**Workshop Notes**

# Towards the Foundation of Data Mining
# Vol 2

A Workshop held at
the 6th Pacific-Asia Conference on Knowledge Discovery and Data Mining

May 6th, 2002

# 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

The amount of digital data processed by computers grows extremely fast. In real-world applications of E-commerce, inestimable amount of transaction records are stored in databases. Therefore, how to discover useful knowledge hidden in large databases is very important.

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. For example, we can analyze the personal information (age, debt, etc.) of applicants for credit cards and find some rules to classify them into an acceptance set and a rejection set. If the classification rate of these rules is accurate enough, the classifier may take the place of human checkers, which can avoid mistakes. However, in data mining applications, the computational resources required can become a problem with large data sets. As a result, 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 [1-4] and fuzzy-based design approaches [5-7]. 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 [7]. 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 [3]. 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, as used in [2], [4]. The general fitness function is defined as follows [4]:

   *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. Generally, weighted-sum approaches are able to obtain a good solution. However, most real-world problems are too large to allow the exact single-objective optimal solution to be found, and thus a set of non-dominated solutions are desirable for the decision maker's consideration. Moreover, 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. As a result, it is inefficient to obtain more non-dominated solutions using the weighted-sum approaches in many separate runs.

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. That is the purpose of MRSPs for reducing the size from a large training set. 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 [8-13]. Therefore, in order to cope with the above-mentioned problems simultaneously, 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) [16,17] is applied to solve multi-objective MRSPs directly. By making use of Pareto dominance concept and experimental design methods, IMOEA is capable of solving multi-objective problems in a single run efficiently and accurately. 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

Optimization is a procedure of finding and comparing feasible solutions until no better solutions can be found. Solutions are termed good or bad in terms of an objective. However, there are many real-world problems which cannot satisfactorily be characterized by a single performance measure. For example, consider the design of an engineering system. An optimal design might seek for minimizing the cost while maximizing the quality. Due to the nature of trade-offs involved, multi-objective optimization problems (MOOPs) seldom have a unique solution. Instead, a set of Pareto-optimal solutions is sought. These solutions are optimal in the wider sense that no other solutions in the search space are superior to them when all the multiple objectives are considered. 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*. An example of dominating relations of solutions in bi-objective space is shown in figure 1.

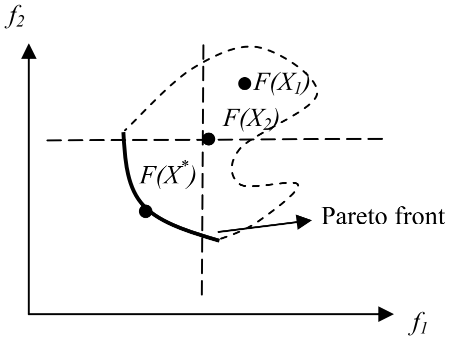


**Fig. 1.** The dominating relations of solutions in bi-objective space in the minimization case. $X_2$ dominates $X_1$ , and $X_2$ is dominated by $X^*$.

The size and shape of Pareto front usually depend on the number of objective functions and interactions among the individual objective functions. If the objectives are conflicting to each other, the resulting Pareto front may be noisy, multi-peaked and discontinues. However, traditional multi-objective optimization approaches are computationally intensive and require a series of separate runs to identify the trade-offs between different design objectives [9]. This essentially means that one needs a robust optimization method that can cope with noisy, multi-peaked performance relations with discontinuities in MOOPs. As a result, GAs and evolutionary algorithms have been recognized to be particularly suitable for solving

MOOPs because their abilities to exploit/explore multiple solutions in parallel and to find an entire set of Pareto-optimal solutions in a single run [8-13].

### 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⊂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).

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

## 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. GPSIFF is a Pareto-based fitness assignment, it is assumed that no information on the preference among objectives is available. The basic idea is that the population is calculated by making direct use of the definition of Pareto concepts, in order to determine the reproduction probability of each individual to generate populations without having to combine the objectives in some way.

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:

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

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 2 illustrate the example of fitness values of twelve participant individuals for a bi-objective optimization problem (c=12). There are three non-dominated solutions represented by the white points. It can be seen that the more individuals one dominates, the higher score one receives. On the contrary, if there are more individuals better than one individual, one will be degraded in proportion to the number of individuals dominate it. For example, considering the individual $A$ with fitness value 13 in figure 2, 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 $3\text{-}2\text{+}12\text{=}13$.

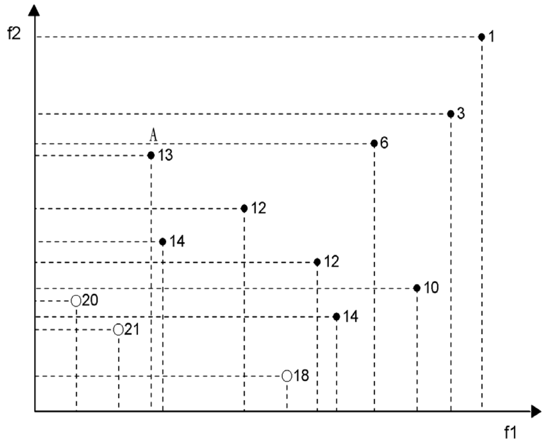


**Fig. 2.** Fitness values of the participant individuals with $c$=12 in the objective space.

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. No additional techniques such as niching methods or specified selection are incorporated.
2. Generality. GPSIFF makes direct use of general Pareto dominance relationships to evaluate the performance of each participant individual. The GPSIFF value is also valid when there is only one objective.
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. The effectiveness arises from that the GPSIFF method assigns each individual a discriminative fitness value while the fitness values of dominated individuals in a cluster are always the same used in the Pareto ranking-based method, such as [12-13].

### 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. The high performance of IC arises from that IC replaces the generate-and-test search for children using a random combination of chromosomes with a systematic reasoning search method using an intelligent combination of selecting better individual genes.

Theoretically analysis and experimental studies for illustrating the superiority of IC with the use of OA and factor analysis can be found in [16-17]. A concise example of illustrating the use of orthogonal array (OA) and factor analysis can be found in [16-18].

#### 3.3.1 OA and Factor Analysis

Briefly, OED makes use of orthogonal arrays (OAs) and factor analysis for determining the combinations of factor levels and for analyzing the experimental results. 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 $\gamma$ factors with two levels for each factor. The total number of experiments is $2^{\gamma}$ 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 $\gamma$ 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 $\omega$ rows and ($\omega$-1) columns, use the first $\gamma$ columns, and ignore the other ($\omega$-$\gamma$-1) columns. Table 1 illustrates an example of OA $L_8(2^7)$. The algorithm of constructing OAs can be found in [19]. OED can reduce the number of experiments for factor analysis. The number of OA experiments required to analyze all individual factors is only $\omega$ or O($\gamma$).

**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, … $\omega$. 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 $S_{jk}$ :

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

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

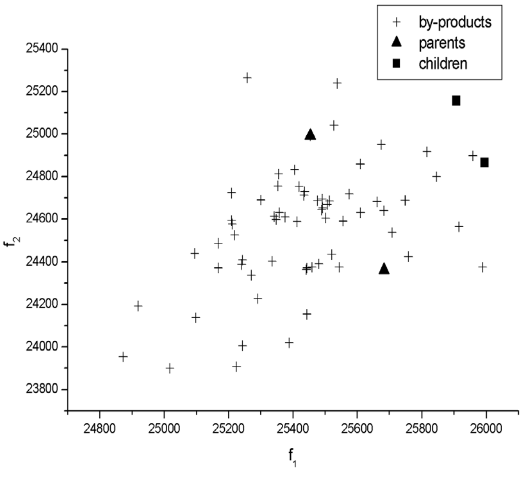


**Fig. 3.** Examples of IC operation in solving a bi-objectives maximization problem.

It takes about $\omega = 2^{\lceil \log_2(\gamma+1)\rceil}$ fitness evaluations for performing an IC operation. The value γ for each IC operation would gradually decrease when evolution proceeds with a decreasing number of non-determinate variables. This behavior can helpfully cope with the large parameter optimization problem of simultaneous improving the classification accuracy and editing reference patterns. Figure 3 illustrates a general example gleaned from IC operation using an OA $L_{64}(2^{63})$ in solving an bi-objective maximization problem. We can see carefully from figure 3 that two children are more promising to be non-dominated individuals rather than the random recombination of their parents as in conventional GA and additional non-dominated solutions may be generated from an IC operation.

### 3.4 Intelligent Multi-objective Evolutionary Algorithm

Conventional GA which is called simple genetic algorithm (SGA) consists of five primary operations: initialization, evaluation, selection, crossover, and mutation. 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 ($\gamma$) 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 $\sigma_{share}$=0.49. The population size and the external population size of SPEA are 40 and 10, respectively. The sharing factor $\sigma_{share}$ 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 4. Simulation results out of thirty runs are summarized in figure 5.

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%). Furthermore, from figure 4 and figure 5, the results illustrate that IMOEA is robust and is capable of generating good non-dominated solution efficiently in this test.

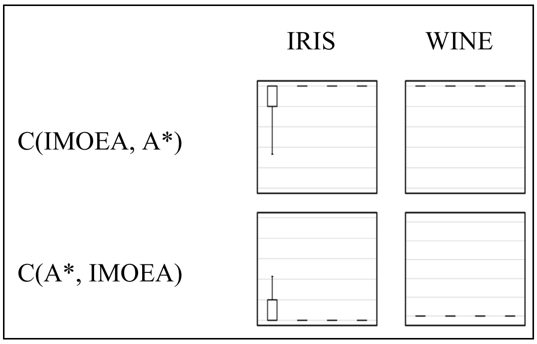


**Fig. 4.** Box plots based on the C measures for MRSPs. Each rectangle contains four box plots representing the distribution of the C measures for a certain ordered pair of algorithms. A* are SPEA, NSGA, NPGA and VEGA, respectively. The scale is 0 at the bottom and 1 at the top per rectangle.

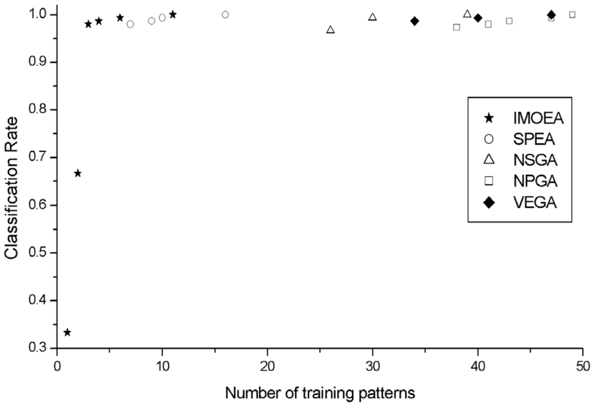


**Fig. 5.** Simulation results out of 30 runs for iris data under $N_{eval}$ = 20000.

### 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 4. The results out of thirty runs are shown in figure 6. Figure 4 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 6, 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%).

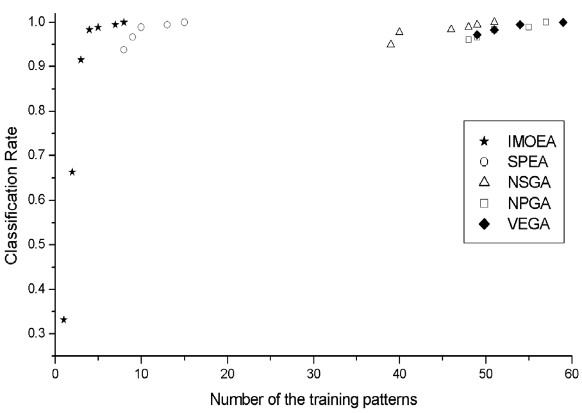


**Fig. 6.** Simulation results out of 30 runs for wine data under $N_{eval} = 30000$.

### 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 7. 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 4-6, 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. The non-dominated solutions obtained by SPEA, NSGA, NPGA and VEGA are all trapped in the biased regions. Except IMOEA and SPEA, NSGA, NPGA and VEGA suffers from both the low convergence speed and low accuracy.

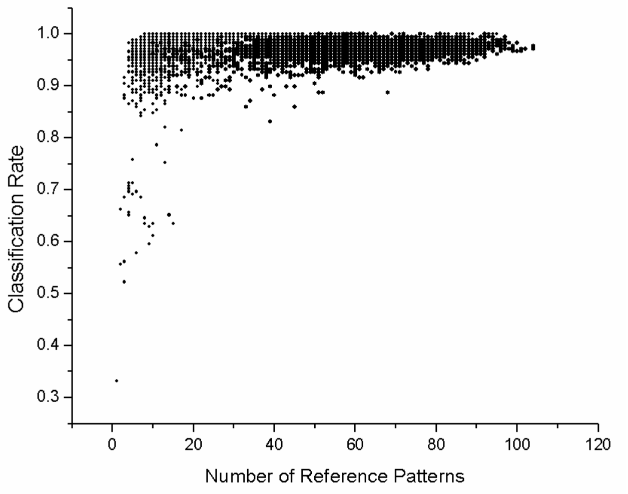


**Fig. 7.** Scatter plots of solutions in solving MRSP with the wine data set. The darker region has higher density of solutions.

## 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. Due to its simplicity, theoretical elegance, generality and superiority, IMOEA can be most widely used for solving MOOPs. 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.: Editing for the k-nearest neighbors rule by a genetic algorithm. Pattern Recognition Letter **16** (1995) 809-814

[2] Kuncheva, L. I.: Fitness function in editing k-NN reference set by genetic algorithms. Pattern Recognition, **30**(6) (1997) 1041-1049

[3] 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

[4] 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

[5] 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

[6] Newton, S. C., Pemmaraju, S., Mitra, S.: Adaptive fuzzy leader clustering of complex data set in pattern recognition. IEEE Trans. Neural Networks 3(5) (1992) 794-800

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

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

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

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

[11] 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

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

[13] 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

[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] 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

[17] 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

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

[19] 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 [36], [37], [40], [44]. 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.

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.

Classification can be thought as the base of ability to knowledge acquisition[24]. Some classification techniques, e.g. decision trees [12], [21], [22], [23], [29], [31], decision lists [10], [33] work well for pattern recognition and process control. Here, we choose these techniques to apply to colon ploype screening analysis [40]. 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 [31], neural networks [34], rule based learners [25], [30], etc., have been applied to producing classifiers for medical decision support systems [41]. 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], [9], [14], [42], 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 [46]. Even by the Quinlan's work [28], the types of uncertainties are not to be probabilistic, appearing as randomness or noise. Pedrycz and Sosnowski [26] 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 [20].

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.

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.

## 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 different 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], [42], [46]. In general, those methods can separated into two types, prefuzzification 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 postfuzzify 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 $C_2$.

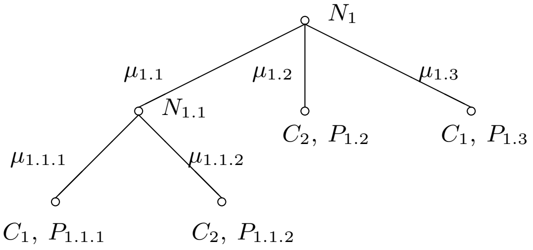


Fig. 1. A sample FCT with $\mathcal{C} = \{C_1, C_2\}$

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], [27], [31].

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 [39]. The foundation of Shannon's formula is based on probability theory. Quinlan [31], 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 [39] 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 $\text{Test}_L$. 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.

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

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 $\alpha$ 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.

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

## 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 [46]. 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 $S_0$.

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

The procedure 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.

## 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

```
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 $\tau_k$ 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 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.

