. If ϕ and ψ are formulas, then so are $\neg\phi$, $\phi \wedge \psi$, and $\phi \vee \psi$.*

The semantics of the language $\mathcal{L}$ can be defined in the Tarski's style through the notions of a model and satisfiability. The model is an information table S, which provides interpretation for symbols and formulas of $\mathcal{L}$.

Definition 5 *The satisfiability of a formula ϕ by an object x, written $x \models_S \phi$ or in short $x \models \phi$ if S is understood, is defined by the following conditions:*

$$
\begin{aligned}
&(1) \quad x \models a = v \text{ iff } I_a(x) = v,\\
&(2) \quad x \models \neg\phi \text{ iff not } x \models \phi,\\
&(3) \quad x \models \phi \wedge \psi \text{ iff } x \models \phi \text{ and } x \models \psi,\\
&(4) \quad x \models \phi \vee \psi \text{ iff } x \models \phi \text{ or } x \models \psi.
\end{aligned}
$$

If ϕ is a formula, the set $m_S(\phi)$ defined by:

$$m_S(\phi) = \{x \in U \mid x \models \phi\}, \tag{1}$$

is called the meaning of the formula ϕ in S. If S is understood, we simply write $m(\phi)$.

The meaning of a formula ϕ is therefore the set of all objects having the property expressed by the formula ϕ. In other words, ϕ can be viewed as the description of the set of objects $m(\phi)$. Thus, a connection between formulas of $\mathcal{L}$ and subsets of U is established.

With the introduction of language $\mathcal{L}$, we have a formal description of concepts. A concept definable in an information table is a pair $(\phi, m(\phi))$, where $\phi \in \mathcal{L}$. More specifically, ϕ is a description of $m(\phi)$ in S, the intension of concept $(\phi, m(\phi))$, and $m(\phi)$ is the set of objects satisfying ϕ, the extension of concept $(\phi, m(\phi))$.

To illustrate the idea developed so far, consider an information table given by Table 1, which is adopted from Quinlan [5]. The following expressions are some of the formulas of the language $\mathcal{L}$:

$$
\begin{aligned}
&\mathbf{height} = \text{tall},\\
&\mathbf{hair} = \text{dark},\\
&\mathbf{height} = \text{tall} \wedge \mathbf{hair} = \text{dark},\\
&\mathbf{height} = \text{tall} \vee \mathbf{hair} = \text{dark}.
\end{aligned}
$$

The meanings of the formulas are given by:

$$
\begin{aligned}
&m(\mathbf{height} = \text{tall}) = \{o_3, o_4, o_5, o_6, o_7\},\\
&m(\mathbf{hair} = \text{dark}) = \{o_4, o_5, o_7\},\\
&m(\mathbf{height} = \text{tall} \wedge \mathbf{hair} = \text{dark}) = \{o_4, o_5, o_7\},\\
&m(\mathbf{height} = \text{tall} \vee \mathbf{hair} = \text{dark}) = \{o_3, o_4, o_5, o_6, o_7\}.
\end{aligned}
$$

By pairing intensions and extensions, we can obtain formal concepts such as $(\mathbf{height} = \text{tall}, \{o_3, o_4, o_5, o_6, o_7\})$ and $(\mathbf{height} = \text{tall} \wedge \mathbf{hair} = \text{dark}, \{o_4, o_5, o_7\})$.

3.3 Definable partition lattices

In an information table, some objects have the same description and hence can not be differentiated. With the indiscernibility of objects, a subset of objects may have to be considered as a whole rather than individuals. Consequently, for an arbitrary subset of the universe, $X \subseteq U$, it may be impossible to find a concept $(\phi, m(\phi))$ such that $m(\phi) = X$. For example, in Table 1, o_4 and o_5 have the same description. The subset $\{o_4, o_5\}$ must be considered a unit. It is impossible to find a formula whose meaning is $\{o_4, o_6, o_7\}$. In the case where we can precisely describe a subset of objects X, the description may not be unique. That is, there may exist two formulas such that $m(\phi) = m(\psi) = X$. For example, the two formulas:

$$
\begin{aligned}
&\mathbf{class} = +,\\
&\mathbf{hair} = \text{red} \vee (\mathbf{hair} = \text{blond} \wedge \mathbf{eyes} = \text{blue}),
\end{aligned}
$$

have the same meaning set $\{o_1, o_3, o_6\}$. Those observations lead us to consider only certain families of partitions from $\Pi(U)$.

Object	**height**	**hair**	**eyes**	**class**
o_1	short	blond	blue	+
o_2	short	blond	brown	-
o_3	tall	red	blue	+
o_4	tall	dark	blue	-
o_5	tall	dark	blue	-
o_6	tall	blond	blue	+
o_7	tall	dark	brown	-
o_8	short	blond	brown	-

Table 1. An information table

Definition 6 *A subset $X \subseteq U$ is called a definable granule in an information table S if there exists at least one formula ϕ such that $m(\phi) = X$.*

Definition 7 *A partition π is called a definable partition in an information table S if every equivalence class is a definable granule.*

In information table 1, two objects o_4 and o_5, as well as o_2 and o_8, have the same description and are indistinguishable. Consequently, the smallest definable partition is $\{\{o_1\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_6\}, \{o_7\}\}$. The partition $\{\{o_1, o_2, o_3, o_4\}, \{o_5, o_6, o_7, o_8\}\}$ is not a definable partition.

If π_1 and π_2 are definable partitions, $\pi_1 \wedge \pi_2$ and $\pi_1 \vee \pi_2$ are definable partitions. The set of all definable partitions $\Pi_{\mathrm{D}}(U)$ is a sub-lattice of $\Pi(U)$.

In many machine learning algorithms, one is only interested in formulas of certain form. Suppose we restrict the connectives of language $\mathcal{L}$ to only the disjunction connective $\wedge$. Each formula is a disjunction of atomic formulas and such a formula is referred to as a disjunctor.

