Domain-Driven Classification Based on Multiple Criteria and Multiple Constraint-Level Programming for Intelligent Credit Scoring

Domain Driven Classification based on Multiple Criteria and Multiple Constraint-level Programming for Intelligent Credit Scoring

Jing He,Yanchun Zhang, Yong Shi, *Senior Member, IEEE*, Guangyan Huang, *Member, IEEE*

Abstract—Extracting knowledge from the transaction records and the personal data of credit card holders has great profit potential for the banking industry. The challenge is to detect/predict bankrupts and to keep and recruit the profitable customers. However, grouping and targeting credit card customers by traditional data-driven mining often does not directly meet the needs of the banking industry, because data-driven mining automatically generates classification outputs that are imprecise, meaningless and beyond users' control. In this paper, we provide a novel domain-driven classification method that takes advantage of multiple criteria and multiple constraint-level programming for intelligent credit scoring. The method involves credit scoring to produce a set of customers' scores that allows the classification results actionable and controllable by human interaction during the scoring process. Domain knowledge and experts' experience parameters are built into the criteria and constraint functions of mathematical programming and the human and machine conversation is employed to generate an efficient and precise solution. Experiments based on various datasets validated the effectiveness and efficiency of the proposed methods.

Index Terms—Credit Scoring, Domain Driven Classification, Mathematical Programming, Multiple Criteria and Multiple Constraint-level Programming, Fuzzy Programming, Satisfying Solution

1 INTRODUCTION

Due to widespread adoption of electronic funds transfer at point of sale (EFTPOS), internet banking and the near-ubiquitous use of credit cards, banks are able to collect abundant information about card holders' transactions [1]. Extracting effective knowledge from these transaction records and personal data from credit card holder holds enormous profit potential for the banking industry. In particular, it is essential to classify credit card customers precisely in order to provide effective services while avoiding losses due to bankruptcy from users' debt. For example, even a 0.01% increase in early detection of bad accounts can save millions.

Credit scoring is often used to analyze a sample of past customers to differentiate present and future bankrupt and credit customers. Credit scoring can be formally defined as a mathematical model for the quantitative measurement of credit. Credit scoring is a straightforward approach for practitioners. Customers' credit behaviors are modeled by a set of attributes/variables. If a weight is allocated to each attribute, then a sum score for each customer can be generated by timing attribute values with their weights; scores are ranked and the bankrupt customers can be distinguished. Therefore, the key point is to build an accurate attribute weight set from the historical sample and so that the credit scores of customers can be used to predict their future credit behavior. To measure the precision of the credit scoring method, we compare the predicted results with the real behavior of bankrupts by several metrics (introduced later).

Achieving accurate credit scoring is a challenge due to a lack of domain knowledge of banking business. Previous work used linear and logistic regression for credit scoring; we argue that regression cannot provide mathematically meaningful scores and may generate results beyond the user's control. Also, traditional credit analysis, such as decision tree analysis [4] [5], rule-based classification [6][7], Bayesian classifiers [8], nearest neighbor approaches [9], pattern recognition [10], abnormal detection [11][12], optimization [13], hybrid approach [14], neural networks [15] [16] and Support Vector Machine (SVM) [17] [18], are not suitable since they are data-driven mining that cannot directly meet the needs of users [2] [3]. The limitation of the data driven mining approaches is that it is difficult to adjust the rules and parameters during the credit scoring process to meet the users' requirements. For example, unified domain related rules and parameters may be set before the classification commences, but it cannot be controlled since data driven mining often runs automatically.