```
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.

```
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.

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 *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.

TABLE I
The error rates of the NTUH checkup data set for polyp screening (1).

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

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).

| Method | Error Rate | |
|---|---|---|
| | *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).

| Method | Error Rate | |
|---|---|---|
| | *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.

## VII. Conclusion

The uncertainties and noise make classification difficult. Missing or imprecise information may prevent a case from being

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* |
| $\leq 0.15$ | 0.0513002 | 0.001686 |
| $\leq 0.1$ | 0.226478 | 0.010175 |

classified at all. It is occurred in the boundaries of the data in two more different classes [17], [32]. 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 [4], [5], [6], [30], [38] 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], [31] 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 [35] 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 [20].

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 $C4.5$.

## 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. Murphy and D. Aha. UCI repository of machine learning databases. Department of Information and Computer Science, University of California at Irvine, 1992.

[20] 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.

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

[22] 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.

[23] 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.

[24] Z. Pawlak. *Rough Sets*. Kluwer Academic, Dordrecht, 1991.

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

[26] 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.

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

[28] J. R. Quinlan. Probabilistic decision trees. In P. Langley, editor, *Proceedings of the Fourth International Workshop on Machine Learning*, Los Altos, CA, 1987.

[29] J. R. Quinlan. Decision trees and decision making. *IEEE Transactions on Systems, Man, and Cybernetics*, 20:339–346, 1990.

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

[31] J. R. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, Los Altos, CA, 1993.

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

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

[34] 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.

[35] 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.

[36] E. Sato, A. Ouchi, and T. Ishidate. Polyps and diverticulosis of large bowel in autopsy population of Akita prefecture, compared with Miyagi: High rate of colorectal cancer in Japan. *Cancer*, 37:1316–1321, 1976.

[37] 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.

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

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

[40] 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.

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

[42] 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.

[43] A. P. White and W. Z. Liu. Bias in information-based measures in decision tree induction. *Machine Learning*, 15:321–329, 1994.

[44] A. R. Williams, B. A. Balasooriya, and D. W. Day. Polyps and cancer of the large bowel: a necropsy study in Liverpool. *Gut*, 23:835–842, 1982.

[45] J. Yerushalmy. Statistical problems in assessing methods of medical diagnosis, with special reference to X-ray techniques. *Public Health Reports*, 62:1432–1449, 1947.

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

# GEC with Generalization and Specialization Mutations Using Varying Partitioning Size on Continuous Attributes

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 increase 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).

Comparing to [8][14] which evolves IF-THEN rules and [3] which evolves fuzzy IF-THEN rules, EGEC evolves a whole population of rules, i.e., gathering all the rules together to form a whole entity. We focus on this metabiosis between the rules. Analogous to a living cell, a single strand of chromosome does not decide how it operates; it is when all the chromosomes gather into DNA strands that decides the behavior of that cell. The emergent behavior (the ability to classify objects) of EGEC occurs only when the rules from different runs are combined. Although there may be some conflicts within the rules themselves, the reported behavior to the outside of this body is consistent (conflicts are resolved internally).

## 2. Encoding the Chromosome

### 2.1. Categorical Data

Enumerating categorical data is not a problem. We use a single bit for each possible value of an attribute to represent it. For an attribute to take *N* possible values, we will have N bits representing it. Consider the following representation of an attribute with 6 possible values in Figure 1 that matches the rule: if one of the enumerated values 2, 4 or 5 is true, then this attribute is satisfied. This mechanism is exactly like GABIL [5]. The difference is that since our mechanism can cope with multi-classes besides binary classes, our representation for the 'class' is also same as the representation of the attributes. This adds flexibility for further development and research.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 1 | 0 | 1 | 1 | 0 |

**Figure 1.** A sample encoding of an attribute having 6 possible values

For a whole chromosome representing a complete rule, consider the case shown in Figure 2. Here we have a example with 3 attributes for which to be classified into 2 classes. Attribute 1 takes 4 possible value, attribute 2 takes 2 and attribute 3 takes 4. Let (a,b) denote the bth value of attribute a being set to true. This rule means: if [(1,1) or (1,4)] and [(2,1)] and [(3,2) or (3,3)] then identify as class 2. In Boolean terminology, our genetic represented rules are in conjunctive normal form (CNF).

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| Attribute 1 | | | | Attribute 2 | | Attribute 3 | | | | Class | | |

**Figure 2.** A sample chromosome representing a complete rule for classification

Clearly, this mechanism also provides a compression of rules. The rule shown in Figure 2 represents 4 separate rules itself, they are:

- If (1,1) and (2,1) and (3,2) then Class 2
- If (1,4) and (2,1) and (3,2) then Class 2
- If (1,1) and (2,1) and (3,3) then Class 2
- If (1,4) and (2,1) and (3,3) then Class 2

This mechanism is an advantage. We can merge similar rules into one rule and thus expand the classifying capability of a single rule.

## 2.2. Micro Partitioning of Continuous Attributes

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.

$$\ldots,\left(\mu-\frac{3}{2}\frac{\rho}{\delta},u-\frac{1}{2}\frac{\rho}{\delta}\right),\left(\mu-\frac{1}{2}\frac{\rho}{\delta},\mu+\frac{1}{2}\frac{\rho}{\delta}\right),\left(\mu+\frac{1}{2}\frac{\rho}{\delta},\mu+\frac{3}{2}\frac{\rho}{\delta}\right),\ldots \tag{1}$$

$$\delta\left\lceil\frac{R}{\rho}\right\rceil+1 \tag{2}$$

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.

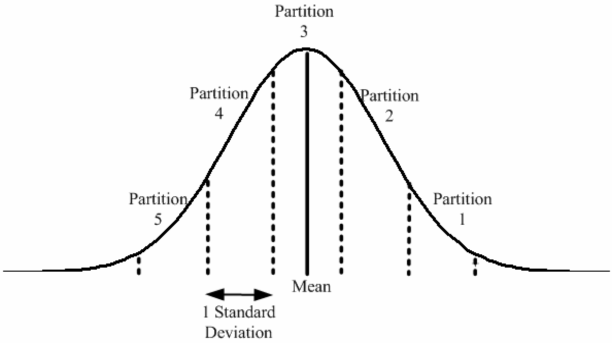


**Figure 3.** A sample chromosome representing a complete rule for classification

Visualization of our partitioning method is shown in Figure 3. 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).

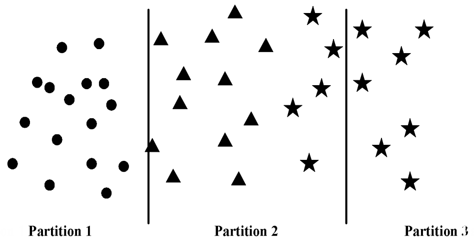


**Figure 4.** A 3 class example with partitioning inadequate resolution

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 4 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 $\delta$. Shown in Figure 5 is the example of increasing $\delta$ 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.

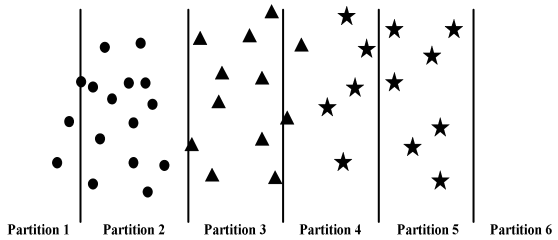


**Figure 5.** Increasing the partitioning resolution of the example in Figure 4

One special case of this encoding mechanism is that when all the bits of an attribute are set to 1, this attribute becomes a "don't care" term.

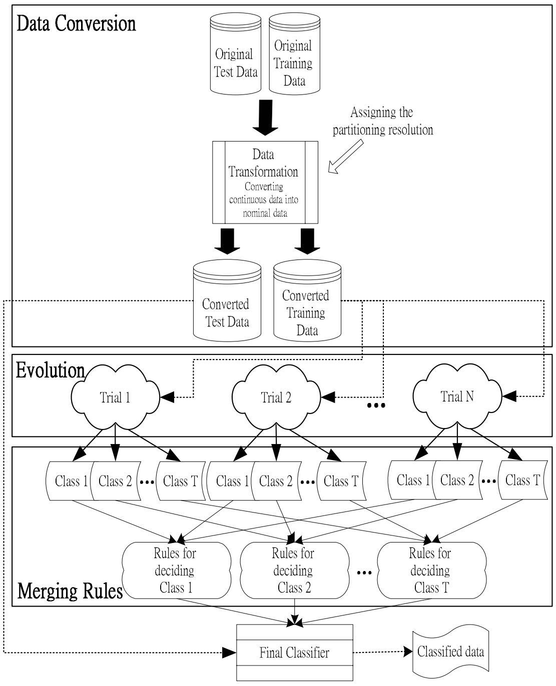


**Figure 6.** Outline of the EGEC

## 3. The Extended Genetic Evolved Classifier

The outline of our EGEC is shown in Figure 6. 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 7. It is based GEC [12] with modifications of adding generalization and specialization mutation operators.

```
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 7.** Outline of the genetic algorithm for evolving classification rules

The algorithm develops equal number of rules for each class in each trial. These rules are expected to cover a certain domain of the whole sample space. We run independent trials to produce rules in hope to obtain the maximum coverage of the sample space. From all these independent trials, we merge the final rules from their output forming a complete classifier. This approach is much like how a shotgun fires its bullets, it scatters around the target in which the probability of a critical hit will increase. Final class of an instance is then decided by majority voting, that is, it receives votes from each rule (deciding which class the instance belongs to), and the class which gets the highest vote from the rules is taken.

### 3.1. Fitness Evaluation

The fitness value is evaluated from formula (3). The ideas of this evaluation method are as follows:

1. Maximize the number of matched correctly (correctly classified) data from the test set.
2. Minimize the number of matched incorrectly (incorrectly classified) data from the test set.
3. Maximize the number of data covered, i.e.,

number of data decided by this certain rule whether matched or mismatched (Unmatched = Total number of data minus both the number of data matched correctly and incorrectly).

$$\text{Fitness} = \frac{\#\,Matched - \#\,Mismatched}{\#\,Unmatched + 1} \qquad (3)$$

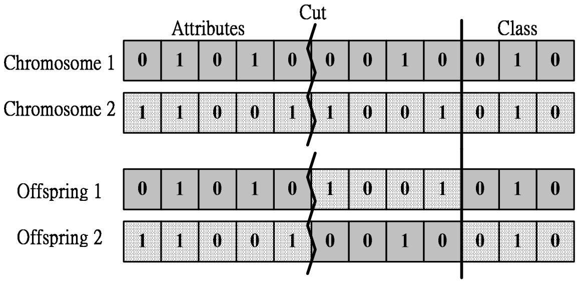
We can see that the more data matched, the less data mismatched, and the more data covered, the larger the fitness value will be. If a rule matches the same number of times as it mismatches, it is considered useless, e.g., fifty-fifty chance is just like flipping a coin. What we want is to let the rule match more often than mismatch. This is also reflected in formula (3).

For rules in the final stage of the algorithm, if it still holds a negative fitness value, then we know that this rule does not work well. It mismatches more than it matches. In this case, we drop this rule in our final classifier.

### 3.2. Genetic Operators

As usual, we do not wish the rule to be bias in any direction or to be dominated by a certain chromosome prematurely. Although uniform selection mechanism is very simple and straightforward, it does provide a fair chance for every member to develop in power. Thus, in EGEC, we use the simplest approach out of all various types of selection mechanisms: uniform selection.

Crossover is done by selecting a random spot on the chromosome and exchanging the two pieces as illustrated in Figure 8. The bits representing the ‘class’ that the rule decides is not considered, i.e., crossover never occurs at these position. This is because EGEC evolves rules for each class separately. If crossover occurs at these points, it would ruin the rule for the certain class.



| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Chromosome 1 | 0 | 1 | 0 | 1 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| Chromosome 2 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Offspring 1 | 0 | 1 | 0 | 1 | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 0 |
| Offspring 2 | 1 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |

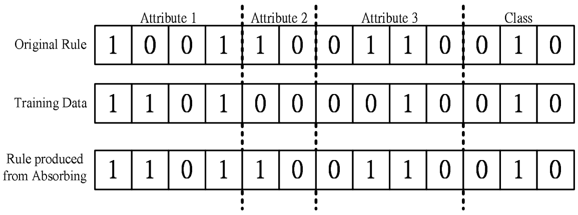
**Figure 8.** An illustration on how the crossover is done

The following mechanism is taken for simple mutation. For each bit in the chromosome, the mutation rate is set to 0.5. The reason for taking such a high mutation rate is that we get a more diverse set of new rules to exploit. Again, the bits representing the ‘class’ is not considered.

The final stage to complete a generation of the GA is selection. We merge the original population, i.e., the parents, and the newly created offsprings together and thus forming a large population pool (according to the outline of Figure 4, our pool size is now 4N). We choose the most suitable N members (highest fitness) for the evolution of the next generation. Notice that EGEC always preserve its elite members.

### 3.3. 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.



| | Attribute 1 | | | | Attribute 2 | | Attribute 3 | | | | Class | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Original Rule | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |
| Training Data | 1 | 1 | 0 | 1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 1 | 0 |
| Rule produced from Absorbing | 1 | 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 | 0 | 0 | 1 | 0 |

**Figure 9.** 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 9. 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 10. 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.

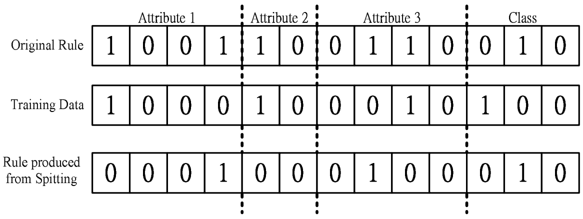


**Figure 10.** The specialization mutation in EGEC

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).

$$x + \sum_{i=1}^{n} a_i \qquad (4)$$

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.

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

With these numbers in mind, we decided 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 11-16 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 11, 12, 14 and 15 that as we decreased the partition size, we get better results. Taking Figure 11 (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. For Figures 13 and 15, especially Figure 13, 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 15 (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.

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, exists in all of the results.

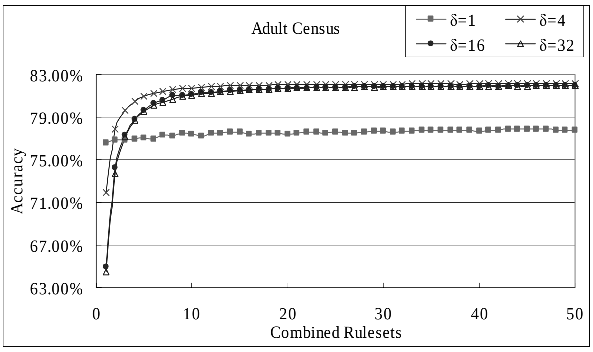


**Figure 11.** Result of EGEC on the adult census database

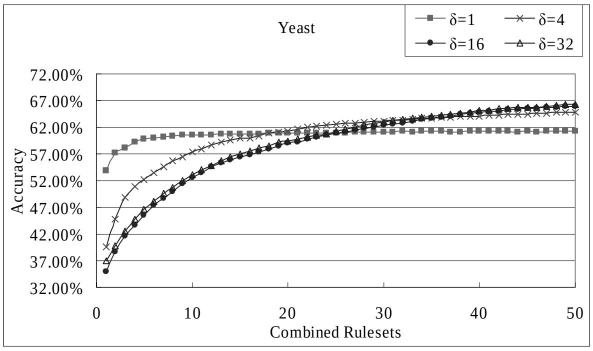


**Figure 12.** Result of EGEC on the yeast database

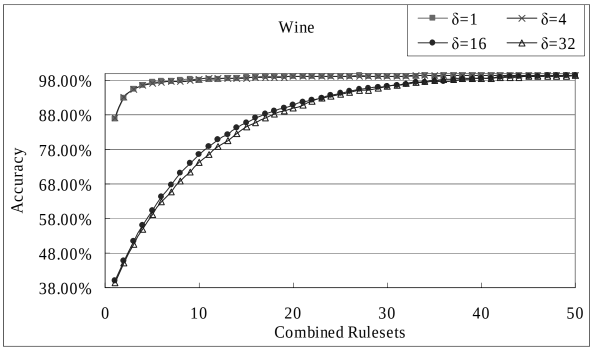


**Figure 13.** Result of EGEC on the wine database

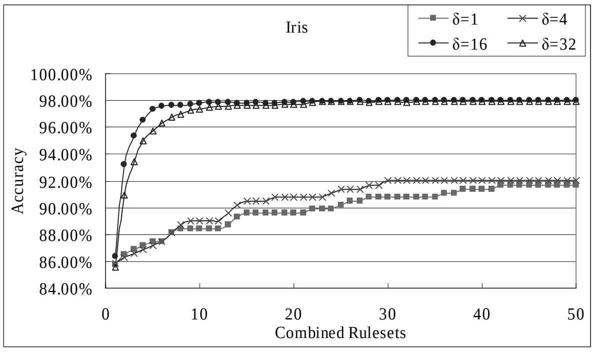


**Figure 14.** Result of EGEC on the iris database

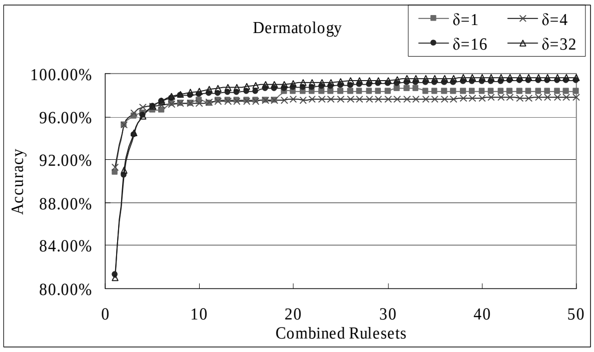


**Figure 15.** Result of EGEC on the dermatology database

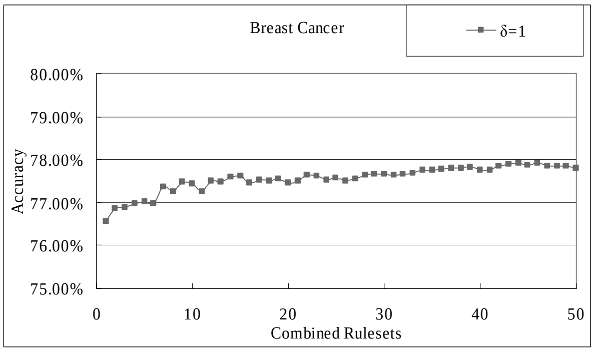


**Figure 16.** Result of EGEC on the breast cancer database

Comparasion 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 Proc. 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 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 $F_\sigma$ 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 $F_\sigma$ 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.

## 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.

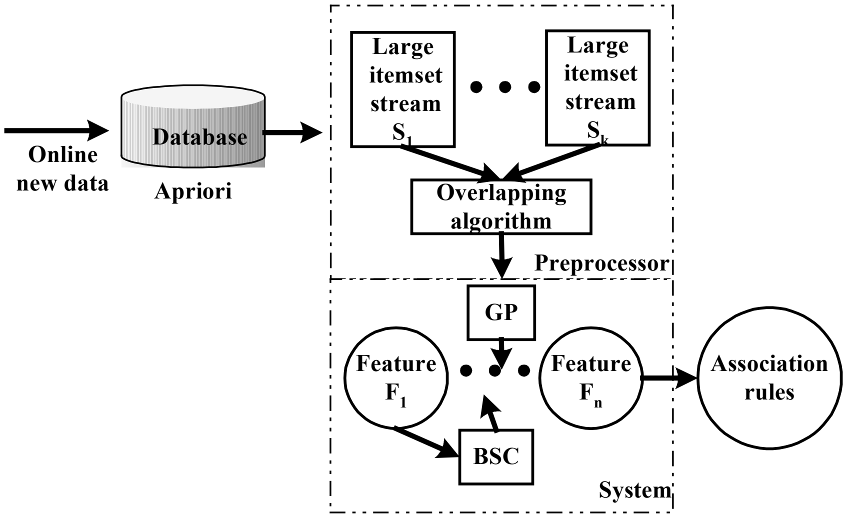


**Figure 1.** The architecture of finding the feature of association rules on huge database in multivariate time series

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.

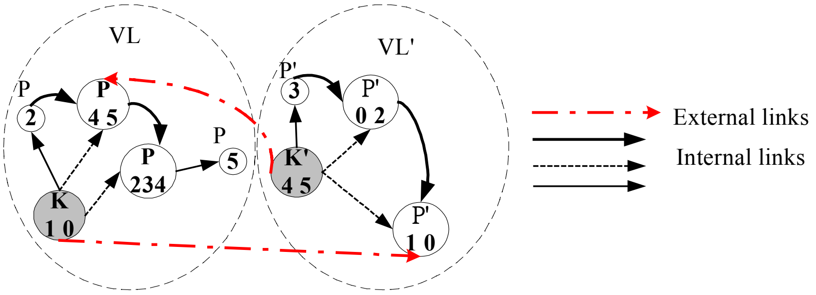


**Figure 2.** The diagram of simulated stereochemistry for virtual lattices

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.

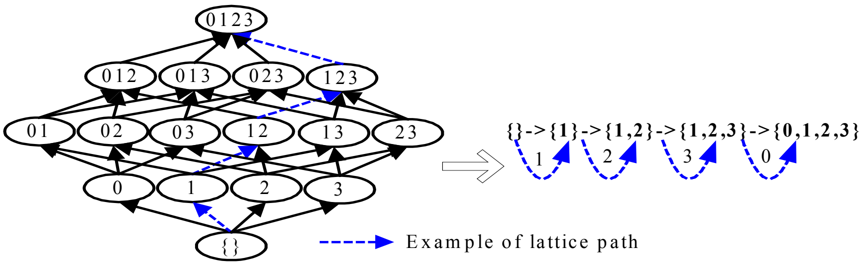


**Figure 3.** The diagram of itemsets use a lattice representation

### 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. 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.

Tree form：

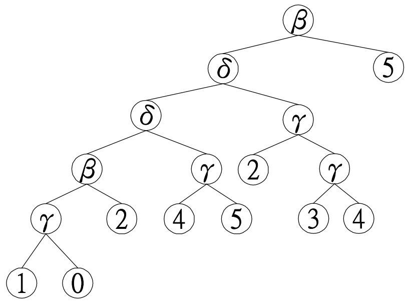


**Figure 4.** Example of a tree form for one virtual lattice

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.

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.

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 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}

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.

Reading Form：

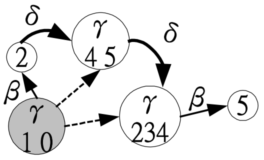


**Figure 6.** Example of a reading form for one virtual lattice

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

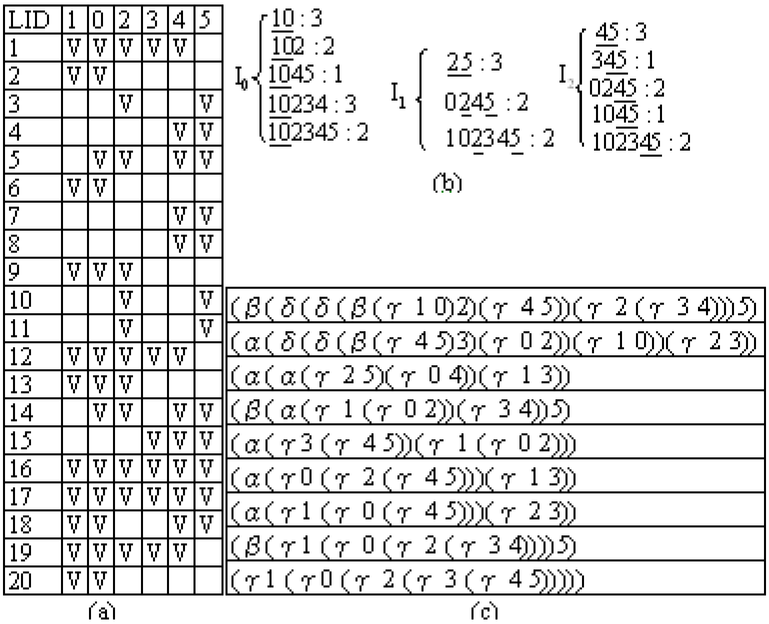
## 4. A comparison of GP plus BCS and the other methods

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*(2n), 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 itemsets.