Definition 8 *A subset $X \subseteq U$ is a disjunctively definable granule in an information table S if there exists a disjunctor ϕ such that $m(\phi) = X$. A partition π is called a disjunctively definable partition if every equivalence class is a disjunctively definable granule.*

The partition $\{\{o_1, o_2, o_6, o_8\}, \{o_3, o_4, o_5, o_7\}\}$ is a definable partition but not a disjunctively definable partition in the information table 1. The join of two disjunctively definable partitions $\{\{o_1, o_2, o_8\}, \{o_3, o_4, o_5, o_6, o_7\}\}$ and $\{\{o_1, o_2, o_6, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}$ is $\{U\}$, which is not a disjunctively definable partition.

The meet, $\pi_1 \wedge \pi_2$, of two disjunctively definable partitions is a disjunctively definable partition. However, the join, $\pi_1 \vee \pi_2$, is not necessarily a disjunctively definable partition. In this case, we only obtain a meet semi-lattice $\Pi_{\mathrm{DD}}(U)$.

A lattice related to $\Pi_{\mathrm{DD}}(U)$ is the lattice formed by partitions defined by various subsets of At. For a subset of attributes A, we can define an equivalence relation E_A as follows:

$$\begin{aligned} xE_Ay &\iff \text{for all } a \in A, I_a(x) = I_a(y) \\ &\iff I_A(x) = I_A(y). \end{aligned} \tag{2}$$

For the empty set, we obtain the coarsest partition $\{U\}$. For a nonempty subset of attributes, the induced partition is disjunctively definable. The family of partition defined by subsets of attributes form a lattice $\Pi_{\mathrm{AD}}(U)$, which is not necessarily a sub-lattice of $\Pi(U)$.

For the information table 1, we obtain the following partitions with respect to subsets of the attributes.

$$\begin{aligned}
\pi_0 :&\quad \emptyset, \\
&\quad \{U\}, \\
\pi_1 :&\quad \{\mathbf{height}\}, \\
&\quad \{\{o_1, o_2, o_8\}, \{o_3, o_4, o_5, o_6, o_7\}\}, \\
\pi_2 :&\quad \{\mathbf{hair}\}, \\
&\quad \{\{o_1, o_2, o_6, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\}, \\
\pi_3 :&\quad \{\mathbf{eyes}\}, \\
&\quad \{\{o_1, o_3, o_4, o_5, o_6\}, \{o_2, o_7, o_8\}\}, \\
\pi_4 :&\quad \{\mathbf{height}, \mathbf{hair}\}, \\
&\quad \{\{o_1, o_2, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}, \{o_6\}\}, \\
\pi_5 :&\quad \{\mathbf{height}, \mathbf{eyes}\}, \\
&\quad \{\{o_1\}, \{o_2, o_8\}, \{o_3, o_4, o_5, o_6\}, \{o_7\}\}, \\
\pi_6 :&\quad \{\mathbf{hair}, \mathbf{eyes}\}, \\
&\quad \{\{o_1, o_6\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_7\}\}, \\
\pi_7 :&\quad \{\mathbf{height}, \mathbf{hair}, \mathbf{eyes}\}, \\
&\quad \{\{o_1\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_6\}, \{o_7\}\}.
\end{aligned}$$

Since each subset defines a different partition, the partition lattice has the same structure as the lattice defined by the power set of the three attributes $\mathbf{height}$, $\mathbf{hair}$, and $\mathbf{eyes}$.

All the notions developed in this section can be defined relative to a particular subset $A \subseteq At$ of attributes. A subset $X \subseteq U$ is called a definable granule with respect to a subset of attributes $A \subseteq At$ if there exists a least one formula ϕ over A such that $m(\phi) = X$. A partition π is called a definable partition with respect to a subset of attributes A if every equivalence class is a definable granule with respect to A. Let $\Pi_{\mathrm{D}(A)}(U)$, $\Pi_{\mathrm{DD}(A)}(U)$, and $\Pi_{\mathrm{AD}(A)}(U)$ denote the partition (semi-) lattices with respect to a subset of attributes $A \subseteq At$, respectively. We have the following connection between partition (semi-) lattices:

$$\begin{aligned}
&\Pi_{\mathrm{AD}}(U) \subseteq \Pi_{\mathrm{DD}}(U) \cup \{U\} \subseteq \Pi_{\mathrm{D}}(U) \subseteq \Pi(U), \\
&\Pi_{\mathrm{AD}(A)}(U) \subseteq \Pi_{\mathrm{DD}(A)}(U) \cup \{U\} \subseteq \Pi_{\mathrm{D}(A)}(U) \subseteq \Pi(U).
\end{aligned}$$

They provide a formal framework of classification problems.

4 Classification as Partition Lattice Search

Classification problem is one of the well studied problems in machine learning and data mining. In this section, we reformulate the classification problem using partition lattice.

4.1 Formulation of the problem

In supervised classification, it is assumed that each object is associated with a unique class label. Objects are divided into disjoint classes which form a partition of the universe. We further assume that information about objects are given by an information table. Without loss of generality, we assume that there is a unique attribute $\mathbf{class}$ taking class labels as its value. The set of attributes is expressed as $At = C \cup \{\mathbf{class}\}$, where C is the set of attributes used to describe the objects. The goal is to find classification rules of the form, $\phi \Longrightarrow \mathbf{class} = c_i$, where ϕ is a formula over C and c_i is a class label.

Let $\pi_{\mathbf{class}} \in \Pi(U)$ denote the partition induced by the attribute $\mathbf{class}$. An information table with a set of attributes $At = C \cup \{\mathbf{class}\}$ is said to provide a consistent classification if all objects with the same description over C have the same class label, namely, if $I_C(x) = I_C(y)$, then $I_{\mathbf{class}}(x) = I_{\mathbf{class}}(y)$. Using the concept of partition lattice, we immediately have the equivalent definition.