- *J. He is with School of Engineering and Science, Victoria University, Australia and Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, China. E-mail: jing.he@vu.edu.au*
- *Y. Zhang is with School of Engineering and Science, Victoria University, Australia. E-mail:Yanchun.Zhang@vu.edu.au*
- *Y. Shi is with the Research Center on Fictitious Economy and Data Science, Chinese Academy of Sciences, China and College of Information Science and Technology, University of Nebraska at Omaha, USA. E-mail: yshi@gucas.ac.cn*
- *G. Huang is with School of Engineering and Science, Victoria University, Australia. E-mail:abysshuang@gmail.com*

Our vision is that human interactions can be used to guide more efficient and precise scoring according to the domain knowledge of the banking industry. Linear Programming (LP) is a useful tool for discriminating analysis of a problem given appropriate groups (eg. ”Good” and ”Bad”). In particular, it enables users to play an active part in the modeling and encourages them to participate in the selection of both discriminate criteria and cutoffs (boundaries) as well as the relative penalties for misclassification [19] [20] [21].

We provide a novel domain-driven classification method that takes advantage of multiple criteria and multiple constraint-level programming (MC2) for intelligent credit scoring. In particular, we build the domain knowledge as well as parameters derived from experts' experience into criteria and constraint functions of linear programming, and employ the human and machine conversation to help produce accurate credit scores from the original banking transaction data, and thus make the final mining results simple, meaningful and easily-controlled by users. Experiments on benchmark datasets, real-life datasets and massive datasets validated the effectiveness and efficiency of the proposed method.

Although many popular approaches in the banking industry and government are score-based [22], such as the Behavior Score developed by Fair Isaac Corporation (FICO) in [23], the Credit Bureau Score (also developed by FICO) and the First Data Resource (FDR)'s Proprietary Bankruptcy Score [24], to the best of our knowledge, no previous research has explored domain-driven credit scoring for the production of actionable knowledge for the banking industry. The two major advantages of our domain-driven score-based approach are:

(1) Simple and meaningful. It outperforms traditional data-driven mining because the practitioners do not like black box approach for classification and prediction. For example, after being given the linear/nonlinear parameters for each attribute, the staff of credit card providers are able to understand the meanings of attributes and make sure that all the important characteristics remain in the scoring model to satisfy their needs.

(2) Measurable and actionable. Credit card providers value sensitivity- the ability to measure the true positive rate (that is, the proportion of bankruptcy accounts that are correctly identified- the formal definition will be given later) but not accuracy [25]. Sensitivity is more important than other metrics. For example: a bank's profit on ten credit card users in one year may be only $10 * \$\ 100 = \$\ 1000$ while the loss from one consumer in one day may be $\$\ 5000$. So the aim of a two-class classification method is to separate the ”bankruptcy” accounts (”Bad”) from the ”non-bankruptcy” accounts (”Good”) and to identify as many bankruptcy accounts as possible. This is also known as the method of ”making black list” [26]. This idea decides not only the design way for the classification model but also how to measure the model. However, classic classification methods such as SVM [17] [18], decision tree [27] and neural network [15] that pay more attention to gaining high overall accuracy but not sensitivity cannot satisfy the goal.

Using LP to satisfy the goal of achieving higher sensitivity, the problem is modeled as follows.

In linear discriminate models, the misclassification of data separation can be expressed by two kinds of objectives. One is to minimize the internal distance by minimizing the sum of the deviations (MSD) of the observations from the critical value and the other is to maximize the external distance by maximizing the minimal distances (MMD) of observations from the critical value. Both MSD and MMD can be constructed as linear programming. The compromise solution in Multiple Criteria Linear Programming (MCLP) locates the best trade-off between linear forms of MMD and MSD as data separation.

In the classification procedure, although the boundary of two classes, b (cutoff), is the unrestricted variable in model MSD, MMD and MCLP, it can be pre-set by the analyst according to the structure of a particular database. First, choosing a proper value of b as a starting point can speed up solving the above three models. Second, given a threshold τ, the best data separation can be selected from results determined by different b values. Therefore, the parameter b plays a key role in achieving the desired accuracy and most mathematical programming based classification models use b as an important control parameter [19], [29], [30], [21], [22], [26], [28],[22], [39].