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.



| LID | 1 | 0 | 2 | 3 | 4 | 5 |
|---|---|---|---|---|---|---|
| 1 | V | V | V | V | V |  |
| 2 | V | V |  |  |  |  |
| 3 |  |  | V |  |  | V |
| 4 |  |  |  |  | V | V |
| 5 |  | V | V |  | V | V |
| 6 | V | V |  |  |  |  |
| 7 |  |  |  |  | V | V |
| 8 |  |  |  |  | V | V |
| 9 | V | V | V |  |  |  |
| 10 |  |  | V |  |  | V |
| 11 |  |  | V |  |  | V |
| 12 | V | V | V | V | V |  |
| 13 | V | V | V |  |  |  |
| 14 |  | V | V |  | V | V |
| 15 |  |  |  | V | V | V |
| 16 | V | V | V | V | V | V |
| 17 | V | V | V | V | V | V |
| 18 | V | V |  |  | V | V |
| 19 | V | V | V | V | V |  |
| 20 | V | V |  |  |  |  |

(a)

$$I_0\begin{cases}\underline{10}:3\\ \underline{10}2:2\\ \underline{10}45:1\\ \underline{10}234:3\\ \underline{10}2345:2\end{cases}\quad I_1\begin{cases}\underline{25}:3\\ 0\underline{2}4\underline{5}:2\\ 10\underline{2}34\underline{5}:2\end{cases}\quad I_2\begin{cases}\underline{45}:3\\ 3\underline{45}:1\\ 02\underline{45}:2\\ 10\underline{45}:1\\ 1023\underline{45}:2\end{cases}$$

(b)

| |
|---|
| (β(δ(δ(β(γ 1 0)2)(γ 4 5))(γ 2 (γ 3 4)))5) |
| (α(δ(δ(β(γ 4 5)3)(γ 0 2))(γ 1 0))(γ 2 3)) |
| (α(α(γ 2 5)(γ 0 4))(γ 1 3)) |
| (β(α(γ 1 (γ 0 2))(γ 3 4))5) |
| (α(γ3 (γ 4 5))(γ 1 (γ 0 2))) |
| (α(γ0 (γ 2 (γ 4 5)))(γ 1 3)) |
| (α(γ1 (γ 0 (γ 4 5)))(γ 2 3)) |
| (β(γ1 (γ 0 (γ 2 (γ 3 4))))5) |
| (γ1 (γ0 (γ 2 (γ 3 (γ 4 5))))) |

(c)

**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.

## 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. 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 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

*Deregulation within the financial service industries and the widespread acceptance of new technologies is increasing competition in the finance marketplace. Central to the business strategy of every financial service company is the ability to retain existing customer and reach new prospective customers. Data mining is adopted to play an important role in these efforts. 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

Deregulation within the financial service industries and the widespread acceptance of new technologies is increasing competition in the finance marketplace. Central to the business strategy of every financial service company is the ability to retain existing customer and reach new prospective customers. Data mining is adopted to play an important role in these efforts. Data mining is an *iterative* process that combines business knowledge, mathematical methods and tools and large amounts of accurate and relevant information to enable the discovery of non-intuitive insights hidden in the organization's corporate data. This information can refine existing processes, uncover trends and help formulate policies regarding the company's relation to its customers and employees. In the financial area, data mining has been applied successfully in determining:

- Who are your future attriters?
- Who are likely to be your profitable customers?
- What is your profitable customers' economic behavior?
- What products are different segments likely to buy?
- What value propositions service different groups?
- What attributes characterize your different segments and how does each play in the person's profile?

In this paper, our focus is on applying data mining techniques to help retailing banks for the attrition analysis. In data mining based direct marketing campaign, it is well understood that targeting every customer is unprofitable and ineffective. With limited marketing budget and staff, data mining models are used to rank the customers and only certain percentage of customers are contacted via mail, phone etc. 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

## 2.1 Brief explanation of the 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. There are customer management program costs and acquisition costs for each account: mailing costs $1/customer and telemarketing costs $5/customer. Cost of incentives (i.e. lowering rates to retain customer) needs to be considered and is dependent upon the offer. 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. The result of this initiative was the launching of the project described in this paper.

## 2.2 Problem Definition

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.
- Cross selling: Identify customers who are likely to purchase alternative products offered to existing loan customers such as life insurance and the like. The increase in relationships is believed to serve as a deterrent to attrition.
- High risk: Customers who are likely to become high risk.
- Pirating: Identify customers likely to transfer their relationship to competing products and away from our client.

These patterns are not unidimensional: a customer can display a subset of these behaviors over the life of the loan. At the same time, he/she can be influenced to change the behavior through the effective use of policies and incentives. Given this, the customer can be thought of operating within a state diagram such as the one depicted in Figure 1:

As the figure shows, a customer, through his actions, can migrate between activity and attrition where each state is defined in terms of a grouping of attributes. Based on this diagram, we decided to concentrate on two attrition problems, namely:

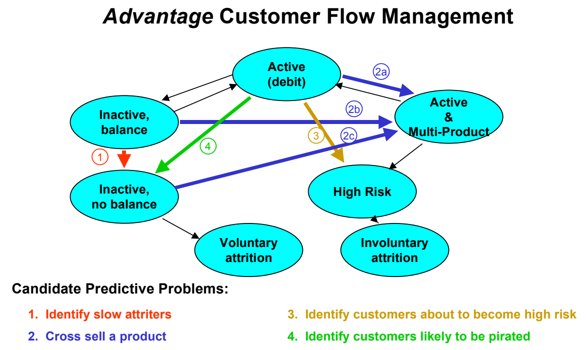

*Advantage* Customer Flow Management
Active (debit)
Inactive, balance
Active & Multi-Product
Inactive, no balance
High Risk
Voluntary attrition
Involuntary attrition
1
2a
2b
2c
3
4
Candidate Predictive Problems:
1. Identify slow attriters
2. Cross sell a product
3. Identify customers about to become high risk
4. Identify customers likely to be pirated

**Figure 1 : Customer's Attrition State Diagram**

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:

**Problem Class #1: Retention of Existing Customers**

The problem requires the stratification of customer segments by leveraging current segmentation model in order to:

- Develop models that predict the customers who are likely to attrite within 30 to 60 days on an ongoing basis.
- Identify the characteristics of the most profitable/desirable customer segments in order to develop policies to ensure their continued support, to grow the group, and to acquire more customers with similar characteristics.

**Problem Class #2: Customer Activation Policies**

Identify customer groups whose characteristics lend them to migrating from unprofitable/dormant to profitable. Once identified, the characteristics can enable the development of risk, maintenance and opportunity policies tailored to a successful migration.

## 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.

## 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

**Table 1: Description of Identified (Potential) Data sources for related data**

| Data Source | Description | Relevance to Retention Modeling |
|---|---|---|
| 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 | Primary source of data for retention modeling problems. |
| Third Party Data | A set of account related demographic and credit bureau information. The data is available from an external provider. | Linked to DDS Warehouse data to provide additional predictive data |
| Segmentation files | Set of account related segmentation values based on our client's segmentation scheme which combines Risk, Profitability and External potential | Combine with DDS data extract to overlay with results of models. |

### 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 2: 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 |

Given this, the next task consisted of generating the additional fields on the "clean" formatted files and adding them to the resulting file. The program was careful not to replicate data whose value is not likely to change over time or, if it changes, is not likely to influence the result. Examples of this are: Account Number, Mother's Maiden name, Address, and the like.

### 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*)

The formula for definition is:

NON_CRD_ST_CD = C (Closed) &&
NON_CRD_STATUS_CHANGE_DATE
Between *beginning_time_period* and
*ending_time_period* &&
NON_CRD_CLS_REA_CD (reason code) in
[0 1 23 25 26 28 29 30 35 36 40 41 42 80 81 82 83 84 97 98 31 32 33 34]

The reason codes for a voluntary attriter (customer requested) are: "0 1 23 25 26 28 29 30 35 36 40 41 42 ", the reason codes for a voluntary attriter (customer requested) related to pricing issues are: "31 32 33 34". According to this definition, the average attrition rate for the section of the data received is 2.2% of all customers.

### 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 analysi**s** 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

In attrition analysis, our goal is to use history information to build an effective data mining model and then use the data mining model to predict the most likely attriters and then take proactive action to prevent the customer attrition. 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. A lift reflects the redistribution of responders in the testing set after the testing examples are ranked. After the learning algorithm ranks all testing examples from most likely responders to least likely responders, we divide the ranked list into some deciles (the top 10 % is finer partitioned in our test experiments: we measure the lift in each percentage), and see how the original responders distributed in these deciles.

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.

## Results

| Pct | cases | Hits boosted BN | % hits | lift | Hits no model |
|---|---|---|---|---|---|
| 1 | 70 | 3 | 4.3% | 1.9 | 1.5 |
| 2 | 141 | 11 | 7.8% | 3.5 | 3.1 |
| 3 | 212 | 15 | 7.1% | 3.2 | 4.7 |
| 4 | 283 | 24 | 8.5% | 3.9 | 6.2 |
| 5 | 354 | 33 | 9.3% | 4.2 | 7.8 |
| 6 | 425 | 41 | 9.6% | 4.4 | 9.3 |
| 7 | 496 | 47 | 9.5% | 4.3 | 10.9 |
| 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 |
| 30 | 2127 | 100 | 4.7% | 2.1 | 46.8 |
| 35 | 2481 | 106 | 4.3% | 1.9 | 54.6 |
| 40 | 2836 | 115 | 4.1% | 1.8 | 62.4 |
| 45 | 3190 | 121 | 3.8% | 1.7 | 70.2 |
| 50 | 3545 | 134 | 3.8% | 1.7 | 78.0 |
| 55 | 3900 | 138 | 3.5% | 1.6 | 85.8 |
| 60 | 4254 | 144 | 3.4% | 1.5 | 93.6 |
| 65 | 4609 | 145 | 3.1% | 1.4 | 101.4 |
| 70 | 4963 | 147 | 3.0% | 1.3 | 109.2 |
| 75 | 5318 | 150 | 2.8% | 1.3 | 117.0 |
| 80 | 5672 | 152 | 2.7% | 1.2 | 124.8 |
| 85 | 6027 | 154 | 2.6% | 1.2 | 132.6 |
| 90 | 6381 | 155 | 2.4% | 1.1 | 140.4 |
| 95 | 6736 | 155 | 2.3% | 1.0 | 148.2 |
| 100 | 7091 | 156 | 2.2% | 1.0 | 156.0 |

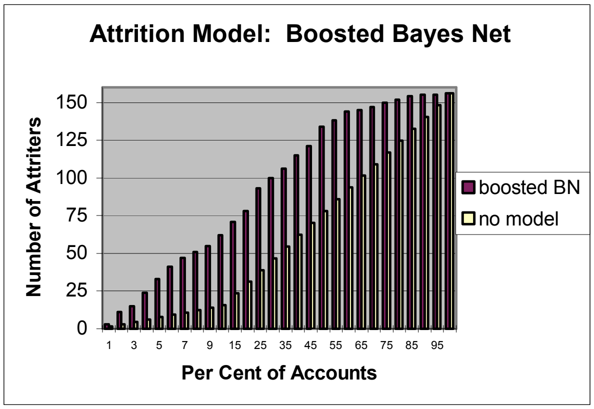


Figure 2: Boosted Naïve Bayes Model Lift Chart

### Variables of Interest

There are 14 most significant positive predictors of the target class picked up by BNB. The top 4 attributes are listed as below in order. Each predictor is a certain value (or numerical range) of a certain attribute. A value of "z" means zero in the original dataset. "counts" is the number of targets versus non-targets with this value of the attribute. "zscore" is the measure of statistical significance.

- Attribute 84 T0_CURRENT_BALANCE {Current Balance carried in hundreds of cents} between -1840.52 and 1277.62: counts 209/86, odds 2.43418, zscore 7.17529

- Attribute 119 T1_CRD_ACCOUNT_FORMAT {Record Format of the Account. Values are: 1 = Never-Active Account, 2 = Account which has been active but is not currently active, 3 = Currently Active Account, 4 = Delinquent Account} between 1.9 and 2.2 : counts 281/154, odds 1.82764, zscore 6.10613
- Attribute 56 T0_NON_CF_LS_MIN_PY_DUE value z {This figure corresponds to the minimum payment due on the last statement. it is used in conjunction with accrued arrears and the number of cycles delinquent to permit automatic delinquency adjustment.} counts 353/214, odds 1.65221, zscore 5.8568
- Attribute 40 T0_NON_CF_LS_OS_BAL {The actual ending balance as it appeared on the cardholder's last statement. this field is not affected by adjustments.} between -1840.52 and 1277.62 : counts 189/98, odds 1.93171, zscore 5.38532

### 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.

## Result

| 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 |

Some of the rules are:

Rule 8: (Lift=5.347, 1-Cover=0.029)
T0_CF_HD_ACT_MNTHS <= 2
T3_CRD_SGMNT_CD = A1
-> class 1 [0.889]

Rule 12: (Lift=4.102, 1-Cover=0.162)
T0_CF_CURRENT_BALANCE <= 407.06
T2_CF_DATE_LAST_STATEMENT <= 1998.055
T3_CRD_SGMNT_CD = A2
-> class 1 [0.859]

Rule 14: (Lift=3.927, 1-Cover=0.318)
T2_CRD_ANN_CHRG_DT <= 1998.164
T0_CF_YTD_NET_NO_PURCHASE <= 0
T0_CF_CURRENT_BALANCE <= 407.06
T3_CRD_SGMNT_CD = A1
-> class 1 [0.812]

Rule 9: (Lift=3.868, 1-Cover=0.385)
T0_CF_CURRENT_BALANCE <= 407.06
T3_CRD_CR_BUR_SCR > 606
T3_CRD_SGMNT_CD = A3
T3_CRD_BKRPCY_REA_CD_3 > 9260
-> class 1 [0.889]

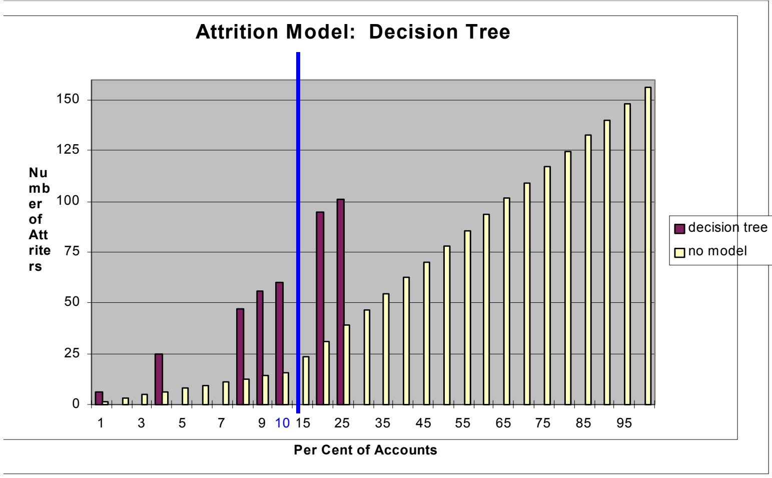


**Figure 3: Decision Tree Model Lift Chart**

### 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) that:

- selects appropriate input transfer functions for fields
- selects subsets of the variables to model the data, and
- builds "constructive" neural network models.

## Results

| PCT | Cases | Hits<br>Neural Net | % hits | lift | Hits<br>no model |
|---|---|---|---|---|---|
| 1 | 70 | 9 | 12.9% | 5.8 | 1.5 |
| 2 | 141 | 16 | 11.3% | 5.2 | 3.1 |
| 3 | 212 | 23 | 10.8% | 4.9 | 4.7 |
| 4 | 283 | 41 | 14.5% | 6.6 | 6.2 |
| 5 | 354 | 41 | 11.6% | 5.3 | 7.8 |
| 6 | 425 | 42 | 9.9% | 4.5 | 9.3 |
| 7 | 496 | 44 | 8.9% | 4.0 | 10.9 |
| 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 |
| 30 | 2127 | 116 | 5.5% | 2.5 | 46.8 |
| 35 | 2481 | 120 | 4.8% | 2.2 | 54.6 |
| 40 | 2836 | 125 | 4.4% | 2.0 | 62.4 |
| 45 | 3190 | 131 | 4.1% | 1.9 | 70.2 |
| 50 | 3545 | 134 | 3.8% | 1.7 | 78.0 |
| 55 | 3900 | 138 | 3.5% | 1.6 | 85.8 |
| 60 | 4254 | 140 | 3.3% | 1.5 | 93.6 |
| 65 | 4609 | 144 | 3.1% | 1.4 | 101.4 |
| 70 | 4963 | 144 | 2.9% | 1.3 | 109.2 |
| 75 | 5318 | 147 | 2.8% | 1.3 | 117.0 |
| 80 | 5672 | 150 | 2.6% | 1.2 | 124.8 |
| 85 | 6027 | 154 | 2.6% | 1.2 | 132.6 |
| 90 | 6381 | 156 | 2.4% | 1.1 | 140.4 |
| 95 | 6736 | 156 | 2.3% | 1.1 | 148.2 |
| 100 | 7091 | 156 | 2.2% | 1.0 | 156.0 |

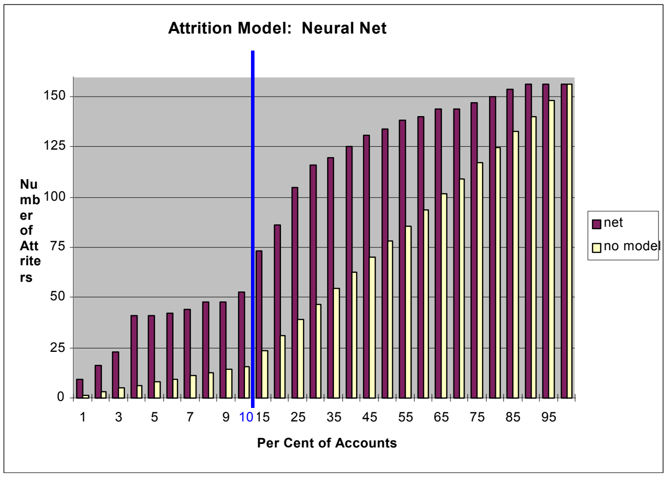


**Figure 4: Neural Net Model Lift Chart**

### 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.

## Results

| PCT | Cases | Hits<br>SelectiveBN | % hits | lift | Hits<br>no model |
|---|---|---|---|---|---|
| 1 | 70 | 5 | 7.1% | 3.2 | 1.5 |
| 2 | 141 | 13 | 9.2% | 4.2 | 3.1 |
| 3 | 212 | 24 | 11.3% | 5.1 | 4.7 |
| 4 | 283 | 27 | 9.5% | 4.3 | 6.2 |
| 5 | 354 | 34 | 9.6% | 4.4 | 7.8 |
| 6 | 425 | 44 | 10.4% | 4.7 | 9.3 |
| 7 | 496 | 49 | 9.9% | 4.5 | 10.9 |
| 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 |
| 30 | 2127 | 112 | 5.3% | 2.4 | 46.8 |
| 35 | 2481 | 114 | 4.6% | 2.1 | 54.6 |
| 40 | 2836 | 118 | 4.2% | 1.9 | 62.4 |
| 45 | 3190 | 124 | 3.9% | 1.8 | 70.2 |
| 50 | 3545 | 125 | 3.5% | 1.6 | 78.0 |
| 55 | 3900 | 129 | 3.3% | 1.5 | 85.8 |
| 60 | 4254 | 136 | 3.2% | 1.5 | 93.6 |
| 65 | 4609 | 140 | 3.0% | 1.4 | 101.4 |
| 70 | 4963 | 145 | 2.9% | 1.3 | 109.2 |
| 75 | 5318 | 150 | 2.8% | 1.3 | 117.0 |
| 80 | 5672 | 153 | 2.7% | 1.2 | 124.8 |
| 85 | 6027 | 154 | 2.6% | 1.2 | 132.6 |
| 90 | 6381 | 156 | 2.4% | 1.1 | 140.4 |
| 95 | 6736 | 156 | 2.3% | 1.1 | 148.2 |
| 100 | 7091 | 156 | 2.2% | 1.0 | 156.0 |