Definition 9 *An information table with a set of attributes $At = C \cup \{\mathbf{class}\}$ is a consistent classification problem if and only if there exists a partition $\pi \in \Pi_{\mathrm{AD}(C)}(U)$ such that $\pi \preceq \pi_{\mathbf{class}}$.*

In the rest of this paper, we restrict our discussion to the consistent classification problem.

Definition 10 *The solution to a consistent classification problem is a definable partition π such that $\pi \preceq \pi_{\mathbf{class}}$. For a pair of equivalence classes $X \in \pi$ and $Y \in \pi_{\mathbf{class}}$ with $X \subseteq Y$, we can derive a classification rule $\phi(X) \Longrightarrow \phi(Y)$, where $\phi(X)$ and $\phi(Y)$ are the formulas whose meaning sets are X and Y, respectively.*

For the information table 1, the definable partition,

$$\{\{o_1, o_6\}, \{o_2, o_7, o_8\}, \{o_3\}, \{o_4, o_5\}\} \preceq \{\{o_1, o_3, o_6\}, \{o_2, o_4, o_5, o_7, o_8\}\} = \pi_{\mathbf{class}},$$

is a solution to the classification problem. The classification rules corresponding to the solution are given by:

$$\begin{aligned}
&\mathbf{hair} = \text{blond} \wedge \mathbf{eyes} = \text{blue} \Longrightarrow \mathbf{class} = +,\\
&\mathbf{eyes} = \text{brown} \Longrightarrow \mathbf{class} = -,\\
&\mathbf{hair} = \text{red} \Longrightarrow \mathbf{class} = +,\\
&\mathbf{hair} = \text{dark} \wedge \mathbf{eyes} = \text{blue} \Longrightarrow \mathbf{class} = -.
\end{aligned}$$

The left hand side of a rule is a formula whose meaning is a block of the solution partition. For example, for the first rule, we have $m(\mathbf{hair} = \text{blond} \wedge \mathbf{eyes} = \text{blue}) = \{o_1, o_6\}$.

For a consistent classification problem, the partition defined by all attributes in C is the smallest partition in the three definable partition lattices. Let π_A denote the partition defined by a subset $A \subseteq C$ of attributes. The smallest partition π_C is a trivial solution to the consistent classification problem.

Depending on the particular partition lattice used, one can easily establish properties of the family of solution partitions. Let $\Pi_\alpha(U)$, where $\alpha = \text{AD}(C), \text{DD}(C), \text{D}(C)$, denote a (semi-) lattice of definable partitions. Let $\Pi^S_\alpha(U)$ be the corresponding set of all solution partitions. We have:

(i). For $\alpha = \text{AD}(C), \text{DD}(C), \text{D}(C)$, if $\pi' \in \Pi_\alpha(U)$, $\pi \in \Pi^S_\alpha(U)$ and $\pi' \preceq \pi$, then $\pi' \in \Pi^S_\alpha(U)$;

(ii). For $\alpha = \text{AD}(C), \text{DD}(C), \text{D}(C)$, if $\pi', \pi \in \Pi^S_\alpha(U)$, then $\pi' \wedge \pi \in \Pi^S_\alpha(U)$;

(iii). For $\alpha = \text{D}(C)$, if $\pi', \pi \in \Pi^S_\alpha(U)$, then $\pi' \vee \pi \in \Pi^S_\alpha(U)$;

It follows that the set of all solution partitions form a lattice or meet semi-lattice.

Mining classification rules can be formulated as a search for a partition from a partition lattice. A definable lattice provides the search space of potential solutions, and the partial order of the lattice provides the search direction. The standard search methods, such as depth-first search, breadth-first search, bounded depth-first search, and heuristic search, can be used to find a solution from a lattice of definable partitions. Depending on the required properties of rules, one may use different definable partition lattice introduced earlier. For example, by search the semi-lattice $\Pi_{\text{DD}(C)}(U)$, we can obtain classification rules whose left hand sides are only disjunction of atomic formulas. The well known ID3 learning algorithm in fact searches $\Pi_{DD(C)}(U)$ for classification rules [5]. By searching the lattice $\Pi_{\text{AD}(C)}(U)$, one can obtain a similar solution.

We can re-express many fundamental notions of classification in terms of partitions.

Definition 11 *For two solutions $\pi_1, \pi_2 \in \Pi_\alpha$ of a consistent classification problem, namely, $\pi_1 \preceq \pi_{\mathbf{class}}$ and $\pi_2 \preceq \pi_{\mathbf{class}}$, if $\pi_1 \preceq \pi_2$, we say that π_1 is a more specific solution than π_2, or equivalently, π_2 is a more general solution that π_1.*

Definition 12 *A solution $\pi \in \Pi_\alpha$ of a consistent classification problem is called the most general solution if there does not exists another solution $\pi' \in \Pi_\alpha$, $\pi \neq \pi'$, such such that $\pi \preceq \pi' \preceq \pi_{\mathbf{class}}$.*

In the information table 1, consider three partitions:

$$\begin{aligned}
\pi_1 :&\quad \{\{o_1\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5\}, \{o_6\}, \{o_7\}\},\\
\pi_2 :&\quad \{\{o_1, o_6\}, \{o_2, o_8\}, \{o_3\}, \{o_4, o_5, o_7\}\},\\
\pi_3 :&\quad \{\{o_1, o_6\}, \{o_2, o_7, o_8\}, \{o_3\}, \{o_4, o_5\}\}.
\end{aligned}$$

from the lattice $\Pi_{\text{DD}(C)}(U)$. We have $\pi_1 \preceq \pi_2 \preceq \pi_{\mathbf{class}}$ and $\pi_1 \preceq \pi_3 \preceq \pi_{\mathbf{class}}$. Thus, π_1 is a more specific solution than both π_2 and π_3. In fact, π_2 and π_3 are two most general solutions.