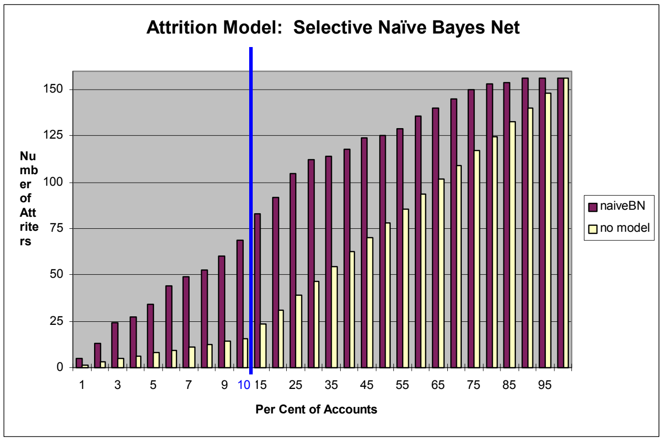


**Figure 5: Selective Naïve Bayesian Network Model Lift Chart**

### 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.

## Results

| Pct | cases | Hits Ensemble of classifier | % hits | lift | Hits no model |
|---|---|---|---|---|---|
| 1 | 70 | 4 | 5.7% | 2.6 | 1.5 |
| 2 | 141 | 12 | 8.8% | 3.8 | 3.1 |
| 3 | 212 | 16 | 7.8% | 3.5 | 4.7 |
| 4 | 283 | 25 | 8.9% | 4.0 | 6.2 |
| 5 | 354 | 36 | 10.3% | 4.6 | 7.8 |
| 6 | 425 | 42 | 9.9% | 4.5 | 9.3 |
| 7 | 496 | 48 | 9.7% | 4.4 | 10.9 |
| 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 |
| 30 | 2127 | 121 | 5.7% | 2.6 | 46.8 |
| 35 | 2481 | 131 | 5.3% | 2.4 | 54.6 |
| 40 | 2836 | 144 | 5.1% | 2.3 | 62.4 |
| 45 | 3190 | 153 | 4.8% | 2.1 | 70.2 |
| 50 | 3545 | 154 | 4.4% | 2.0 | 78.0 |
| 55 | 3900 | 155 | 4.0% | 1.8 | 85.8 |
| 60 | 4254 | 156 | 3.6% | 1.7 | 93.6 |
| 65 | 4609 | 156 | 3.3% | 1.5 | 101.4 |
| 70 | 4963 | 156 | 3.1% | 1.4 | 109.2 |
| 75 | 5318 | 156 | 2.9% | 1.3 | 117.0 |
| 80 | 5672 | 156 | 2.7% | 1.2 | 124.8 |
| 85 | 6027 | 156 | 2.5 % | 1.1 | 132.6 |
| 90 | 6381 | 156 | 2.4% | 1.1 | 140.4 |
| 95 | 6736 | 156 | 2.3% | 1.0 | 148.2 |
| 100 | 7091 | 156 | 2.2% | 1.0 | 156.0 |

## 4. Data Mining Findings

The initial studies unveiled a number of relationships between variables as well as threshold values that justify further discussion and analysis. Following is a summary of the more salient points and their possible meaning:

| Var | Results | Method | Implication |
|---|---|---|---|
| Most recent Current Balance *(CURRENT_BALANCE)* | The most recent current balance showed a strong predictive value when the amount fell below approximately $1000.00<br>A small but significant segment was of those with negative balances, i.e., of customers who overpay. | DTtree<br>NNet<br>BNet | Review of accounts whose previous balance falls below the threshold may be candidates for proactive action.<br>Such candidates can also be "Possibilities" subjects during inbound calls.<br>It can also be an indicator of "Balance Attriters" in the case of negative balances. |
| Current Balance with constant values *(CUR-* | The accounts with values of $12.00 and $15.00 dlls. Showed prominently among the results for attrition. | Stat | If these are interest charges or related to annual charges, it could hint at policies for retention/exiting/ |

|  |  |  |  |
|---|---|---|---|
| *RENT_BALANCE =12 NON_CF_CURRENT_BALANCE <15)* |  |  | win back of customers. |
| Segment *(CRD_SGMNT_CD)* | The association of a group to a specific segment (as defined in the DDS Data Warehouse) was a significant value for segments A1-A4. | DTree | With a larger sample group, we intends to allocate groups to segments (as defined by the Marketing group) in order to run more focused models |
| Annual charge date *(CRD_ANN_CHRG_DT)* | The billing period within the first trimester of the year is predictive of impending attrition for customers with reduced balance *(CURRENT_BALANCE <407)* | DTree<br>NNet<br>BNet | The results point an attrition pattern for lagging users of "zero balance" users who take the charge as a disincentive to maintain the product. |
| Number of payments<br>*(NO_PY)* | Accounts with a number of payments made over the same billing period or payments made to cover low balances over a continuous period *(YTD_NET_PURCHASE_AMT <=62.0)* can be predictive of attrition | DTree | A request for Payoff could indicate, for some accounts, a likelihood of closing (due to a recent annual charge or competitor's bid) which could be averted |
| Incentive Interest<br>*(INCENT_PRI_ANN_MRCH_R)* | Incentive pricing appeared to be somewhat predictive at the value of 4.9% | DTree | This result may warrant a more in-depth study of segment-based review of "rate chaser" population |

The table above shows that several specific values (or ranges of numerical values) of several attributes are useful predictors of retention and/or attrition. These explanations increase our confidence that these values of these attributes will continue to be predictors in the future.

**Field Test:** 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

# Sets of Interesting Association Rules and 4ft-Miner

Jan Rauch, Milan Šimùnek
*Faculty of Informatics and Statistics,University of Economics Prague, Czech Republic*
*rauch@vse.cz,* simunek@vse.cz

## Abstract

*Association rules* **j » y** *are introduced. The association rule* **j » y** *means that Boolean attributes* **j** *and* **y** *are associated in the way given by the symbol ». This symbol is called 4ft quantifier. A condition concerning a four-fold contingency table of* **j** *and* **y** *is associated to each 4ft quantifier. Various types of implication or equivalency of* **j** *and* **y** *can be expressed. It is also possible to express relations corresponding to statistical hypotheses tests. Conditional association rules* **j » y /c** *are also introduced. Conditional association rule* **j » y / c** *means that when the condition* **c** *is satisfied then the Boolean attributes* **j** *and* **y** *are associated in the way given by 4ft quantifier ». The procedure 4ft-Miner mining for association rules* **j » y** *and* **j » y /c** *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→Y where X and Y are sets of items [1]. The intuitive meaning of X→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 $\varphi$ and $\psi$ are derived Boolean attributes. The intuitive meaning of association rule $\varphi \approx \psi$ is that Boolean attributes $\varphi$ and $\psi$ are associated in the way corresponding to the condition given by the symbol ≈. Symbol ≈ is called 4ft-quantifier. It denotes a condition concerning a four-fold contingency table of $\varphi$ and $\psi$. Various types of implication or equivalency of X and Y can be expressed. It is also possible to express relations corresponding to statistical hypotheses tests (e.g. $\chi^2$-test or Fisher’s test).

We use the following ideas formulated in connection with GUHA method [2]:

- The Boolean attributes $\varphi$ and $\psi$ can be derived from columns of analysed data matrix There are clear syntactical rules describing how $\varphi$ and $\psi$ can be derived. These rules ensure that association rules $\varphi \approx \psi$ are *ultimately understandable pattern*.
- 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 set of potentially interesting association rules.
- Output of the mining procedure consists of all prime association rules. The association rule is prime if (i) it is *valid in the analysed data matrix* and (ii) it does not follow immediately from other more simple output association rules.
- There is software enabling us to find really *novel* association rules among the output *valid* association rules.

We describe main features of the procedure 4ft-Miner mining for the association rules of the form $\varphi \approx \psi$. 4ft-Miner is the GUHA procedure in the sense of [2]. The procedure 4ft-Miner mines for more general association rules than the GUHA procedure PC-GUHA [4] 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 LiSp-Miner for support of research and teaching of knowledge discoverz in databases. For more details see http://lispminer.vse.cz/.

Association rules and conditional association rules are introduced in section 2. Possibilities of a definition of the set of *interesting* (i.e. *potentially useful*) association rules are outlined in section 3. An example of application of 4ft-Miner is given in the section 4.

The procedure 4ft-Miner does not use the A-priori algorithm [1]. The main principles applied in the procedure 4ft-Miner are briefly introduced in the section 5. These principles were used already in the early implementations of the GUHA method [6]. Logical properties of association rules are used to optimise the 4ft-Miner algorithm. Some of them are mentioned in the section 6.

The last goal of this paper is to outline a possibility to define multi-relational association rules see section 7. Multi-relational association rules can be also understood as formulae of special many sorted calculi see section 7.

## 2 Association rules

The association rule is the expression $\varphi \approx \psi$. The intuitive meaning of $\varphi \approx \psi$ is that Boolean attributes $\varphi$ and $\psi$ are associated in the way corresponding to the condition given by the 4ft-quantifier $\approx$. Association rule concerns the analysed data matrix. Boolean attributes $\varphi$ and $\psi$ are derived from columns of analysed data matrix. An example of the analysed data matrix is the data matrix Loans in Fig. 1.

| *Client* | *Clients* | | | | *Loans* | | | |
|---|---|---|---|---|---|---|---|---|
| | Age | Sex | Salary | District | Amount | Payment | Years | Quality |
| *1* | 45 | M | high | Prague | 48 000 | 1 000 | 4 | good |
| *2* | 32 | F | low | Plzen | 120 000 | 10 000 | 1 | bad |
| ... | ... | ... | ... | ... | ... | ... | ... | ... |
| *6180* | 54 | M | avg | Kolin | 10 000 | 1 000 | 1 | bad |
| *6181* | 24 | F | high | Brod | 36 000 | 2 000 | 2 | good |

Figure 1. – Data matrix Loans

Each row of the matrix describes a loan of a client of a bank. There are 6 181 loans. Attributes Age, Sex, Salary and District correspond to clients, attributes Amount, Payment Duration and Quality correspond to loans of clients. The first row describes a loan that get a 45 years old man. This man has a high salary and lives in the district Prague. He borrowed 48 000 Czech crowns. He repays 1 000 Czech crowns and he will pay 4 years. The quality of his loan is good.

Association rule $\varphi \approx \psi$ is verified on the basis of four-fold table of $\varphi$ and $\psi$ in the analysed data matrix *M* see Tab. 1.

| *M* | $\psi$ | $\neg\psi$ | |
|---|---|---|---|
| $\varphi$ | a | b | r |
| $\neg\varphi$ | c | d | s |
| | k | l | n |

Table 1. – Four-fold table 4ft($\varphi$, $\psi$, *M*) of $\varphi$, $\psi$ in *M*

Here *a* is the number of objects satisfying both $\varphi$ and $\psi$, *b* is the number of objects satisfying $\varphi$ and not satisfying $\psi$, *c* is the number of objects not satisfying $\varphi$ and satisfying $\psi$, and *d* is the number of objects satisfying neither $\varphi$ nor $\psi$. Further $r = a + b$ is the number of objects satisfying $\varphi$, similarly for *s*, *k*, and *l*, *n* is the number of all objects. The quadruple < a,b,c,d > is called *four-fold table of attributes j and y in data matrix M*, symbolically 4ft($\varphi$, $\psi$, *M*).

Association rule $\varphi \approx \psi$ can be true or false in the given data matrix *M*. If the condition associated to 4ft quantifier $\approx$ *is satisfied* for four-fold table 4ft($\varphi$, $\psi$, *M*) then association rule $\varphi \approx \psi$ *is true in M*. If this condition is *not satisfied* for four-fold table 4ft($\varphi$, $\psi$, *M*) then association rule $\varphi \approx \psi$ *is false in M*. Various 4ft quantifiers are defined in [2], [3] and [8], 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 $\varphi$ satisfy also $\psi$"* or "*$\varphi$ implies $\psi$ on the level 100p per cent*".

- **Lower critical implication** $\Rightarrow^{!}_{p;\propto;Base}$ with parameters $0 < p \leq 1$, $Base > 0$ and $0 < \boldsymbol{\mu} \leq 0.5$. The condition $\sum_{i=a}^{a+b} \frac{(a+b)!}{i!(a+b-i)!} * p^{i} * (1-p)^{a+b-i} \leq \boldsymbol{a} \wedge a \geq Base$ is associated to 4ft quantifier $\Rightarrow^{!}_{p;\propto;Base}$. The association rule $\varphi \Rightarrow^{!}_{p;\propto;Base} \psi$ corresponds to a test (on the level $\alpha$) of a null hypothesis $H_0$: $P(\varphi|\psi\ ) \leq p$ against the alternative one $H_1$: $P(\varphi|\psi) > p$. If association rule $\varphi \Rightarrow^{!}_{p;\propto;Base} \psi$ is true in data matrix *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 $\Leftrightarrow_{p;Base}$. The association rule $\varphi \Leftrightarrow_{p;Base} \psi$ can be interpreted as *"100p per cent of objects satisfying* $\varphi$ *or* $\psi$ *satisfy both* $\varphi$ *and* $\psi$" 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 $\equiv_{p;Base}$. The association rule $\varphi \equiv_{p;Base} \psi$ can be interpreted as *"100p per cent of objects have the same value for* $\varphi$ *and* $\psi$".

- **Fisher´s quantifier** $\sim_{\boldsymbol{a},Base}$ with parameters $0 < \boldsymbol{\mu} \leq 0.5$ and $Base > 0$. The condition $ad > bc \wedge \sum_{i=a}^{\min(a+b,a+c)} \frac{r!s!k!l!}{n!i!(r-i)!(k-i)!(n-r-k-i)!} \leq \boldsymbol{a} \wedge a \geq Base$ is associated to 4ft quantifier $\sim_{\boldsymbol{a},Base}$. The association rule $\varphi \sim_{\boldsymbol{a},Base} \psi$ corresponds to a test (on the level $\alpha$) of the null hypothesis of independence of $\varphi$ and $\psi$ 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 support and *S* is confidence.

- A further relation of two Boolean attributes is defined in [13]. It can be understood as a generalised quantifier with $\approx^{E}_{\boldsymbol{d}}$ with the condition $\frac{b}{a+b} < \boldsymbol{d} \wedge \frac{c}{c+d} < \boldsymbol{d}$ associated to it.

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 $\varphi$, $\psi$ and $\chi$ are derived Boolean attributes. The intuitive meaning of association rule $\varphi \approx \psi / \chi$ is that Boolean attributes $\varphi$ and $\psi$ are associated in the way given by the 4ft-quantifier $\approx$ when the condition $\chi$ 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 $M / \chi$. The data matrix $M /\chi$ is data matrix consisting from rows of data matrix *M* satisfying Boolean attribute $\chi$. We suppose that at least one row of *M* satisfy $\chi$.

## 3. Set of Interesting Association Rules

The procedure 4ft-Miner 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

$$\text{District}(\textit{Prague}, \textit{Plzen}) \wedge \text{Salary}(\textit{low}) \Rightarrow_{0.7;30} \text{Quality}(\textit{bad}).$$

Here District(*Prague*, *Plzen*), Salary(*low*) and Quality(*bad*) are *literals*. Literal District(*Prague, Plzen*) is true in a row of data matrix Loans (see Fig. 1) if there are values *Prague* or *Plzen* in this row.

The set of *interesting* association rules to be automatically generated and tested on the given 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,
  - 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(*subset_of_categories*);
- negative literal is the expression ¬A(*subset_of_categories*).

Here A is the attribute and *subset_of_categories* is an own subset of the set of categories of attribute A. The category of the attribute A is one of its possible values. The set of categories of attribute Salary is the set {low, awg, high}. The set *subset_of_categories* is a *coefficient* of literal A(*subset_of_categories*). There are various types of coefficient.

The literal A(*subset_of_categories*) is the Boolean attribute that is true in the row of analysed data matrix if the value of the attribute A in this row belongs to the set *subset_of_categories*. Otherwise the literal A(*subset_of_categories*) is false in the this row. A negative literal ¬A(*subset_of_categories*) is a Boolean negation of positive literal A(*subset_of_categories*).

District(*Prague*), District(*Prague, Plzen*) and ¬District(*Prague*) are examples of literals derived from the attribute District. Further, *Prague* is the coefficient of literals District(*Prague*) and ¬District(*Prague*). *Prague, Plzen* is the coefficient of the literal District(*Prague, Plzen*). Examples of values of these literals are in Table 2 (true= 1 and false= 0):

| Client | District | District(*Prague*) | District(*Prague, Plzen*) | ¬District(*Prague*) |
|---|---|---|---|---|
| 1 | *Prague* | 1 | 1 | 0 |
| 2 | *Plzen* | 0 | 1 | 1 |
| 3 | *Brno* | 0 | 0 | 1 |
| ... | ... | ... | ... | ... |
| 6180 | *Prague* | 1 | 1 | 0 |
| 6181 | *Brod* | 0 | 0 | 1 |

Table 2 – Examples of values of literals

The set of all literals to be generated from the particular attribute is given by:

- A type of the coefficient, there are five types of coefficients: subsets, intervals, left cuts, right cuts, and cuts.
- Minimal and maximal number of categories in the 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(1,2,3) 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 question is what does it mean that the association rule Ant ≈ Suc immediately follows from more simple association rule $\mathrm{Ant}_1 \approx \mathrm{Suc}_1$.

The definition of prime association rule depends on properties 4ft-quantifier. If we use the 4ft-quantifier $\Leftrightarrow_{p;Base}$ of double founded implication then the association rule Ant $\Leftrightarrow_{p;Base}$ Suc is prime if and only if it is true. There is not any reasonable more simple association rule $\text{Ant}_1 \Leftrightarrow_{p;Base} \text{Suc}_1$ such that Ant $\Leftrightarrow_{p;Base}$ Suc follow immediately follow from $\text{Ant}_1 \Leftrightarrow_{p;Base} \text{Suc}_1$.

The definition of prime association rule for the 4ft quantifier $\Rightarrow_{p;Base}$ must take into that e.g if association rule Sex(M) $\Rightarrow_{p;Base}$ District(*Prague*) is true then the association rule Sex(M)$\Rightarrow_{p;Base}$District(*Prague,Plzen*) is also true. Thus the association rule Sex(M) $\Rightarrow_{p;Base}$ District(*Prague, Plzen*) immediately follow from more simple association Sex(M) $\Rightarrow_{p;Base}$ District(*Prague*). The precise definition of the prime association rules is out of the range of this paper.

## 4. 4ft-Miner – an example

Our example concerns data matrix Loans see Figure 1. We search for interesting segments of clients and types of loans. Segments of clients are defined by attributes Age, Sex, Salary and District. An example of segment of client is segment of all men in the age of 31 – 35 years with high salary. It is defined by the conjunction Sex(M) ∧ Age(31,…,35) ∧ Salary(*high*). An other example is segment of all clients with low salary living in Prague or in Plzen. It is defined by conjunction Salary(*low*) ∧ District(*Prague, Plzen*).

Types of loans are defined by attributes Amount and Payment. Two steps are used. In the first step new attributes Amount and Payment are defined. The new attribute Amount has categories < 20, <20, 50), <50, 100), <100, 250), <250, 500) > 500. These categories are intervals of thousands of Czech crowns. Thus Amount<20, 50) is true if the value of the "old" attribute Amount is in the interval <20 000, 50 000). The new attribute Payment has categories (0,1>, (1,2>, …, (9,10>. These categories are again intervals of thousands of Czech crowns. These transformations can be done the module DataSource. The module DataSource as well as the procedure 4ft-Miner is involved in the system LISp-Miner see http://lispminer.vse.cz/.

We are interested in all segments SEGMENT of clients such that „to be a member of the SEGMENT" is nearly equivalent to "the loan is bad" when we consider loans of particular types. We accept that „to be a member of the SEGMENT" is nearly equivalent to " the loan is bad " for the loans of type TYPE if the condition $\frac{a}{a+b+c} \geq 0.95 \wedge a \geq 30$ is satisfied. Here a, b and c are frequencies from the four-fold table 4ft(SEGMENT, Quality(*bad*), Loans/TYPE) of Boolean attributes SEGMENT and Quality(*bad*) in data matrix Loans / TYPE see Table 3.

| Loans / TYPE | Quality(*bad*) | ¬ Quality(*bad*) |
|---|---|---|
| SEGMENT | a | b |
| ¬ SEGMENT | c | d |

Table 3 - Four-fold table 4ft(SEGMENT, Quality(*bad*), Loans / TYPE)

It means that a is the number of loans of the type TYPE satisfying both SEGMENT and Quality(*bad*). Similarly b a is the number of loans of the type TYPE satisfying SEGMENT and not satisfying Quality(*bad*). Further c a is the number of loans of the type TYPE not satisfying SEGMENT and satisfying Quality(*bad*).

The condition $\frac{a}{a+b+c} \geq 0.95 \wedge a \geq 30$ means that at least 95 per cent of of loans of the type TYPE satisfying SEGMENT or Quality(*bad*) satisfy both SEGMENT and Quality(*bad*). If this condition is satisfied then the association rule

$$\text{SEGMENT} \Leftrightarrow_{0.95;30} \text{Quality}(bad) / \text{TYPE}$$

is true in data matrix Loans. The symbol $\Leftrightarrow_{0.95;30}$ is 4ft quantifier of double founded implication see section 2.

We are interested in all SEGMENTs and all TYPEs such that „to be a member of the SEGMENT" is nearly equivalent to " the loan is bad " when we consider loans of particular TYPE. We use the procedure 4ft Miner. Thus we have to define the set of all interesting association rules SEGMENT $\Leftrightarrow_{0.95;30}$ Quality(*bad*) / TYPE. Segments of clients are defined as literals or as conjunctions of literals. Further examples of segments follow.

- Age<21,30> - segment of all clients in the age 21-30 years.
- Age <31,40> ∧ Sex(*M*) - segment of all clients - men in the age 21-30 years.
- Sex(*M*) ∧ Salary(*high*) - segment of all clients - men with high salary.
- Sex(*M*) ∧ Salary(*high*) ∧ District(*Prague, Plzen*) - segment of all men with high salary living in Prague or in Plzen.

All segments - literals or conjunctions of literals can be coded in this way (the symbol ? stands for coefficient):

- Age(?), Sex(?), Salary(?), District(?).
- Age(?) ∧ Sex(?), Age(?) ∧ Salary(?),
  Age(?) ∧ District(?), Sex(?) ∧ Salary(?),
  Sex(?) ∧ District(?), Salary(?) ∧ District(?).
- Age(?) ∧ Sex(?) ∧ Salary(?),
  Age(?) ∧ Sex(?) ∧ District(?),
  Sex(?) ∧ Salary(?) ∧ District(?).
- Age(?) ∧ Sex(?) ∧ Salary(?) ∧ District(?).

We define the set of all interesting coefficients for each of literals Age(?), Sex(?), Salary(?) and District(?). Further we show only one of many possibilities:

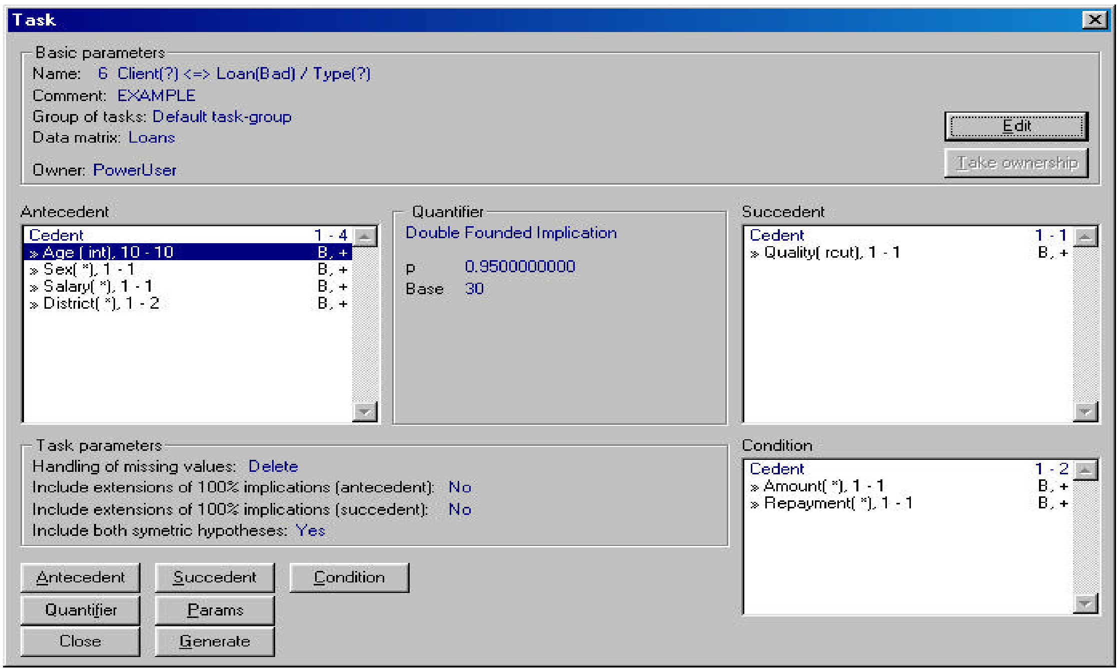
- We use intervals of length 10 (i.e. sliding window of the length 10) for the attribute Age. The attribute Age has categories 21 – 67 thus 38 literals Age(?) are defined this way. All these literals are symbolically denoted by **Age(int), 10 - 10**.
- We use all single categories (subsets consisting of one category) for the attribute Sex, symbolically **Sex(*), 1 – 1**. Two literals are defined this way.
- We use all single categories for the attribute Salary, symbolically **Salary(*), 1 – 1**. Three literals are defined this way (there are categories *low*, *awg* and *high*).
- We use all single categories and all pairs of categories (subsets with 1 - 2 categories) for the attribute District, symbolically **District(*), 1 – 2**. The attribute District has 77 categories thus 3 003 literals District (?) are defined this way.

We define all SEGMENT as antecedents - conjunctions of 1 – 4 literals. The total number of SEGMENTs defined this way is 1 060 477.

Similarly we define the set of all TYPEs as all conjunction of 1 – 2 literals derived from attributes Amount and Payment.

- We use all single categories (subsets consisting of one category) for the attribute Amount, symbolically **Amount(*), 1 – 1**. The attribute Amount has 6 categories (see above) thus 6 literals are defined this way.
- We use all single categories for the attribute Payment, symbolically **Payment(*), 1 – 1**. The attribute Payment has 10 categories ((0,1>, (1,2>, …, (9,10>) thus 10 literals are defined this way.

The total number of TYPEs defined this way is 76. There are more about 80.5 millions of association rules SEGMENT $\Leftrightarrow_{0.95;30}$ Quality(*bad*) / TYPE defined this way. The overview of the input parameters of 4ft-Miner is in Figure 2.



Task

Basic parameters
Name: 6 Client(?) <=> Loan(Bad) / Type(?)
Comment: EXAMPLE
Group of tasks: Default task-group
Data matrix: Loans
Owner: PowerUser

Antecedent

| Cedent | 1 - 4 |
|---|---|
| » Age ( int), 10 - 10 | B, + |
| » Sex( *), 1 - 1 | B, + |
| » Salary( *), 1 - 1 | B, + |
| » District( *), 1 - 2 | B, + |

Quantifier
Double Founded Implication
p 0.9500000000
Base 30

Succedent

| Cedent | 1 - 1 |
|---|---|
| » Quality( rcut), 1 - 1 | B, + |

Task parameters
Handling of missing values: Delete
Include extensions of 100% implications (antecedent): No
Include extensions of 100% implications (succedent): No
Include both symetric hypotheses: Yes

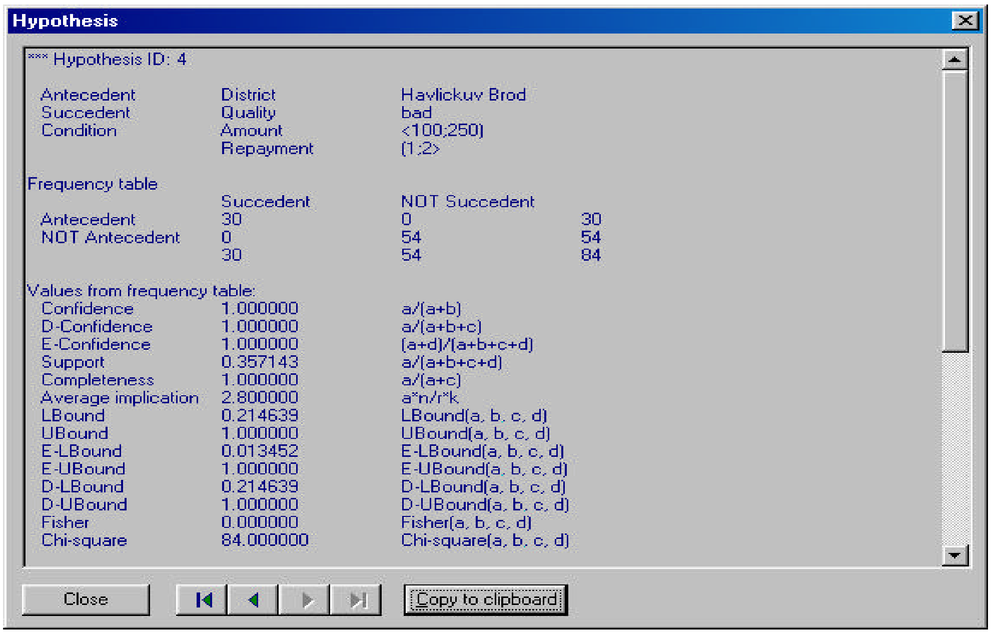
Condition

| Cedent | 1 - 2 |
|---|---|
| » Amount( *), 1 - 1 | B, + |
| » Repayment( *), 1 - 1 | B, + |

Figure 2. – Overview of input parameters of 4ft-Miner

The procedure generates and verifies all these association rules in about 13 minutes (Pentium III with 128 MB). The result consists of 4 association rules true in the data matrix Loans. An example of output association rule is in Figure 3. It is the association rule

$$\text{District}(\mathit{Havlickuv\ \ Brod}) \Leftrightarrow_{1.0;30} \text{Quality}(\mathit{bad}) \;/\; \text{Amount}(100, 250) \wedge \text{Payment } (1,2)\,.$$



```
*** Hypothesis ID: 4

  Antecedent           District          Havlickuv Brod
  Succedent            Quality           bad
  Condition            Amount            <100;250)
                       Repayment         (1;2>

Frequency table
                       Succedent         NOT Succedent
  Antecedent           30                0                    30
  NOT Antecedent       0                 54                   54
                       30                54                   84

Values from frequency table:
  Confidence           1.000000          a/(a+b)
  D-Confidence         1.000000          a/(a+b+c)
  E-Confidence         1.000000          (a+d)/(a+b+c+d)
  Support              0.357143          a/(a+b+c+d)
  Completeness         1.000000          a/(a+c)
  Average implication  2.800000          a*n/r*k
  LBound               0.214639          LBound(a, b, c, d)
  UBound               1.000000          UBound(a, b, c, d)
  E-LBound             0.013452          E-LBound(a, b, c, d)
  E-UBound             1.000000          E-UBound(a, b, c, d)
  D-LBound             0.214639          D-LBound(a, b, c, d)
  D-UBound             1.000000          D-UBound(a, b, c, d)
  Fisher               0.000000          Fisher(a, b, c, d)
  Chi-square           84.000000         Chi-square(a, b, c, d)
```

Figure 3. – Example of the output association rule

## 5. Implementation of the 4ft-Miner

Let us remember that the procedure 4ft-Miner mines for association rules Ant ≈ Suc and for conditional association rules Ant ≈ Suc / Cond. The Boolean attributes Ant, Suc and Cond are automatically generated conjunctions of literals. Ant is called antecedent, Suc is called succedent and Cond is called condition. We will use the notion **cedent** to denote any of Boolean attributes antecedent, succedent or condition.

A huge number of interesting association rules is usually defined by input parameters of 4ft-Miner. E.g. there are about 80.5 millions of the interesting association rules defined by parameters of 4ft-Miner in Fig. 2, see section 4. The fast algorithm generating and verifying the huge number of association rules is crucial for reasonable work with 4ft-Miner.

Two main principles are applied in 4ft-Miner:

- Analysed data matrices are represented by suitable strings of bits; see section 5.1.
- Various optimisation tools are applied in the depth first search for the prime association rules used see section 5.2.

Let us remark that these principles were used already in the early implementations of the GUHA method [6].

### 5.1. Applying strings of bits

We use very simple example to show how the strings of bits are used to represent analysed data matrix and to compute four-fold tables in a very fast way. Let us suppose that we have data matrix $M_A$ with five objects and two attributes A and B see Table 4.

| OBJECT ID | ATTRIBUTE A | ATTRIBUTE B |
|---|---|---|
| 1 | x | p |
| 2 | z | p |
| 3 | y | q |
| 4 | z | p |
| 5 | x | q |

Table 4 - Data matrix $M_A$ with attributes A , B

The attribute A has three possible values *x*, *y* and *z*. Thus the attribute A is represented by three literals (see section 3) *A(x)*, *A(y)* and *A(z)* in the way shown in the Table 5. Similarly the attribute B is represented by two literals *B(p)* and *B(q)*.

| OBJEKT ID | A | B | *A(x)* | *A(y)* | *A(z)* | *B(p)* | *B(q)* |
|---|---|---|---|---|---|---|---|
| 1 | x | p | 1 | 0 | 0 | 1 | 0 |
| 2 | z | p | 0 | 0 | 1 | 1 | 0 |
| 3 | y | q | 0 | 1 | 0 | 0 | 1 |
| 4 | z | p | 0 | 0 | 1 | 1 | 0 |
| 5 | x | q | 1 | 0 | 0 | 0 | 1 |

Table 5 – Representation of the attributes A, B by literals

Literals *A(x)*, *A(y)*, *A(z)*, *B(p)* and *B(q)* are Boolean attributes and thus they can be represented by strings of bits. This representation makes possible to apply very fast bit operations *AND* and *OR* to compute strings of bits representing particular cedents Ant, Suc and Cond in association rules Ant ≈ Suc and Ant ≈ Suc / Cond . It is done in the way given by the following examples:

- String of bits representing the literal *A(x, y)* is computed as disjunction *A(x) OR A(y)* of strings representing literals *A(x)* and *A(y)*.
- String of bits representing the cedent *A(x, y)* ∧ *B(p)* is computed as the conjunction *A(x, y) AND B(p)* of strings representing literals *A(x, y)* and *B(p)*.

The representation of cedents by strings of bits makes possible to compute frequencies of four-fold table. The operation *Count*(α) returns number of values 1 in the string α. We show how the frequencies a,b,c,d from the four-fold table 4ft(Ant, Suc, *M*) of Ant, Suc in the data matrix *M* (see Table 6) are computed. This table is used in the verification of association rule Ant ≈ Suc in the data matrix *M* .

| *M* | Suc | ¬ Suc |
|---|---|---|
| Ant | a | b |
| ¬ Ant | c | d |

Table 5 – Four-fold table 4ft(Ant, Suc, *M*) of Ant, Suc in the data matrix *M*

Let us suppose that the string of bits *Ant* represents the antecedent Ant and that the string of bits *Suc* represents the succedent Suc. Then the particular frequencies are computed in the following way:

- a = *Count(Ant AND Suc)*
- b = *Count(Suc) – a*
- c = *Count(¬ Ant AND Suc)*
- d = n – a – b – c

where n is the total number of rows in the data matrix *M*.

The total size of memory required representing all the attributes by strings of bits if acceptable. There are eight attributes in the data matrix Loans see Figure 1. Particular attributes have various numbers of categories. The total size of memory necessary to represent all categories by corresponding strings of bits is shown in the Table 6. Several data matrices with various numbers of rows are considered.

| Atttribute | # of ctg. | number of rows in data matrix Loans | | | |
|---|---|---|---|---|---|
| | | 6 181 | $10^5$ | $10^6$ | $10^7$ |
| Sex | 2 | 2 KB | 19 KB | 190 KB | 1.9 MB |
| Salary | 3 | 2 KB | 38 KB | 380 KB | 3.8 MB |
| Months | 5 | 4 KB | 63 KB | 630 KB | 6.3 MB |
| Age | 6 | 5 KB | 75 KB | 750 KB | 7.5 MB |
| Amount | 6 | 5 KB | 75 KB | 750 KB | 7.5 MB |
| Payment | 10 | 8 KB | 128 KB | 1.3 MB | 12 MB |
| District | 77 | 60 KB | 963 KB | 9.6 MB | 96 MB |
| Total | 109 | 84 KB | 1.4 MB | 14 MB | 140 MB |

Table 6. – Memory necessary to represent particular attributes and the whole data matrix

## 5.1. Search for Prime Association Rules

The algorithm of generating and testing of association rules Ant ≈ Suc is outlined in Figure 4. The algorithm for conditional association rules Ant ≈ Suc / Cond is similar.

```
ANTECEDENT := First_Relevant_Antecedent
while not End_of_Antecedents
  SUCCEDENT := First_Relevant_Succedent(ANTECEDENT)
  while not End_of_Succedents(ANTECEDENT)
   if True_Association_Rule(ANTECEDENT,SUCCEDENT)
        then if Prime_Association_Rule(ANTECEDENT,SUCCEDENT)
                then Output(ANTECEDENT,SUCCEDENT)
    SUCCEDENT := Next_Succedent(ANTECEDENT,SUCCEDENT)
  end while
 ANTECEDENT := Next_Antecedent(ANTECEDENT)
end while
```

Figure 4. – The algorithm of generating and testing of association rules

The particular cedents (Ant, Suc and Cond) are generated “depth first”. Let us suppose that the set of interesting association rules is given according to the Figure 2., see section 4. It means e.g. that the procedure First_Relevant_Antecedent returns the antecedent Age<21,30>. The procedure Next_Antecedent then generates the sequence of antecedents outlined in the Figure 6.

Age<21,30> ∧ Sex(M)
Age<21,30> ∧ Sex(M) ∧ Salary(low)
Age<21,30> ∧ Sex(M) ∧ Salary(low)
∧ District(Benesov, Brno)
…
Age<21,30> ∧ Sex(M) ∧ Salary(low) ∧ District(Benesov,Zlin)
…
Age<21,30> ∧ Sex(M) ∧ Salary(low)
∧ District(Brno, Chomutov)
…
Age<21,30> ∧ Sex(M) ∧ Salary(low) ∧ District(Brno, Zlin)
… …

Figure 6. – The sequence of antecedents generated by the procedure Next_Antecedent

This way generation of cedents makes possible to use very effectively strings of bits representing literals and cedents. Further various ways of optimisations are used. We show two examples:

- Let us suppose that we use the quantifier $\Leftrightarrow_{p;Base}$ of double founded implication and that the number of rows in the analysed data matrix satisfying Ant is smaller than Base (*Count(Ant)* < *Base*). Then it has no sense to generate and verify any of association rules Ant $\Leftrightarrow_{p;Base}$ Suc because it will be also a = *Count(Ant AND Suc)* < *Base*. Thus the condition $\frac{a}{a+b+c} \geq p \wedge a \geq Base$ corresponding to $\Leftrightarrow_{p;Base}$ cannot be satisfied.
- Let us suppose that we use the quantifier $\Rightarrow_{p;Base}$ of founded implication and that the association rule Sex(M) $\Rightarrow_{p;Base}$ District(*Prague*) is true. Then it has no sense to generate any of association rules Sex(M) $\Rightarrow_{p;Base}$ District(*Prague,* *) because of each of these association rules will be true but it cannot be prime. Here * stands for any value of the attribute District. For example the association rule Sex(M) $\Rightarrow_{p;Base}$ District(*Prague, Plzen*) is true but not prime, see the end of the section 3.