For a consistent classification problem, the partition defined by all attributes in C is the smallest partition in Π_α. Thus, a most general solution always exists. However, a most general solution may not be unique. There may exist many more general solutions.

The roles played by each attribute, well studied in the theory of rough sets [4], can be re-expressed as follows.

Definition 13 *An attribute $a \in C$ is called a core attribute if $\pi_{C-\{a\}}$ is not a solution to the consistent classification problem.*

Definition 14 *An attribute $a \in C$ is called a superfluous attribute if $\pi_{C-\{a\}}$ is a solution to the consistent classification problem, namely, $\pi_{C-\{a\}} \preceq \pi_{\mathbf{class}}$.*

Definition 15 *A subset $A \subseteq C$ is called a reduct if π_A is a solution to the consistent classification problem and π_B is not a solution for any proper subset $B \subset A$.*

For a given consistent classification problem, there may exist more than one reduct.

In the information table 1, attributes $\mathbf{hair}$ and $\mathbf{eyes}$ are core attributes. Attribute $\mathbf{height}$ is a superfluous attribute. The only reduct is the set of attributes $\{\mathbf{hair}, \mathbf{eyes}\}$.

4.2 ID3 type search algorithms

ID3 type learning algorithms can be formulated as a heuristic search of the semi-lattice $\Pi_{\mathrm{DD}(C)}(U)$. The heuristic used is based on an information-theoretic measure of dependency between the partition defined by $\mathbf{class}$ and another disjunctively definable partition with respect to the set of attributes C. Roughly speaking, the measure quantifies the degree to which a partition $\pi \in \Pi_{\mathrm{DD}(C)}(U)$ satisfies the condition $\pi \preceq \pi_{\mathbf{class}}$ of a solution partition.

Specifically, the direction of ID3 search is from coarsest partitions of $\Pi_{\mathrm{DD}(C)}(U)$ to more refined partitions. Largest partitions in $\Pi_{\mathrm{DD}(C)}(U)$ are the partitions defined by single attributes in C. Using the information-theoretic measure, ID3 first selects a partition defined by a single attribute. If an equivalence class in the partition is not a disjunctively definable granule with respect to $\mathbf{class}$, the equivalence class is further divided into smaller granules by using an additional attribute. The same information-theoretic measure is used for the selection of the new attribute. The smaller granules are disjunctively definable granules with respect to C. The search process continues until a partition $\pi \in \Pi_{\mathrm{DD}(C)}(U)$ is obtained such that $\pi \preceq \pi_{\mathbf{class}}$.

4.3 Rough set type search algorithms

Algorithms for finding a reduct in the theory of rough sets can be viewed as heuristic search of the partition lattice $\Pi_{\mathrm{AD}(C)}(U)$. Two directions of search can be carried, either from coarsening partitions to refinement partitions or from refinement partitions to coarsening partitions.

The smallest partition in $\Pi_{\mathrm{AD}(C)}(U)$ is π_C. By dropping an attribute a from C, one obtains a coarsening partition $\pi_{C-\{a\}}$. Typically, a certain fitness measure is used for the selection of the attribute. The process continues until no further attributes can be dropped. That is, we find a subset $A \subseteq C$ such that $\pi_A \preceq \pi_{\mathbf{class}}$ and $\neg(\pi_B \preceq \pi_{\mathbf{class}})$ for all proper subsets $B \subset A$. The resulting set of attributes A is a reduct.

The largest partition in $\Pi_{\mathrm{AD}(C)}(U)$ is $\pi_\emptyset$. By adding an attribute a, one obtains a refined partition π_a. The process continues until we have a partition satisfying the condition $\pi_A \preceq \pi_{\mathbf{class}}$. The resulting set of attributes A is a reduct.

5 Conclusion

The granular computing model for data mining is used to reformulate the consistent classification problems. We explore the structures of partitions of a universe. The consistent classification problems are expressed as the relationships between partitions of the universe. Three definable partition lattices are introduced. Depending on the properties of classification rules, a solution to a consistent classification problem is a definable partition in one of the lattices. Such a solution can be obtained by searching that lattice. Our formulation is similar to the well established version space search method for machine learning [3].

The new formulation enables us to precisely and concisely define many notions, and to present a more general framework for classification. To illustrate its the potential usefulness and generality, we briefly describe the ID3 and rough set learning algorithms using the proposed model.

References

[1] Demri, S. and Orlowska, E. Logical analysis of indiscernibility, in: *Incomplete Information: Rough Set Analysis*, Orlowska, E. (Ed.), Physica-Verlag, Heidelberg, pp. 347-380, 1998.

[2] Fayyad, U.M. and Piatetsky-Shapiro, G. (Eds.) *Advances in Knowledge Discovery and Data Mining*, AAAI Press, 1996.

[3] Mitchell, T.M. Generalization as search, *Artificial Intelligence*, **18**, 203-226, 1982.

[4] Pawlak, Z. *Rough Sets, Theoretical Aspects of Reasoning about Data*, Kluwer Academic Publishers, Dordrecht, 1991.

[5] Quinlan, J.R. Learning efficient classification procedures and their application to chess end-games, in: *Machine Learning: An Artificial Intelligence Approach*, Vol. 1, Michalski, J.S., Carbonell, J.G., and Mirchell, T.M. (Eds.), Morgan Kaufmann, Palo Alto, CA, pp. 463-482, 1983.

[6] Wille, R. Concept lattices and conceptual knowledge systems, *Computers Mathematics with Applications*, **23**, 493-515, 1992.

[7] Yao, Y.Y. Granular computing: basic issues and possible solutions, *Proceedings of the 5th Joint Conference on Information Sciences*, pp.186-189, 2000.

[8] Yao, Y.Y. On modeling data mining with granular computing, *Proceedings of COMPSAC 2001*, pp.638-643, 2001.

[9] Yao, Y.Y. and Zhong, N. Potential applications of granular computing in knowledge discovery and data mining, *Proceedings of World Multiconference on Systemics, Cybernetics and Informatics*, pp.573-580, 1999.