## 6. Logical Properties of Association Rules

The procedure 4ft – Miner deals with association rules defined in section 2. The definition of association rule can be done in 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. [5], [12].

Logical calculi formulae of which correspond to association rules were defined and studied e.g. in [2], [7], [8] and [9]. 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 $\frac{j \approx y}{j\,' \approx y'}$ were achieved. The deduction rule $\frac{j \approx y}{j\,' \approx y'}$ is correct if the following is satisfied for each data matrix $M$: If the association rule φ ≈ ψ is true in $M$ then also the association rule φ’ ≈ ψ’ is true in $M$.

An example of very simple correct deduction rule is deduction rule $\frac{j \Rightarrow_{p,Base} y}{j \Rightarrow_{p,Base} y \vee c}$. It says: If the association rule φ $\Rightarrow_{p;Base}$ ψ is true in $M$ then also the association rule φ $\Rightarrow_{p;Base}$ ψ ∨ χ is true in $M$.

Correct deduction rules of the form $\frac{j \approx y}{j\,' \approx y'}$ are used in procedure 4ft-Miner in two ways:

- If it is known that the association rule φ ≈ ψ is true than it is not necessary to generate and test the association rules φ’ ≈ ψ’ because of they are sure true.
- If it is known that φ ≈ ψ is true than it is not necessary to put into the output association rule φ’ ≈ ψ’ because of it is sure true. This approach but requires only *transparent* deduction rules.

Let us remark that deduction rules not only of the form $\frac{j \approx y}{j' \approx y'}$ are used in the procedure 4ft-Miner.

Various classes of 4ft-quantifiers can be defined e.g. implication 4ft-quantifiers, double implication 4ft-quantifiers and equivalency 4ft-quantifiers.

For all quantifiers used in the procedure 4ft-Miner there is a relatively simple condition equivalent to the fact that the deduction rule $\frac{j \approx y}{j' \approx y'}$ is correct [8]. This condition depends on the class the quantifier ≈ belongs to. The condition concerns a propositional formula Φ(φ, ψ, φ', ψ') derived from Boolean attributes φ, ψ, φ', ψ'. Deduction rule $\frac{j \approx y}{j' \approx y'}$ is correct if and only if the formula Φ(φ, ψ , φ', ψ') is a tautology of propositional calculus. A more detailed description of properties of deduction rules concerning association rules is out of range of this paper.

## 7. Multi-relational association rules

We outline a definition of multi-relational association rules. We use two simple relations (i.e. data matrices): Loans (see Fig. 1) and Transactions (see Fig. 4).

| Trans. | Client | T_Amount | Type | Bank | Account | Date |
|---|---|---|---|---|---|---|
| *1* | *1* | 1 500 | TX | A | A17 | 20001231 |
| *2* | *1* | 300 | TY | B | B21 | 20010102 |
| ... | ... | ... | ... | ... | ... | ... |
| *199* | *1* | 875 | TZ | A | A31 | 20020210 |
| *200* | *1* | 349 | TY | D | D12 | 20010227 |
| *201* | *2* | 1 200 | TU | A | A17 | 20001231 |
| *202* | *2* | 680 | TX | E | E23 | 20010102 |
| ... | ... | ... | ... | ... | ... | ... |
| *299* | *2* | 240 | TW | B | B14 | ... |
| *300* | *2* | 5440 | TY | A | A17 | 20010227 |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| ... | ... | ... | ... | ... | ... | ... |
| *500 001* | *6181* | 2 400 | TZ | F | F54 | 20000324 |
| *500 002* | *6181* | 760 | TU | F | F54 | 20000401 |
| *500 003* | *6181* | 5 430 | TV | Z | Z19 | 20000419 |
| ... | ... | ... | ... | ... | ... | ... |
| *500 149* | *6181* | 4 540 | TY | A | A23 | 20020211 |
| *500 150* | *6181* | 2 400 | TZ | C | C12 | 20020214 |

Figure 4 – Data matrix Transactions

Each client having a row in the data matrix Loans (see Figure 1) has several dozens of transactions described in the data matrix Transactions. The client number 1 has 200 transactions: 1, …, 200. The client number 2 has 100 transactions: 201, …, 300 and the client number 6181 has 150 transactions 500 002 – 500 150. Let us remark that the data matrices are related by the attribute Client. The attribute Client can be understood as a function from data matrix Transaction to data matrix Loans. It assigns a row from data matrix Loans to each row of data matrix Transactions.

Each transaction is described by 5 attributes. The attribute T_Amount describes amount of transferred money (in CZK). The attribute Type describes the type of transaction. There are 6 types of transactions. The attribute Bank describes the bank from/to the transaction goes; there are 26 particular banks A, …, Z. The attribute Account describes the account of the bank from/to the transaction goes. There are several dozens of accounts in each bank. The attribute Date gives the date of the transaction in the format YYYYMMDD.

Various attributes concerning particular clients can be derived from transactions. An example is the attribute Average[Client; T_Amount; Bank(A)]. The value of the attribute Average[Client; T_Amount; Bank(A)])] for the client 1 is the average of values of the attribute T_Amount for all transactions concerning the client 1 (i.e. transactions 1, … 200) such that the value of the attribute Bank is A. The value of the attribute Average[Client; T_Amount; Bank(A)])] for other clients is defined analogously.

The attribute Average[Client; T_Amount; Bank(A)])] can be used in the tasks similar to the task solved in section 4. We can define new categories *small*, *average*, *high* and *very high* e.g. as intervals <0,1 000), <1000, 5000), <5000, 10000), <10 000, inf.). These transformations can be done by the module DataSource, see section 4.

The attribute Average[Client; T_Amount; Bank(A)] is defined using *two relations* (i.e. data matrices) Loans and Transaction connected by the function Client (see above). Thus the association rule

$$\text{District}(\textit{Prague}) \wedge \text{Average[Client; T\_Amount; Bank(A)]}(\textit{small}) \Rightarrow_{0.95;30} \text{Quality}(\textit{bad})$$

concerning data matrix Loans can be understood as a *multi-relational association rule*. It means that at least 95 per cent of clients living in Prague and having small average amount of transactions with bank A have bad quality of loans and that there are at least 30 such clients.

Let us remark that there are 26 banks A, B, …, Z. Thus we can define further 25 analogous attributes Average[Client; T_Amount; Bank(B)] ,…, Average[Client; T_Amount; Bank(Z)]. The set of all 26 such defined attributes can be coded by Average[Client; T_Amount; Bank(?)]. This set is called *a family of attributes (at data matrix Loans) 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 attribute

Average[Client; T_Amount; TYPE(TX), Bank(A)].

The value of the attribute

Average[Client; T_Amount; TYPE(TX), Bank(A)]

for the client 1 is the average of values of the attribute T_Amount for all transactions concerning the client 1 (i.e. transactions 1, … 200) such that the value of the attribute Bank is A and the value of the attribute TYPE is TX. There are 26 particular banks and 6 particular types of transactions. Thus the family of attributes Average[Client; T_Amount; TYPE(?), Bank(?)] has 156 members.

We can also use further functions e.g. Sum, Min etc. to define further families of attributes
Sum[Client; T_Amount; TYPE(?), Bank(?)],
Min[Client; T_Amount; TYPE(?), Bank(?)] etc.

We can also use 4ft-quantifier to define family of attributes. We show only very simple example. Let us start with the conditional association rule

$$\text{Bank(A)} \Rightarrow_{0.8;10} \text{Account(A17)} / \text{Client(1)}.$$

This conditional association rule concerns the data matrix Transactions. It is true if the association rule $\text{Bank(A)} \Rightarrow_{0.8;10} \text{Account(A17)}$ is true in the data matrix Transactions / Client(1) (see the end of section 2). In other words the association rule

$$\text{Bank(A)} \Rightarrow_{0.8;10} \text{Account(A17)} / \text{Client(1)}$$

is true if at least 80 per cent of transactions of the client 1 with the bank A concerns account A17 and if there are at least 10 such transactions.

Let us emphasize that the association rule

$$\text{Bank(A)} \Rightarrow_{0.8;10} \text{Account(A17)} / \text{Client(1)}$$

defines a Boolean value for the client 1. Thus this way we can define a Boolean attribute $\text{Bank(A)} \Rightarrow_{0.8;10} \text{Account(A17)}$ on the data matrix Loans. The value of the attribute $\text{Bank(A)}\Rightarrow_{0.8;10}\text{Account(A17)}$ for the object – client C is the value of the conditional association rule $\text{Bank(A)}\Rightarrow_{0.8;10}\text{Account(A17)} / \text{Client(C)}$ in the data matrix Transactions.

Let us suppose that we defined the set SET_IR of interesting association rules (not conditional) on the data matrix Transactions by means of 4ft-Miner (see section 3). Each association rule $\alpha \in$ SET_IR defines a Boolean attribute B_$\alpha$ on data matrix Loans. The value of Boolean attribute B_$\alpha$ for the client C is equal to the value of conditional association rule $\alpha$ / Client(C) on the data matrix Transactions. The set of all Boolean attributes B_$\alpha$ is called *family of attributes (at data matrix Loans) defined by the 4ft-quantifier*.

Attributes that are members of families of attributes defined by the function and of families of attributes defined by the 4ft-quantifier are called *multi-relational attributes*. We can define the set of interesting multi-relational association rules by modification of the definition of the set of interesting association rules given in section 3. The precise definition is out of range of this paper. We only outline main features of such definition:

- Two data matrices are given: parent data matrix (e.g. Loans) and child data matrix (e.g. Transactions). There is a function assigning a row from the child data matrix to each row of the parent data matrix. (The attribute Client realizes this function for the matrices Transactions and Loans.)
- At least one family of attributes at the parent data matrix is defined using the child data matrix.
- Definition of all antecedents to be generated consists of:
    - A list of attributes *and families of attributes* from which literals of antecedent will be generated,
    - Simple definition of the set of all literals to be generated from each attribute and from each family of attributes.
    - Minimal and maximal number of literals in antecedent.
- Definitions of all succedents and of all conditions be automatically generated are analogous.

There are some activities concerning modification of the procedure 4ft-Miner such that the modified procedure will mine for multi relational association rules. It is important that main tools used in the original 4ft-Miner can be applied (namely representation of analysed data by suitable strings of bits).

Let us also mention that multi-relational association rules can be understood as formulae of special logical calculus. The association rules can be understood as formulae of modified predicate calculi (see previous section). The multi-relational association rules can be understood as formulae of modified many – sorted predicate calculi. There are also interesting logical properties of calculi of multi-relational association rules analogous to logical properties of association rules [7], [8] and [9].

## Literature

[1] Aggraval, R. et all.: Fast Discovery of Association Rules. in Fayyad, U. M. et al.: Advances in Knowledge Discovery and Data Mining. AAAI Press / The MIT Press, 1996. pp 307-328

[2] Hájek, P. - Havránek T.: Mechanising Hypothesis Formation - Mathematical Foundations for a General Theory. Springer-Verlag, 1978, 396 pp.

[3] Hájek, P. - Havránek T., Chytil M.: Metoda GUHA. Praha, Academia, 1983, 314 p. (in Czech)

[4] Hájek, P. - Sochorová, A. - Zvárová, J.: GUHA for personal computers. Computational Statistics & Data Analysis 19, 1995, 149 - 153

[5] Mendelson, E.: Introduction to Mathematical Logic. Princeton, D. Van Nostrand Company, Inc., 1964

[6] Rauch J.: Some Remarks on Computer Realisations of GUHA Procedures. International Journal of Man-Machine Studies, 10, 1978, pp. 23 - 28

[7] Rauch, J.: Logical Calculi for Knowledge Discovery in Databases, in Principles of Data Mining and Knowledge Discovery, (J. Komorowski and J. Zytkow, eds.), Springer Verlag, Berlin, 47-57, 1997.

[8] Rauch, J.: Classes of Four-Fold Table Quantifiers. In Principles of Data Mining and Knowledge Discovery, (J. Zytkow, M. Quafafou, eds.), Springer-Verlag, 203-211, 1998.

[9] Rauch, J.: Four-fold Table Calculi and Missing Information. In JCI'S98 Association for Intelligent Machinery, Vol. II. Red WANG Paul, Durham, Duke University 1998.

[10] Rauch, J., Šimùnek M.: Mining for 4ft Association Rules. In Discovery Science 2000. Red. Arikawa, S. – Morishita S. Springer Verlag 2000, pp. 268 - 272

[11] Rauch, J. - Šimùnek M.: Mining for 4ft Association Rules by 4ft-Miner. in: INAP 2001, The Proceeding of the International Conference On Applications of Prolog. Prolog Association of Japan, Tokyo October 2001, pp. 285 - 294

[12] Yasuhara, A.: Recursive Function Theory and Logic. Academic Press, New York 1971, 338 p.

[13] Zembowicz R. - Zytkow J.: From Contingency Tables to Various Forms of Knowledge in Databases. in Fayyad, U. M. et al.: Advances in Knowledge Discovery and Data Mining. AAAI Press/ The MIT Press, 1996. s. 329 - 349.

*This paper has been supported by project LN00B107 of the Ministry of Education of the Czech Republic and by the grant EU No IST-1999-11495: Data Mining and decision support for business competitiveness: Solomon European Virtual Enterprise.*

# Sampling in Data Mining

**(An Invited Tutorial)**

**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.

$$\frac{n}{N}\frac{(n-1)}{(N-1)}\cdots\frac{1}{(N-n+1)}=\frac{n!(N-n)!}{N!}=\frac{1}{C_n^N} \tag{3.1}$$

### 3.1 Some basic estimates

Although sampling is undertaken for various different purposes, the most interested measures and their estimators are the following five characteristics of populations:

(1) Population mean $\overline{X}=\sum_{i=1}^{N}X_i$ and

the estimator of $\left(\overline{X}\right)=\overline{x}=\sum_{i=1}^{n}x_i/n$, the sample mean.

(2) Population total *X* and

the estimator of $\left(X\right)=N\overline{x}=\sum_{i=1}^{n}x_i/n$ .

(3) Population ratio *R* and

the estimator of $(R)=\overline{X}/\overline{x}$.

(4) Proportion of units that fall into some predefined categories (population of cars with cruise control).

(5) The variance of the $x_i$ in a finite population is usually defined as

$$\sigma^2=\frac{\sum_{1}^{N}\left(x_i-\overline{x}\right)^2}{N} \text{ or } S^2=\frac{\sum_{1}^{N}\left(x_i-\overline{x}\right)^2}{N-1} \tag{3.2}$$

the $S^2$ is used by those who approach sampling theory by means of the analysis of variance.

### 3.2 Properties

(1) $E\left(\overline{x}\right)=\overline{X}$, the sample mean $\overline{x}$ is an unbiased estimate of $\overline{X}$.

(2) $E(N\overline{x})=X$, the $N\overline{x}$ is an unbiased estimate of the population total *X*.

(3) The variance of the mean $\overline{x}$ from a simple random sample is

$$V\left(\overline{x}\right)=E(\overline{x}-\overline{X})^2=\frac{S^2(N-n)}{nN}=\frac{S^2}{n}(1-f), \tag{3.3}$$

where $f = n/N$ is the sample fraction.

(4) In a simple random sample, the sample variance

$$s^2=\frac{\sum_{1}^{n}\left(x_i-\overline{x}\right)^2}{n-1}$$ is an unbiased estimate of

$$S^2=\frac{\sum_{1}^{N}\left(x_i-\overline{X}\right)^2}{N-1} \tag{3.4}$$

(5) Unbaised estimates of variances of $\overline{x}$ and $N\overline{x}$ are

$$v(\overline{x})=\frac{s^2}{n}(1-f)$$

and $$v(N\overline{x})=\frac{N^2s^2}{n}(1-f), \tag{3.5}$$

respectively.

### 3.3 Random sampling with replacement

Let $t_i$ be the number of *i*-th unit appears in the sample. Then

$$\overline{x}=\frac{1}{n}\sum_{i=1}^{N}t_iy_i\,, E(t_i)=n/N, \tag{3.6}$$

$$V(t_i)=n(\frac{1}{N})(1-\frac{1}{N}),\ Cov(t_it_j)=-\frac{n}{N^2}, \tag{3.7}$$

and

$$V(\overline{x})=\frac{N-1}{N}\frac{S^2}{n} \tag{3.8}$$

## 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 hypergeometric distribution.

### 4.1 Some basic Estimates

(1) Population and sample means

For a given unit $x_i$ in the sample or population, we define

$$x_i = \begin{cases} 1, & if \quad x_i \in G, \\ 0, & if \quad x_i \notin G'. \end{cases}$$

Thus, the population mean and the sample mean are given

$$\overline{X} = \frac{\sum_1^N x_i}{N} = \frac{A}{N} = P \text{ , and } \overline{x} = \frac{\sum_1^n x_i}{n} = \frac{a}{n} = p. \qquad (4.1)$$

(2) Population and sample standard errors

$$S^2 = \frac{\sum_1^N (x_i - \overline{X})^2}{N-1} = \frac{\sum_1^N x_i^{\,2} - N\overline{X}^2}{N-1} = \frac{NP - NP^2}{N-1}$$

$$= \frac{N}{N-1} PQ, \quad where \quad Q = 1 - P. \qquad (4.2)$$

$$s^2 = \frac{\sum_1^n (x_i - \overline{x})^2}{n-1} = \frac{n}{n-1} pq, \qquad (4.3)$$

where $q = 1 - p$.

(3) The variance of $p$

$$V(p) = E(p - P)^2 = \frac{S^2}{n}\frac{N-n}{N} = \frac{N-n}{n(N-1)} PQ. \qquad (4.4)$$

(4) The variance of *Np* in the group *G*

$$V(Np) = \frac{N^2 PQ}{n}\frac{N-n}{N-1} \qquad (4.5)$$

(5) An unbiased estimate of the variance of $p$ from the sample.

$$v(p) = s_p^{\;2} = \frac{N-n}{(n-1)N} pq \qquad (4.6)$$

### 4.2 Properties

(1) The sample proportion $p = a/n$ is an unbiased estimate of the population proportion $P = A/N$.

(2) If *p* and *P* are the sample and population percentages, respectively, for the group *G*. The variance of *p* is

$$v(p) = s_p^{\;2} = \frac{N-n}{(n-1)N} pq \qquad (4.7)$$

Two important distributions, the binomial and the hypergeometric distributions can characterize the properties of sampling proportions and percentages.

### 4.3 The binomial distribution

Let consider a sampling of *n* number of units we want to be selected from the population of size *N*, and satisfies the following four conditions:

(1) Each selection unit must belong to either *G* or *G*’.
(2) The probability that a unit is selected from *G* is the same at each selection.
(3) There are *n* selections and *n* is a constant.
(4) All these *n* selections are independent.

The probability function satisfies above conditions is called the binomial distribution.

$$b(x; n, p) = \frac{n!}{a!(n-a)!} p^x q^{n-x}. \;\; x = 0, 1, 2, \ldots, n. \qquad (4.8)$$

where the *a* is the number of units in the sample and were in the group *G*.

The mean and the variance of the binomial distribution random variable *X* are $E(X) = np$ and $V(X) = npq$, respectively.