[10] Zadeh, L.A. Towards a theory of fuzzy information granulation and its centrality in human reasoning and fuzzy logic, *Fuzzy Sets and Systems*, **19**, 111-127, 1997.

Object Mining in Image Data Using Neural Networks

Mengjie Zhang
School of Mathematical and Computing Sciences
Victoria University of Wellington
P.O. Box 600, Wellington, New Zealand
mengjie@mcs.vuw.ac.nz

Abstract

Neural networks have been widely applied to data mining since the late 1980s. However, they are often criticised and regarded as a "black box" due to the lack of interpretation ability. This paper describes a domain independent approach to the use of neural networks for mining multiple class objects in large images and shows neural networks are not just a black box.

In this approach, the networks use a square input field which is large enough to contain every single object of interest and are trained by the back propagation algorithm on examples which have been cut out from the large images. The trained networks are then applied, in a moving window fashion, over the large images to mine/detect the objects of interest. During the mining process, both the classes and locations of the objects are determined. This approach has been examined on three multiple class object mining problems of increasing difficulty. The results suggest that this approach can be used to mine simple and regular objects with translation and limited rotation invariance in large images against a relatively uniform background.

Visualisation of the learned network weights not only gives an intuitive way of representing hidden patterns encoded in neural networks for object mining problems, but also shows that neural networks are an expression or a model of hidden patterns extracted/discovered during the data mining process.

Keywords Network training, data mining, object recognition, object detection, weight visualisation.

1 Introduction

As more and more data is collected as electronic form, the need for data mining is increasing extraordinarily. Due to the high tolerance to noisy data and the ability to classify unseen data on which they have not been trained, neural networks have been widely applied to data mining [2, 4, 5].

However, neural networks have been criticised for their poor interpretability, since it is difficult for humans to interpret the symbolic meaning behind the learned network weights. For this reason, a neural network is often regarded as a black box classifier or a prediction engine [4]. In this paper, we argue that it is not always true, particularly for data mining in image data. We use the "weight matrices" to represent/interpret the "hidden patterns" in learned networks for multiple class object mining problems.

This paper addresses the problem of mining a number of different kinds of small objects in a set of large images. The common characteristic of such problems can be phrased as "Given $subimage_1, subimage_2$, ..., $subimage_n$ which are examples of the object of interest, find all images which contain this object and its location(s)". Figure 1 shows examples of problems of this kind. In the problem illustrated by Figure 1 (b), we want to find centers of all of the 5 cent and 20 cent coins and determine whether the head or the tail side is up. Examples of other problems of this kind include target detection problems [3, 10] where the task is to find, say, all tanks, trucks or helicopters in an image. Unlike most of the current work in the object mining/detection area, where the task is to find only objects of one class [3, 6, 7], our objective is to mine objects from a number of classes.

Neural networks have been applied to object classification and mining problems [1, 6, 9]. In these approaches, various features/attributes such as brightness, colour, size and perimeter are extracted from the sub-images of the objects and used as inputs to the networks. These features are usually quite different and specific for different problem domains. Extracting and selecting good features is very time consuming and programs for feature extraction and selection often need to be hand-crafted. The approach described in this paper directly uses raw pixels as inputs to the networks.

1.1 Multiclass Object Mining

Multiclass object mining here refers to the detection of small objects of a number of classes in large images. It

consists of both *object classification*, which determines the classes of the objects of interest, and *object localisation*, which identifies the positions of all the objects in the large images. This problem is also known as *multiclass object detection*.

Performance in object mining is measured by detection rate and false alarm rate. The detection rate is the number of objects correctly reported as a percentage of the total number of real objects and false alarm rate is the number of non-objects incorrectly reported as objects as a percentage of the total number of real objects. It is important to note that mining/detecting objects in images with very cluttered backgrounds is an extremely difficult problem and that false alarm rates of 200-2,000% are common [7, 9]. Also note that most research which has been done in this area so far either only presents the results of object classification where all the objects have been properly localised and segmented, or only gives the object localisation results where all objects of interest belong to a single class. The results presented in this paper are the performance for the whole mining/detection task (both object localisation and classification).

1.2 Goals

The overall goal of this paper is to investigate a domain independent approach to the use of neural networks for mining multiple class objects in large images and to investigate a way of interpreting weights in learned networks. Instead of using specific image features, this approach directly uses raw image pixels as inputs to neural networks. This approach is detailed in Section 3. This approach will be examined on a sequence of object mining/detection problems of increasing difficulty (see Section 2). Specifically, we investigate:

- Will this approach work for a sequence of multiclass object mining problems of increasing difficulty?
- Will the performance deteriorate as the degree of difficulty of the detection problems increases?
- Can the weights in learned networks be interpreted in some ways and "hidden patterns" be successfully discovered and represented?

2 The Image Databases

We used three different databases in the experiments. Example images and key characteristics are given in Figure 1. The images were selected to provide object mining problems of increasing difficulty. Database 1 (Easy) was generated to give well defined objects against a uniform background. The pixels of the objects were generated using a Gaussian generator with different means and variances for each class. There are three classes of small objects of interest in this database: black circles (*class*1), grey squares (*class*2) and white circles (*class*3). The coin images (database 2) were intended to be somewhat harder and were taken with a CCD camera over a number of days with relatively similar illumination. In these images the background varies slightly in different areas of the image and between images and the objects to be detected are more complex, but still regular. There are four object classes of interest: the head side of 5 cent coins (class *head*005), the head side of 20 cent coins (class *head*020), the tail side of 5 cent coins (class *tail*005) and the tail side of 20 cent coins (class *tail*020). All the objects in each class have a similar size. They are located at arbitrary positions and with different rotations. The retina images (database 3) were taken by a professional photographer with special apparatus at a clinic and contain very irregular objects on a highly cluttered background. The objective is to find two classes of retinal pathologies – haemorrhages (class *haem*) and micro aneurisms (class *micro*). Note that in each of the databases the background (*non-object*) counts as a class (class *other*), but not a class of interest.