### 4.4 The hypergeometric distribution

Another important discrete probability distribution function is the hypergeometric distribution. Both the binomial and hypergeometric distributions have applications in the manufacture quality control.

Let us consider a sampling from finite population of *N* objects which can be classified according to some characteristic into *M* of one group and *N* – *M* of another. For example, we might partition manufactured items as defective and non-defective items. Consider *n* be the number of sample size; the total number of possible samples of size is given by $C_n^N$ . The number of samples which will contain exactly x defectives is $C_x^M \times C_{n-x}^{N-M}$ which is the product of the number of ways *x* defectives can be selected from the *M* defectives present and the number of ways in which the remaining (*n* – *x*) non-defective can be chosen from ($N – M$) non-defective items. Then the probability that the selection of $x$ defective items out of sample size is defined by the following probability mass function:

$$h(x;n,N) = \frac{C_x^M C_{n-x}^{N-M}}{C_n^N}, \tag{4.9}$$

where $x = 0, 1, 2,\ldots, n$.

The mean of a hypergeometric random variable $X$ is

$$E(X) = \sum_{k=0}^{n} k \frac{C_k^M C_{n-k}^{N-M}}{C_n^M} = nM / N. \tag{4.10}$$

Similarly, the variance of X can be shown to be

$$V(X) = n\frac{M}{N}\frac{N-M}{N}\frac{N-n}{N-1}. \tag{4.11}$$

Consider a special case, we set $n = N$, we have

$$E(X) = NM / N = M$$

and $$V(X) = \frac{M(N-M)(N-N)}{N\times N\times N} = 0 \tag{4.12}$$

For a special case, if the sampling is done with replacement, then $X$ would be a binomial random variable with parameters $n$ and $p = M/N$, then the mean $E(X) = np = nM/N$ and the variance

$$V(X) = npq = n\,(M / N)(N - M) / N. \tag{4.13}$$

## 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%.

$$V(p) \approx \frac{PQ}{n}, \quad \sigma(p) \approx \sqrt{PQ/n} \tag{5.1}$$

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 $\alpha$ that the actual error is larger than $d$, thus we have the probability

$$P_r(|p-P| \geq d) = \alpha \tag{5.2}$$

For simple random sampling and $p$ is taken as normally distributed. From the last section we have

$$\sigma_p = \sqrt{\frac{N-n}{N-1}}\sqrt{\frac{PQ}{n}} \tag{5.3}$$

Hence, the formula that connects $n$ with the desired degree of precision is

$$d = t\sqrt{\frac{N-n}{N-1}}\sqrt{\frac{PQ}{n}}, \tag{5.4}$$

where $t$ is the abscissa of the normal curve that cuts off the α at the tail end. Solve for $n$ and we have

$$n = \frac{\dfrac{t^2PQ}{d^2}}{1+\dfrac{1}{N}\left(\dfrac{t^2PQ}{d^2}-1\right)} \tag{5.5}$$

For practical use, we may take $p$ of $P$ substituted in the above formula. If $N$ is large, let

$$n_0 = \frac{t^2pq}{d^2} = \frac{pq}{V} \tag{5.6}$$

and this is called initial estimate. Hence,

$$n = \frac{n_0}{1+(n_0-1)/N} \approx \frac{n_0}{1+\left(n_0/N\right)} \qquad (5.7)$$

Consider a similar example as in (1), let

$$d = 0.05, \qquad p = 0.5, \qquad \alpha = 0.05, \qquad t = 2$$

$$n_0 = \frac{4\times 0.5\times 0.5}{0-0.0025} = 400$$

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

$$P_r\left(\left|\frac{\bar{x}-\bar{X}}{\bar{X}}\right| \ge r\right) = P_r\left(\left|\bar{x}-\bar{X}\right| \ge r\bar{X}\right) = \alpha \qquad (5.7)$$

to be a small probability. We assume that $\bar{x}$ is normally distributed, from Section 3, we have

$$V(\bar{x}) = E(\bar{x}-\bar{X})^2 = \frac{S^2(N-n)}{nN}\text{, or}$$

$$\sigma_{\bar{x}} = \sqrt{\frac{N-n}{n}}\,\frac{S}{\sqrt{n}} \qquad (5.8)$$

Thus,

$$r\bar{X} = t\sigma_{\bar{x}} = t\sqrt{\frac{N-n}{N}}\,\frac{S}{\sqrt{n}} \qquad (5.9)$$

Hence, solving for n, we have

$$n = \left(\frac{tS}{r\bar{X}}\right)^2 \Bigg/ \left[1+\frac{1}{N}\left(\frac{tS}{r\bar{X}}\right)^2\right] \qquad (5.10)$$

For the first approximation we take

$$n_0 = \left(\frac{tS}{r\bar{X}}\right)^2 \qquad (5.10)$$

Hence,

$$n = \frac{n_0}{1+(n_0/N)}\,. \qquad (5.11)$$

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

$$n_0 = \frac{t^2S^2}{r^2\bar{X}} = \frac{4\times 86.3}{1.9^2} \approx 96$$

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.

### 6.1 Properties of estimates

The population mean of all units, the estimate in stratified sampling is

$$\bar{x}_{st} = \frac{\sum_{i=1}^{L} N_i \bar{x}_i}{N} = \sum_{i=1}^{L} W_i \bar{x}_i \tag{6.1}$$

where $W_i = \frac{N_i}{N}$ is called *self-weight* sample.

The sample mean is

$$\bar{x} = \frac{\sum_{i=1}^{L} n_i \bar{x}_i}{n} \tag{6.2}$$

and in every stratum the correct weight

$$\frac{n_i}{n} = \frac{N_i}{N} \quad or \quad \frac{n_i}{N_i} = \frac{n}{N} \quad or \quad f_i = f \tag{6.3}$$

**Theorem 6.1.** If the sample estimate $\bar{x}_i$ of every stratum is unbiased estimate, then $\bar{x}_{st}$ is an unbiased estimate of the population mean $\bar{X}$.

*Proof.*

$$E(\bar{x}_{st}) = E\sum_{i=1}^{L} W_i \bar{x}_i = \sum_{i=1}^{L} W_i \bar{X}_i \tag{6.4}$$

The population mean $\bar{X}$ is

$$\bar{X} = \frac{\sum_{i=1}^{L}\sum_{k=1}^{N_i} \bar{x}_{ik}}{N} = \frac{\sum_{i=1}^{L} N_i X_i}{N} = \sum_{i=1}^{L} W_i \bar{X}_i \tag{6.5}$$

**Theorem 6.2.** If the samples are drawn independently in different strata,

$$V(\bar{x}_{st}) = \sum_{i=1}^{s} W_i^2 V(\bar{x}_i), \tag{6.6}$$

where $V(\bar{x}_i)$ is the variance of $\bar{x}_i$ over repeated samples from all stratum.

**Theorem 6.3.** The variance of the estimate $\bar{x}_{st}$ is

$$\begin{aligned} V(\bar{x}_{st}) &= \frac{1}{N^2}\sum_{i}^{L} N_i (N_i - n_i)\frac{S_i^2}{n_i} \\ &= \sum_{i=1}^{L} W_i^2 \frac{S_i^2}{n_i}(1 - f_i). \end{aligned} \tag{6.7}$$

### 6.2 The estimated variance and confidence limits

If a simple random sample is taken within each strata over the population, an unbiased estimate of $S_i^2$ is

$$s_i^2 = \frac{1}{n_i - 1}\sum_{k=1}^{n_i}(x_{ik} - \bar{x}_i^2) \tag{6.8}$$

Therefore, we have the theorem

**Theorem 6.4.** An unbiased estimate of the variance of $\bar{x}_{st}$ is

$$\upsilon(\bar{x}_{st}) = s^2(\bar{x}_{st}) = \frac{1}{N^2}\sum_{i=1}^{L} N_i (N_i - n_i)\frac{s_i^2}{n_i} \tag{6.9}$$

or

$$s^2(\bar{x}_{st}) = \sum_{i=1}^{L} \frac{W_i^2 s_i^2}{n_i} - \sum_{i=1}^{L} \frac{W_i s_i^2}{N} \tag{6.10}$$

The confidence intervals for population mean and for population total are given

$\bar{x}_{st} \pm ts(\bar{x}_{st})$ *and* $N\bar{x}_{st} \pm tNs(\bar{x}_{st})$, respectively. (6.11)

In these two formulas, we assume that the $\bar{x}_{st}$ is normally distributed, $s(\bar{x}_{st})$ is the standard error of $\bar{x}_{st}$, and $t$ is the standard score of the normal distribution.

### 6.3 Estimating sample size with continuous data

Formulas for estimated sample sizes can be considered by two different cases, one from the estimation of population mean and the other from the estimation of the total population.

**(1) Sample size from the estimation of the population mean**

Consider $s_i$ be the estimate of $S_i$ and let $w_i = n_i/n$. From the Theorem 6.3

$$V = \frac{1}{n}\sum_{i}^{L}\frac{W_i^2 s_i^2}{w_i} - \frac{1}{N}\sum_{i=1}^{L}W_i s_i^2, \tag{6.12}$$

where $W_i = N_i/N$.

From this equation we may solve for $n$ and obtain the following formula:

$$n = \frac{\sum\frac{W_i^2 s_i^2}{w_i}}{V + \frac{1}{N}\sum W_i s_i^2}$$

and let $n_0 = \frac{1}{V}\sum\frac{W_i^2 s_i^2}{w_i}$ be the initial estimate. Then we have

$$n = \frac{n_0}{1 + \frac{1}{VN}\sum W_i s_i^2} \tag{6.13}$$

For more convenient, consider presumed optimum allocation with fixed $n$: $w_i \propto W_i s_i$

$$n = \frac{\left(\sum W_i s_i\right)^2}{V + \frac{1}{N}\sum W_i s_i^2} \tag{6.14}$$

For proportional allocation with fixed $w_i = W_i = N_i/N$.

The initial approximation proportion

$$n_0 = \frac{\sum W_i s_i^2}{V} \tag{6.15}$$

Hence $$n = \frac{n_0}{1 + n_0/N}. \tag{6.16}$$

**(2) Sample size from the estimation of the population total**

The general formula;

$$n = \frac{\sum\frac{N_i^2 s_i^2}{w_i}}{V + \sum N_i s_i^2} \tag{6.17}$$

For fixed $n$

$$n = \frac{\left(\sum N_i s_i\right)^2}{V + \sum N_i s_i^2} \tag{6.18}$$

Proportional the initial approximation

$$n_0 = \frac{N}{V}\sum N_i s_i^2 \tag{6.19}$$

The sample size

$$n = \frac{n_0}{1 + n_0/N}. \tag{6.20}$$

**Example**

The department of education in a southern state is planning to estimate the total enrollments for the next academic year in order to determine the number of new teachers need. The state has six school districts each of them naturally forms a stratum. Each district has various numbers of schools from the smallest 13 to the largest 72. The data for estimating sample size are given.

| Stratum | $N_i$ | $s_i$ | $N_i s_i$ | $n_i$ |
|---|---|---|---|---|
| 1 | 14 | 325 | 4550 | 9 |
| 2 | 17 | 190 | 3230 | 7 |
| 3 | 25 | 190 | 4750 | 10 |
| 4 | 26 | 189 | 4914 | 10 |
| 5 | 42 | 83 | 3486 | 9 |
| 6 | 73 | 85 | 6205 | 13 |
| Total | 197 | | 27135 | 58 |

The target is a coefficient of variation of 5%in estimated total enrollment. The current enrollment is 56,500 pupils. Thus the desired standard error is

$$(0.05)(56500) = 2825$$

The desired variance is

V=(2825)(2825)=7980625

First, we estimate the initial approximation $n_0$.

$$n_0 = \frac{\left(\sum N_i s_i\right)^2}{V} = \frac{(27135)^2}{7980625} = 92.26$$

Hence,

$$n = \frac{n_0}{1 + \frac{1}{V}\sum N_i s_i^2} = \frac{92.26}{1 + \frac{4761753}{7980625}} = 57.78$$

A sample size of 58 is chosen. The sample of each stratum is provided in the tale above.

**(3) What are problems that suitable use stratified random sampling?**

Examples that suitable using stratified random sampling are schools in surveys in the number of students, supermarkets in surveys of number of employees, hospitals in surveys of number of beds, income tax in surveys of the among of taxable incomes, and farms surveys in the size of acreages or gross incomes.

**(4) Conditions that give large gains in precision**

It is natural to ask a question like under what conditions that can give large gains in precision for the stratified random sampling. The following three conditions are probably the answer for this question.

1. The population is composed of institution varying widely in size.
2. The major variable to be measures is closely related to the size of institutions.
3. A good measure of size is available for setting up the strata.

## 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.

### 7.1 Estimated mean and its variance

Let $\bar{x}_{sy}$ be the mean of a systematic sample, *k* be the number of clusters each containing *n* elements and $N = nk$. It is easy to verify that $\bar{x}_{sy}$ is an unbiased estimate of $\bar{X}$.

**(1) The variance of the mean of a systematic sample**

The variance of the mean of a systematic sample is

$$V(\bar{x}_{sy}) = \frac{N-1}{N} S^2 - \frac{k(n-1)}{N} S_{wsy}^2 \qquad (7.1)$$

where

$$S_{wsy}^2 = \frac{1}{k(n-1)} \sum_{i=1}^{k} \sum_{j=1}^{n} (x_{ij} - \bar{x}_i)^2 \qquad (7.2)$$

and $\bar{x}_i$ is called the mean of the mean of the *i*-th sample.

**(2) Relation between the mean of systematic sample and the mean of a simple random sample**

The mean of systematic sample is more precise than the mean of a simple random sample if and only if

$$S_{wsy}^2 > S^2$$

This is an important result and it applies to general cluster sampling. The proof of this result can be found in a standard graduate textbook.

### 7.2 Relation with other samplings

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.

### 8.1 Simple random sample of clusters-unbiased estimate

Let $M_i$ be the number of elements in the *i*-th unit. We want to estimate the population total $X$ of $x_{ij}$ that is an unbiased estimate of $X$

$$x_i = \sum_{i=1}^{M_i} x_{ij} = M_i\bar{x}_i \tag{8.1}$$

where $x_i$ is the item total for the *i*th cluster unit.

For a simple random sample of size *n* of *N* population units, an unbiased estimate of *X* is given

$$\hat{X} = \frac{N}{n}\sum_{i=1}^{n} x_i \tag{8.2}$$

By Section 3.3, the variance of $\hat{X}$ is

$$V(\hat{X}) = \frac{N^2(1-f)}{n}\,\frac{\sum_{i=1}^{n}(x_i - \bar{X})^2}{N-1} \tag{8.3}$$

where $\bar{X} = X/N$ is the population mean per cluster unit.

### 8.2 Simple random sample of clusters-ratio to size estimate

Let $M_0 = \sum_{i=1}^{N} M_i$ be the total number of elements in the population. Consider a ratio estimate $\hat{X}_R = M_0$ , the sample mean per element and the population ratio $R = X/M_0 = \bar{\bar{X}}$, this is called the population mean per element. Then the variance of $\hat{X}_R$ is

$$V(\hat{X}_R) \approx \frac{N^2(1-f)}{n}\,\frac{\sum_{i=1}^{N}(x_i - M_i\bar{\bar{X}})^2}{N-1}$$

$$\approx \frac{N^2(1-f)}{n}\,\frac{\sum_{i=1}^{N}M_i^2(\bar{x}_i - \bar{\bar{X}})^2}{N-1} \tag{8.4}$$

### 8.3 Sampling with probability proportional to size

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.

# Neural Networks for Multiclass Object Mining

*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.*

*The network behaviour is interpreted by analysing the weights in learned networks. Visualisation of these 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 not just a black box but an expression or a model of hidden patterns extracted/discovered during the data mining process.*

**Keywords** Network training, network testing, network sweeping, data mining, object detection, object recognition, object classification, target 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, 5, 7].

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 and the network internal behaviour. For this reason, a neural network is often regarded as a black box classifier or a prediction engine [5]. 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"[1] 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. For example, it may be necessary to find all tumors in a database of x-ray images, all cyclones in a database of satellite images or a particular face in a database of photographs. 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 [4, 14, 15] 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 [4, 10, 11], our objective is to mine objects from a number of classes.

Neural networks have been applied to object classification and mining problems [1, 9, 10, 13]. In these

[1] The term *pattern* appeared in this paper refers to some high level description of knowledge, such as features, rules, or other expression or model of data. It is quite different from the term "pattern" commonly used in pattern recognition.

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/detection 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. For example, a mining/detection system looking for grey squares in Figure 1 (a) may report that there are 25. If 9 of these are correct the detection rate will be $(9/18) * 100 = 50\%$. The false alarm rate will be $(16/18) * 100 = 88.9\%$. 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 [11, 13]. 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 problem (both object localisation and classification). Thus, it is inappropriate to simply compare the results in this paper with those obtained by other systems which are developed only for the classification problem. It is also inappropriate to directly compare the performance achieved in this paper for multiclass detection problems with the performance for one class detection problems reported in other papers.

### 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 2.1). Specifically, we investigate:

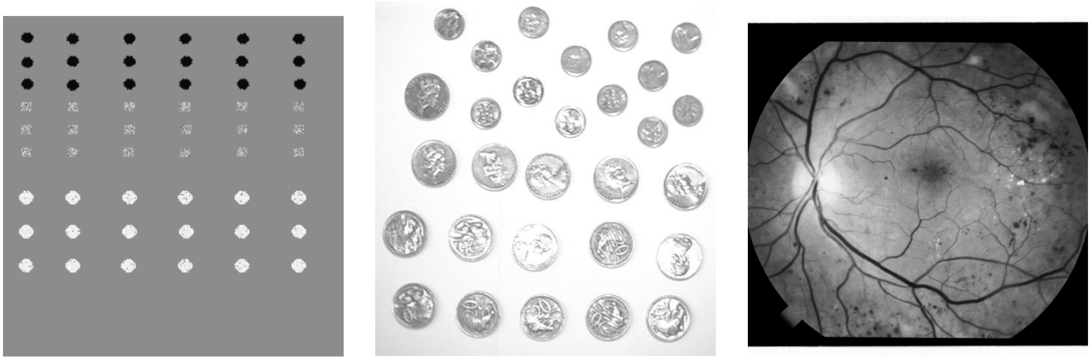
- Will this approach work for a sequence of multiclass object mining problems of increasing difficulty?
- Will the performance deteriorate as the degree of difficulty of the detection problems increases?
- Can the weights in learned networks be interpreted in some ways and "hidden patterns" be successfully discovered and represented?
- Will the number of training examples affect object mining performance?

In the remainder of this paper, we first describe the image databases and our neural network approach, then present a set of experimental results. After making a detailed analysis and interpretation of the learned network weights, we conclude this approach and give a number of future directions.

## 2 The Image Databases

### 2.1 Three 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 ($class1$), grey squares ($class2$) and white circles ($class3$). 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 $head005$), the head side of 20 cent coins (class $head020$), the tail side of 5 cent coins (class $tail005$) and the tail side of 20 cent coins (class $tail020$). All the objects in each class have a similar size. They



Number of Images: 7
Number of Objects: 240
Number of Classes: 4
Image Size: 700x700
Input Field Size: 14x14

Easy (Circles and Squares)
(a)

Number of Images: 20
Number of Objects: 600
Number of Classes: 5
Image Size: 640x480
Input Field Size: 14x14

Medium Difficulty (Coins)
(b)

Number of Images: 15
Number of Objects: 164
Number of Classes: 3
Image Size: 1024x1024
Input Field Size: 14x14

Very Difficult (Retinas)
(c)

Figure 1: Object Detection Problems of Increasing Difficulty

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 very cluttered background. The objective is to find two classes of retinal pathologies – haemorrhages (class *haem*) and micro aneurisms (class *micro*). To give a clear view of representative samples of the target objects in the retina images, one sample piece of these images is presented in Figure 2. In this figure, haemorrhage and micro-aneurism examples are labeled using white surrounding squares. Note that in each of the databases the background (*non-object*) counts as a class (class *other*), but not a class of interest.