3 The Method

3.1 Overview

An overview of the approach is briefly outlined as follows:

1. Assemble a database of images in which the locations and classes of all the objects of interest are manually determined. Divide these entire images into two sets: a *detection training set* and a *detection test set*.
2. Determine an appropriate size ($n \times n$) of a square which will cover all single objects of interest and form the input field of the networks. Generate a classification data set by cutting out squares of size $n \times n$ from the detection training set. Each of the squares, called *cutouts* or *sub-images*, only contains a single object and/or a part of the background. Randomly split these cutouts into a classification training set and a classification test set.
3. Determine the network architecture. A three layer feed forward neural network is used in this approach. The $n \times n$ pixel values form the inputs of the training data and the classification is the output.

Number of Images: 7
Number of Objects: 240
Number of Classes: 4
Image Size: 700x700

Easy (Circles and Squares)
(a)

Number of Images: 20
Number of Objects: 600
Number of Classes: 5
Image Size: 640x480

Medium Difficulty (Coins)
(b)

Number of Images: 15
Number of Objects: 164
Number of Classes: 3
Image Size: 1024x1024

Very Difficult (Retinas)
(c)

Figure 1: Object Detection Problems of Increasing Difficulty

4. Train the network by the back propagation algorithm [8] on the classification training data. The trained network is then tested on the classification test set to measure the object classification performance. This step is designed to find the best trained network for object mining/detection.

5. Use the trained network as a moving window template to mine/detect the objects of interest in the detection test set. If the output of the network for a class exceeds a given threshold then report an object of that class at the current location.

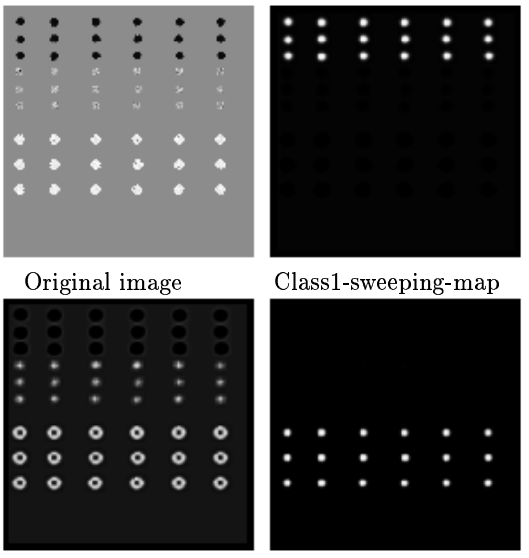
6. Evaluate the object mining performance of the network by calculating the detection rate and the false alarm rate.

3.2 Object Mining/Detection

While object classification corresponds to network training and network testing on the cutouts in the classification data sets, object mining/detection corresponds to network sweeping on the entire images in the detection test set, which were not used in any form for network training. *Network sweeping* involves both object classification and localisation.

During network sweeping, the successfully trained neural network is used as a template matcher, and is applied, in a moving window fashion, over the large images to detect the objects of interest. The template is swept across and down these large images, pixel by pixel in every possible location.

After the sweeping process is done, an *object sweeping map* for each detected object class will be produced. An object sweeping map corresponds to a grey level image. Sample object sweeping maps for *class1*, *class2* and *class3* together with the original image for the easy detection problems are shown in Figure 2. During the sweeping process, if there is no match between a square in an image and the template, then the neural network output is 0, which corresponds to black in the sweeping maps. A partial match corresponds to grey on the centre of the object, and a good match is close to white.



Original image | Class1-sweeping-map

Class2-sweeping-map | Class3-sweeping-map

Figure 2: Sample object sweeping maps in object mining/detection.

In the sweeping maps, if a pixel value at a location is greater than or equal to a *threshold*, then report an object at that location. If two or more "objects"

for different classes at the same position are found, the decision will be made according to the network activations at this position. For example, if one object for *class2* and one for *class3* at position *(260, 340)* for the easy detection problem are reported and the activations for the three classes of interest and the background at this position are `(0.27, 0.57, 0.83, 0.23)`, then the object for *class3* will be considered the mined/detected object at this position since the activation for this class is the biggest (`0.83`).

4 Results

4.1 Object Classification Results

To classify the object cutouts for a particular problem, the number of hidden nodes of the network needs to be empirically determined. We tried a series of numbers of hidden nodes for the three object mining problems: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 15, 20, 30, 50, 100, 150, 200, 300, and 500. The experiments indicated that 196-4-4,[1] 576-3-5 and 256-5-3 with a set of learning parameters gave the best performance for the easy, the coin, and the retina problems, respectively.

Network training and testing results for object classification are presented in Table 1. In all cases, the network training and testing procedure was repeated 15 times and the average results are presented. Line 1 shows that the best network for the easy images is 196-4-4, the average number of epochs used to train the network and get it converged is 199.40, and the trained network can achieve 100% accuracy on the cutouts of both the classification training set and test set. For the coin images, we can also achieve the ideal performance for object classification. However, this is not the case for the retina images, where only 71.83% accuracy was obtained on the test set of the cutouts.

Image Databases	Network Architecture	Training Epochs	Training Accuracy	Test Accuracy
Easy Images	196-4-4	199.40	100%	100%
Coin Images	576-3-5	234.6	100%	100%
Retina Images	256-5-3	475.8	81.62%	71.83%

Table 1: Object classification — Network training and testing results for the three databases.

4.2 Object Mining/Detection Results

This section describes the detection performance of this approach on the three detection problems. For each problem, the 15 trained networks obtained in object classification are used to mine/detect objects in the detection test set and the average results are presented.

The best detection rates and the corresponding false alarm rates for the three classes in the easy images are presented in Table 2. The best detection rates achieved for all the three classes are 100%, showing that this approach can successfully detect all the objects of interest in this database. At this point, detecting *class1* and *class 3* did not produce any false positives, while detecting *class2* resulted in a 91.2% false alarm rate on average.

Easy Images	Object Classes		
	Class1	Class2	Class3
Detection Rate(%)	100	100	100
False Alarm Rate(%)	0	91.2	0

Table 2: Object mining/detection results for the easy images.

The detection results for the coin images are described in Table 3. As in the easy images, the trained neural networks achieved 100% detection rates for all the classes, showing that this approach correctly detects all the objects of interest from different classes in the coin images. In each run, it was always possible to find a threshold for the network output for class *head005* and *tail005* which resulted in detecting all of the objects of these classes with no false alarms. However, detecting classes *head020* and *tail020* was a relatively difficult problem. The average false alarm rates for the two classes at a 100% detection rate were 182% and 37.5% respectively.

Coin Images	Object Classes			
	head005	tail005	head020	tail020
Detection Rate (%)	100	100	100	100
False Alarm Rate (%)	0	0	182	37.5

Table 3: Object mining/detection results for the coin images.

Compared with the performance of the easy and coin images, the results of the very difficult retina images are disappointing. All the objects of class *micro* were correctly detected (the detection rate is 100%), however, with a very high false alarm rate of 10104%. The best detection rate for class *haem* was only 73.91%. with a high false alarm rate of 2859%.

4.3 Analysis of Results

As can be seen from the detection results obtained here, it was always possible to detect all objects of interest in the easy and the coin images. This reflects the fact that the objects in the two databases

[1] 196-4-4 refers to a feed forward network architecture with 196 input nodes, 4 hidden nodes and 4 output nodes. In this paper, we use a similar way to express other network architectures.

are simple or regular and the background is uniform or relatively uniform. While detecting objects in the easy images only resulted in a few false alarms, detecting objects in the coin images resulted in a relatively higher false alarm rate. This is mainly because the mining/detection problems in the coin images are more difficult than in the easy images.

Due to the high degree of difficulty of the detection problems, the results for the retina images are not good. For class *micro*, while all objects were correctly detected, a very high number of false alarms were produced. This is mainly because these objects are irregular and complex and the background is highly cluttered. For class *haem*, it was not possible to detect all objects of interest (the best detection rate was 73.91%). This is mainly due to the size variance of these objects (from 7×7 to 14×14 pixels). Another reason that we did not obtain good results on the retina images is the insufficient number of training object examples.

Comparing the results obtained in object classification and object mining/detection, it can be found that object classification results are better than the object mining results for all the three databases. This is in turn proved that multiclass object mining/detection task is generally much more difficult than only the classification task on the same problem domains.

5 Visualisation and Analysis of Learned Network Weights

This section interprets the network internal behaviour through visual analysis of the weights in the trained networks. For presentation convenience, we use the trained networks for regular object detection in the coin images. Most other networks contained similar patterns.

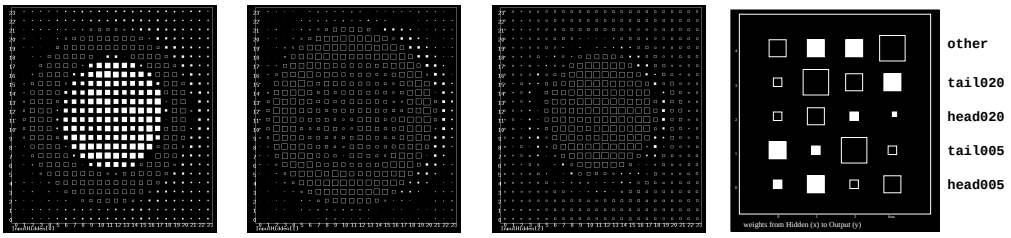
Figure 3 shows the weights from a trained 576-3-5 network which has been successfully applied to the coin images. In this figure the full squares represent positive weights and the outline squares represent negative weights, while the size of the square is proportional to the magnitude of the weight. Matrices (a), (b) and (c) show the weights from the input nodes to the first, the second and the third hidden nodes. The weights are shown in a 24×24 matrix to facilitate visualisation. Figure 3 (d) shows the weights from the hidden nodes to the output nodes and the biases of these output nodes. The five rows in this matrix correspond to the classes *other, tail020, head020, tail005* and *head005*. The first three of the four columns correspond to weights from the three hidden nodes (associated with weight matrices (a), (b) and (c)) to the five output nodes. The last column corresponds to the biases of the five output nodes.

Inspection of the first column of Figure 3 (d) reveals that weight matrix (a) has a positive influence on 5 cent coins and a negative influence on 20 cent coins. It has a strong influence on class *tail005* but a weak influence on class *head005*. The same matrix has a strong negative effect on class *other* (background). Inspection of the second column reveals weight matrix (b) has a positive effect on the 5 cent coins and a negative effect on the 20 cent coins. Moreover, it has a strong influence on class *head005* but a week influence on class tail005. Also it has a very strong positive influence on class *other*. This indicates that the combination of matrices (a) and (b) not only can separate the 5 cent coins from the 20 cent coins and the background, but also can discriminate the 5 cent tails from the 5 cent heads. Inspection of the third column reveals that the weight matrix (c) has a strong negative influence on the tails of both 5 cent and 20 cent coins and a week influence on the heads of both 5 cent and 20 cent coins. It also strongly supports the background and has a strong negative influence on the tails of 5 cent coins. The fourth column suggests that the biases also play a complementary role for mining objects, particularly for class *tail020*, class *other* and class *head005*. If we regard the nodes of the hidden layers as representing feature detectors learnt by the network, then Figures 3 (a)-(c) are a visual representation of these features. Visually these features 'make sense' as there are regions corresponding to the 5 cent coins, the annulus remaining to the background when a 5 cent coin is 'removed' from the centre of a 20 cent coin.