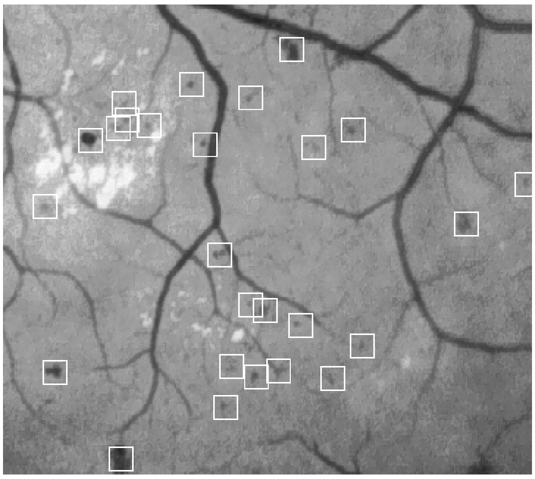


Figure 2: An enlarged view of one piece of the retina images

## 2.2 Image Data Sets

To avoid confusion, we define a number of terms related to the image data.

A set of entire images in a database constitutes an *image data set* for a particular problem domain. In this paper, it is randomly split into two parts: a *detection training set*, which is used to learn a detector, and a *detection test set*, which is used for measuring object detection performance. *Cutouts* refer to the object examples which are cut out from a detection training set. These cutouts form a *classification data set*, which is randomly split into two parts: a *classification training set* used for network training, and a *classification test set* for network testing. The latter set plays two roles: one is to measure object classification performance, the other is used as a "tuning/validation set" for monitoring network training process in order to obtain good parameters of the learned network for object mining in large images. An *input field* refers to a square within a large image. It is used as a moving window for the network sweeping (detection) process. The size of the input field is the same as that of the cutouts for network training. The relationships between the various data sets are shown in Figure 3.

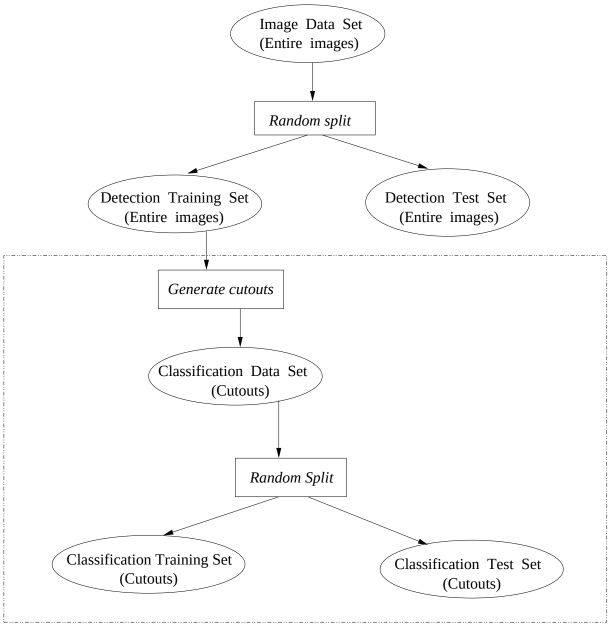


Figure 3: Relationship between the classification and detection data sets

# 3 The Method

## 3.1 Overview

An overview of the approach is shown in Figure 4. A brief outline of the method is as follows:

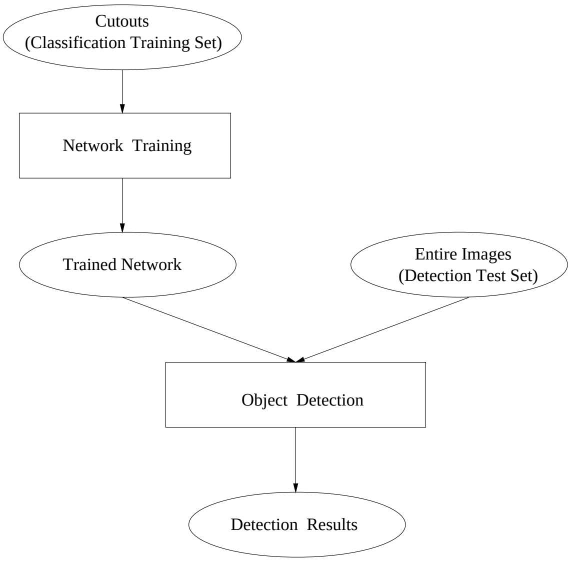


Figure 4: Overview of the approach

1. Assemble a database of images in which the locations and classes of all the objects of interest are manually determined. Divide these full 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. One one hidden layer is used in this approach.[2] The number of hidden nodes is empirically determined.

4. Train the network by the back propagation algorithm [12] on the classification training data. The trained network is then tested on the classification test set to measure the object classification performance. The classification test data is also used to avoid overtraining the network. 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. It is important to note that the thresholds for different classes are different.

6. Evaluate the object mining performance of the network by calculating the detection rate and the false alarm rate.

## 3.2 Object Mining/Detection

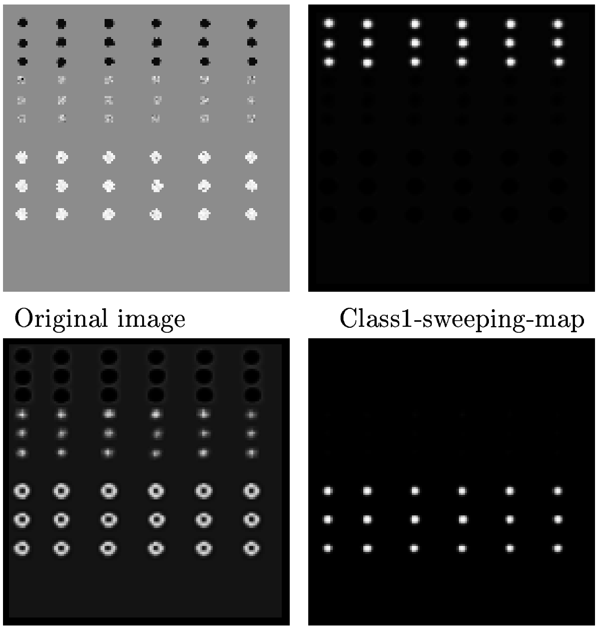
While object classification corresponds to network training and network testing on the cutouts in the classification data sets, object mining/detection corresponds to network sweeping on the entire images in the detection test set, which were not used in any

[2] The theoretical results provided by Irie and Miyake [6] and Funahashi [3] have proved that any continuous mapping can be approximated by a network with a single hidden layer.

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 5. 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 5: 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,[3] 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.

## 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. During the object mining procedure, various thresholds result in different detection results. The higher the threshold, the fewer the objects that can be detected by the trained network, which results in a lower detection rate but also a lower false alarm rate. Similarly, the lower the threshold is selected, the higher the detection rate and the higher the false alarm rate are produced. Thus there is generally a trade-off between the detection rate and the corresponding false alarm rate.

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

[3] 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.

| 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.

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* (black circles) and *class 3* (white circles) did not produce any false positives (false alarm rates are zero), while detecting *class2* (grey squares) resulted in a 91.2% false alarm rate on average.

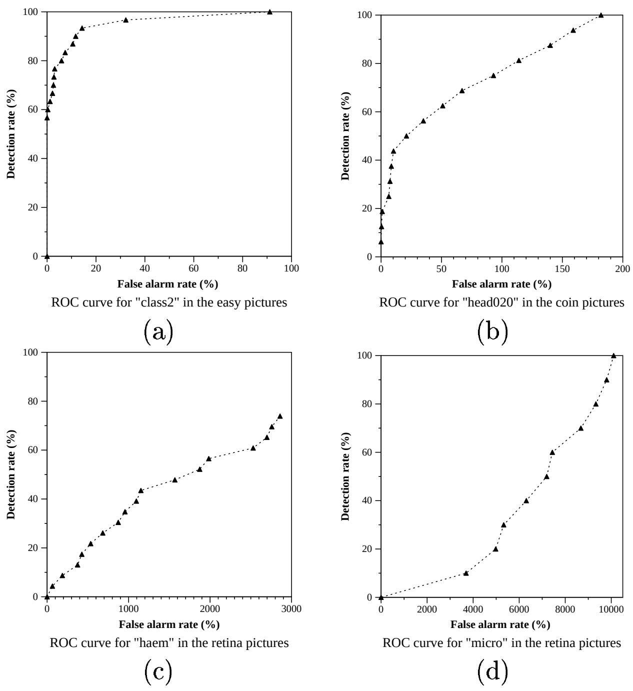
| Easy Images | Object Classes | | |
|---|---|---|---|
| | Class1 | Class2 | Class3 |
| Detection Rate(%) | 100 | 100 | 100 |
| False Alarm Rate(%) | 0 | 91.2 | 0 |

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.

Compared with the performance of the easy and coin images, the results of the two classes *haem* and *micro* in the very difficult retina images are disappointing. All the objects of class *micro* were correctly detected (a detection rate of 100%), however, with a very high false alarm rate. The best detection rate for class *haem* was only 73.91%. Even at a detection rate of 50%, the false alarm rate was still quite high.

To give an intuitive view, the extended ROC curves (Receiver/Relative Operating Characteristic curve [8, 16]) for class *class2* in the easy images, class *head020* in the coin images and classes *haem* and *micro* in the retina images are presented in Figure 6 (a), (b), (c) and (d) respectively.



(a) (b) (c) (d)

Figure 6: Detailed results (extended ROC curves) for some "difficult" classes in the three databases

### 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 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 pro-

| 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.

duced. 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 $7\times 7$ to $14\times 14$ pixels). Another reason that we did not obtain good results on the retina images is the insufficient number of training object examples (only 164, see Figure 1).

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. In the coin images, for example, the trained networks achieved perfect object classification results, that is, 100% accuracy on classifying all the objects in the four classes. While mining 5 cent coins achieved ideal performance, that is, all objects of the two classes (*head005, tail005*) were correctly mined with zero false alarm rate, mining the two 20 cent coin classes (*head020, tail020*) produced a number of false positive objects. The results for the other databases showed a similar pattern. This is due mainly to the fact that multiclass object mining/detection task is generally much more difficult than only the classification task on the same problem domains since multiclass object mining includes both object classification and object localisation.

## 5 Visualisation and Analysis of Learned Network Weights

To analyse why this approach can be used for multiclass object mining/detection, 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 7 shows the network architecture (576-3-5) used in the coin images. The weight groups between the input nodes and the hidden nodes and between the hidden nodes and the output nodes are also presented. The weight matrices (a), (b), (c) and (d) shown in Figure 8 correspond to weight groups (a), (b), (c) and (d) in the network architecture in this figure.

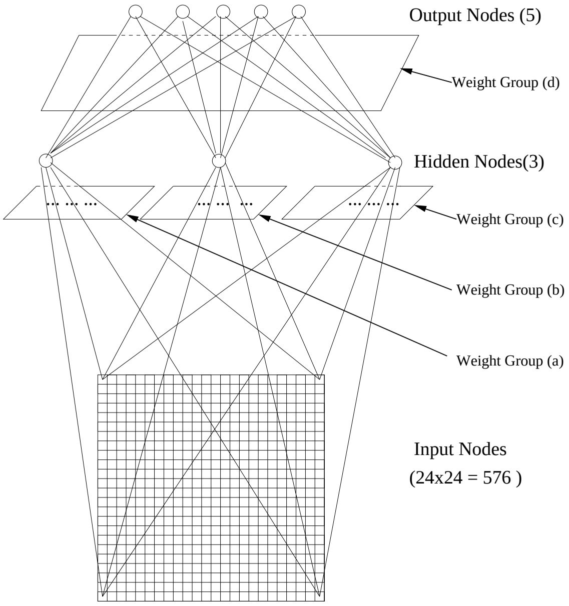


Figure 7: Network architecture with four weight groups for object mining/detection in the coin images.

Figure 8 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 $24\times 24$ matrix (corresponding to the $24\times 24$ input field) to facilitate visualisation. Figure 8 (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 8 (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

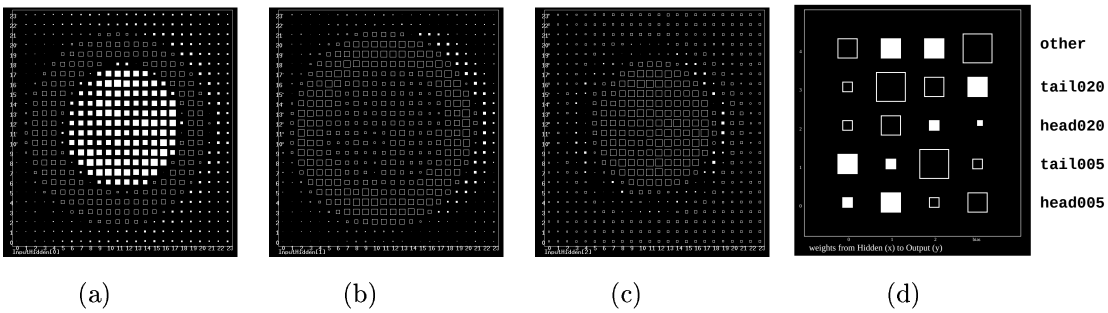


Figure 8: Weights in a trained network for object mining/detection in the coin images.

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 8 (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 reveals 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.[4]

## 6 Conclusions

This paper presented a pixel based approach for mining multiple class objects in large images using multilayer feed forward neural networks. The back propagation algorithm was used to train the network on the subimages which had been cut out from the large images. The trained network was then applied, in a moving window fashion, over the entire images to detect the objects of interest. Object detection performance was measured on the large images in the detection test set. 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.[5] As expected, 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 advantages:

- Raw image pixel data are used as inputs to neural networks, and accordingly traditional specific feature extraction and selection is avoided.

[4] A mathematical model of the patterns in learned neural networks can be seen from Appendix A.

[5] Strictly speaking, we can not make this conclusion, since the number of training examples is not sufficient. This will be addressed later in this section.

- 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.

The approach also has a number of disadvantages, which need to be addressed in the future:

- 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.
- In this approach, the network was trained based on the object cutouts but the trained network was directly applied to the entire images. This might be one of the main reasons which result in many false positives in some difficult detection problems. We are investigating ways of refining the trained networks based on the entire images to improve the detection performance.
- The insufficient number of training examples led to poor object mining performance in the retina images. It would be very interesting to investigate whether this approach can achieve good results using sufficient training examples in the future — this will reveal whether this approach can be used to mine/detect complex and irregular objects against a highly cluttered background.

## Acknowledgement

I would like to thank Dr. Peter Andreae, Dr. Dong Qian Wang 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] Ken-Ichi Funahashi. On the approximate realization of continuous mappings by neural networks. *Neural Networks*, Volume 2, Number 3, pages 183–192, 1989.

[4] 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.

[5] Robert Groth. *Data Mining: Building Competitive Advantage*. Prentice Hall PTR, 2000.

[6] B. Irie and S. Miyake. Capability of three-layered perceptrons. In *Proceedings of the IEEE 1988 International Conference on Neural Networks*, pages I(641–648), New York, 1988. IEEE. Vol. 1.

[7] Jiawei Jan and Michaeline Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[8] Charles E. Metz. ROC methodology in radiologic imaging. *Investigative Radiology*, Volume 21, Number 9, pages 720–732, September 1986.

[9] Steven K. Rogers, John M. Colombi, Curtis E. Martin, James C. Gainey, Ken H. Fielding, Tom J. Burns, Dennis W. Ruck, Matthew Kabrisky and Mark Ocley. Neural networks for automatic target recognition. *Neural Networks*, Volume 8, Number 7/8, pages 1153–1184, 1995.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Yonggwan Won, Paul D. Gader and Patrick C. Coffield. Morphological shared-weight networks with applications to automatic target recognition. *IEEE Transactions on neural networks*, Volume 8, Number 5, pages 1195–1203, September 1997. ISSN 1045-9227.

[16] Mengjie Zhang. *A Domain Independent Approach to 2D Object Detection Based on the Neural and Genetic Paradigms*. Ph.D. thesis, Department of Computer Science, RMIT University, Melbourne, Australia, August 2000.

# A Pattern Modelling

Section 5 visualised the learned features as weight matrices, which can be considered hidden patterns extracted from learning. With the analysis of these patterns, we showed that a learned network was an expression of these patterns. In this appendix, we investigate a formal mathematical model. To do so, we define the following symbols:

$\vec{I}$: the network input data. Its dimension is the number of pixels in the input field or the training object cutout. This number is the same as the number of input nodes of the network. In the standard multilayer feed forward back propagation networks, $\vec{I}$ is also the output of the network input nodes.

$\vec{O}$: the network output vector. Its dimension is the number of output nodes defined in the network.

$\vec{H}$: the output of the hidden nodes. As mentioned earlier, only one hidden layer is used in our approach.

$f$: the transfer function. In the back propagation algorithm, $f$ is the sigmoid function, that is, $f(x) = \frac{1}{1+e^{-x}}$. Here we extend the transfer function as a vector $\vec{f}$:

$$\vec{f}(\vec{x}) = (\frac{1}{1+e^{-x_1}}, \frac{1}{1+e^{-x_2}}, ..., \frac{1}{1+e^{-x_n}}) \qquad (1)$$

where $\vec{x} = (x_1, x_2, ..., x_n)$.

$\vec{W}_H$: the weights between the input layer and the hidden layer. The dimension of this vector is the number of hidden nodes times the number of input nodes.

$\vec{W}_O$: the weights between the hidden layer and the output layer. The dimension is the number of output nodes by the number of the hidden nodes.

Based on these definitions, the network output will be:[6]

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{H}) \qquad (2)$$

where:

$$\vec{H} = \vec{f}(\vec{W}_H \cdot \vec{I}) \qquad (3)$$

Combining Equations 2 and 3, we obtain:

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{f}(\vec{W}_H \cdot \vec{I}) \qquad (4)$$

Generally, for any three layer learned network for object mining problems, we have:

$$\vec{W}_H = (\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}) \qquad (5)$$

where $\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}$ are the weight matrices similar to those in Figure 8 (a), (b) and (c), and $m$ is the number of hidden nodes in the learned network. In this case, the network output is:

$$\vec{O} = \vec{f}(\vec{W}_O \cdot \vec{f}((\vec{W}_{h_1}, \vec{W}_{h_2}, ..., \vec{W}_{h_m}) \cdot \vec{I})) \qquad (6)$$

For the coin image example, $\vec{I}$ is a 576 dimensional vector which corresponds to a two dimensional array of 24×24 with the values of (normalised) object pixels. $\vec{O}$ is a five dimensional vector which corresponds to the five classes. $\vec{W}_O$ corresponds to the weight matrix $\vec{W}_d$ in Figure 8 (d). $\vec{W}_H$ is a three dimensional vector:

$$\vec{W}_H = (\vec{W}_a, \vec{W}_b, \vec{W}_c) \qquad (7)$$

where $\vec{W}_a, \vec{W}_b$ and $\vec{W}_c$ are hidden patterns extracted, corresponding to the weight matrices shown in Figure 8 (a), (b) and (c) respectively. Each of them has 24×24 weights. Thus, for the object mining problems in the coin images, we have:

$$\vec{O} = \vec{f}(\vec{W}_d \cdot \vec{f}((\vec{W}_a, \vec{W}_b, \vec{W}_c) \cdot \vec{I})) \qquad (8)$$

Clearly, the network output is a function of $\vec{W}_a, \vec{W}_b, \vec{W}_c, \vec{W}_d$ and $\vec{I}$. Thus, if these weight matrices are regarded as features or patterns extracted from learning, then the learned network is a model of these patterns.

[6] For presentation convenience, we do not put the biases in these equations. This does not change our conclusion.