For object mining in large images described here, the weight matrices can be considered "hidden patterns" encoded in learned neural networks. Visualisation of the learned network weights revealed that patterns contained in learned networks can be intuitively represented, which strongly supports the idea that neural networks are not just a "black box", but a model or an expression of patterns discovered during learning.

6 Conclusions

The goal of this paper is to investigate a domain independent neural approach to mining multiple class objects in large images. This goal was achieved by using raw pixel values as inputs to neural networks. The experimental results showed that this approach performed very well for mining a number of simple and regular objects against a relatively uniform background. It did not perform well on the difficult detection problems in the retina images, which indicates that it can not be well suited to detecting complex and irregular objects against a highly cluttered background. As expected,



(a) (b) (c) (d)

Figure 3: Weights in a trained network for object mining/detection in the coin images.

the performance degrades when the approach is applied to object mining problems of increasing difficulty.

Visualisation of the weights in trained neural networks resulting from this approach revealed that trained networks contained feature detectors which "made sense" for the problem domain and could discriminate objects from different classes. This provides a way of revealing hidden patterns in learned neural networks for object mining problems. This approach also shows that neural networks are not just a black box, but a model of patterns which were discovered through the learning process.

The approach has the following characteristics:

- Raw image pixel data are used as inputs to neural networks, and accordingly traditional specific feature extraction and selection is avoided.
- It is a domain independent approach and can be directly applied to multiclass object detection problems in different areas.
- Multiple class objects can be mined/detected (classified and localised) in large images with a single trained neural network.
- Patterns encoded in learned neural networks can be visually represented by using weight matrices.
- This approach can mine translation and limited rotation invariant objects but cannot successfully detect objects with the size invariance such as class *haem* in the retina images.

Acknowledgement

I would like to thank Dr. Peter Andreae and Dr. Victor Ciesielski for useful discussions and Chris Kamusinski who provided and labelled the retina images.

References

[1] David P. Casasent and Leonard M. Neiberg. Classifier and shift-invariant automatic target recognition neural networks. *Neural Networks*, Volume 8, Number 7/8, pages 1117–1129, 1995.

[2] Usama Fayyad, Gregory Piatetsky-Shapiro and Padhraic Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, Volume 17, Number 3, pages 37–53, 1996.

[3] Paul D. Gader, Joseph R. Miramonti, Yonggwan Won and Patrick Coffield. Segmentation free shared weight neural networks for automatic vehicle detection. *Neural Networks*, Volume 8, Number 9, pages 1457–1473, 1995.

[4] Robert Groth. *Data Mining: Building Competitive Advantage*. Prentice Hall PTR, 2000.

[5] Jiawei Jan and Michaeline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[6] H. L. Roitblat, W. W. L. Au, P. E. Nachtigall, R. Shizumura and G. Moons. Sonar recognition of targets embedded in sediment. *Neural Networks*, Volume 8, Number 7/8, pages 1263–1273, 1995.

[7] Michael W. Roth. Survey of neural network technology for automatic target recognition. *IEEE Transactions on neural networks*, Volume 1, Number 1, pages 28–43, March 1990.

[8] D. E. Rumelhart, G. E. Hinton and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart, J. L. McClelland and the PDP research group (editors), *Parallel distributed Processing, Explorations in the Microstructure of Cognition, Volume 1: Foundations*, Chapter 8. The MIT Press, Cambridge, Massachusetts, London, England, 1986.

[9] Mukul V. Shirvaikar and Mohan M. Trivedi. A network filter to detect small targets in high clutter backgrounds. *IEEE Transactions on Neural Networks*, Volume 6, Number 1, pages 252–257, Jan 1995.

[10] Allen M. Waxman, Michael C. Seibert, Alan Gove, David A. Fay, Ann Marie Bernandon, Carol Lazott, William R. Steele and Robert K. Cunningham. Neural processing of targets in visible, multispectral ir and sar imagery. *Neural Networks*, Volume 8, Number 7/8, pages 1029–1051, 1995.

Foundations of Data Mining
A Position Paper

Dr. Bhavani Thuraisingham
The MITRE Corporation
(at present with the National Science Foundation)

Data Mining is the process of posing queries to large amounts of data sources and extracting patterns and trends using statistical and machine learning techniques. It integrates various technologies including database management, statistics and machine learning. Data mining has applications in numerous disciplines including medical, financial, defense and intelligence. Data mining tasks include classification, clustering, making associations and anomaly detection. For example, data mining can extract various associations between people, places or words. During recent years there have been many developments in data mining. Various data mining techniques have been developed. These include techniques for extracting associations, neural networks, inductive logic programming, decision trees, fuzzy logic and rough sets. Furthermore, data mining has gone beyond mining relational databases to mining text and multimedia data. Also, data mining is being applied to areas such as information security and intrusion detection.

While thee have been many practical developments, we still have major challenges. One of the most important challenges is scalability. For example, the data mining techniques often seem to work on small quantities of data. But do these techniques work for say petabyte sized databases? If data mining is to be useful we need to mine very large databases. Therefore, it is critical that we need to understand the limitations of the data mining algorithms. To understand the limitations, we need to study the foundations of data mining. We need to explore the time and space complexity of the algorithms. For example, can these algorithms be completed in polynomial time? Are there any undecidable problems? If the problems are decidable what is the complexity of the problems? To date little research has been carried out on the foundations of data mining. There are techniques such as inductive logic programming and rough sets that have underpinnings in logic and mathematics. One needs to explore these techniques for data mining and examine the computational complexity aspects. We also need to understand the complexity of the various search algorithms being used for market basket analysis.

We are at a time where data mining applications are exploding. Data mining is becoming one of the fastest growing fields in computer science. We are also at a time where we have a much better understanding of the problem. Therefore, the time is right to start active research on the foundations of data mining. But we also have to be cautious. That is, we need to carry out incremental research. In some cases, we may not be able to explore the foundations. Therefore, we have to be very selective. We need to decide which of the techniques are to be examined and then carry out theoretical research step by step.