Selection of the b candidate is time consuming but important because it will affect the results significantly. Mathematical programming based classification calls for a flexible method to find a better cutoff b to improve the precision of classification for real-time datasets. For this reason, we convert the selection process of cutoff as the multiple constraints and construct it into the programming model. With the tradeoff between MSD and MMD, the linear programming has been evolved as multiple criteria and multiple constraint-level linear programming. The potential solution in MC2 locates the best trade-off between MMD and MSD with a better cutoff in a given interval instead of a scalar.

However, this trade-off cannot be guaranteed to lead to the best data separation even if the trade-off provides the better solution. For instance, one method with a high accuracy rate may fail to predict a bankruptcy. The decision on the ”best” classification depends powerfully on the preference of the human decision makers. The main research problem addressed by this paper is to seek an alternative method with trade off of MMD and MSD, cutoff constraint and human cooperation for the best possible data separation result: we shall refer to it as the domain driven MC2-based classification method.

Thus, following the idea of L. Cao, C. Zhang and etc. [2] [3], the core of our novel MC2-based method is as follows: we model domain knowledge into criteria and constraint function for programming. Then, by measurement and evaluation, we attain feedback for in-depth modeling. In this step, human interaction is adopted for choosing suitable classification parameters as well as remodeling after monitoring and tracking the process, and also for determining whether the users accept the scoring model. This human machine conversation process will be employed in the classification process. A framework of domain-driven credit scoring and decision making process is shown in Fig. 1.

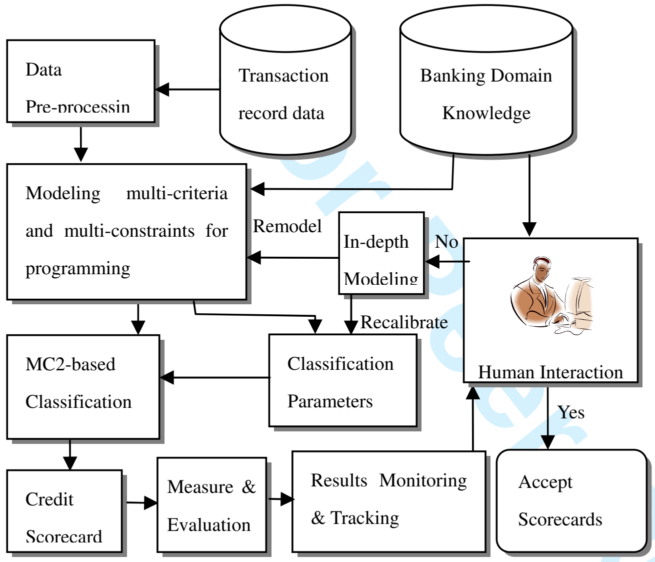


Fig. 1. Domain Driven Credit Scoring and Decision Making Process

An advantage of our domain-driven MC2-based method is that it allows addition of criterion and constraint functions to indicate human intervention. In particular, the multiple criteria for measuring the deviation and the multiple constraints to automatically select the cutoff with a heuristic process to find the user's satisfied solution will be explored to improve the classification model. Another advantage of our MC2 model is that it can be adjusted manually to achieve a better solution, because the score may be not consistent with the user's determination. Instead of overriding the scoreboard directly, the credit manager can redefine the objectives/constraints/upper boundary and the interval of cutoff b to verify and validate the model.

Therefore, in this paper, our first contribution is to provide a flexible process using sensitivity analysis and interval programming according to domain knowledge of banking to find a better cutoff for the general LP based classification problem. To efficiently reflect the human intervention, we combine the two criteria to measure the deviation into one with the interval cutoff and find the users' satisfying solution instead of optimal solution. The heuristic process of MC2 is easily controlled by the banking practitioners. Moveover, the time-consuming process of choosing and testing the cutoff is changed to a automatic checking process. The user only needs to input an interval and the MC2 linear programming can find a best cutoff b for improving precision of classification.

To evaluate precision, there are four popular measurement metrics: accuracy, specificity, sensitivity and K-S (Kolmogorov-Smirnov) value, which will also be used to measure the precision of credit scoring throughout this paper, given as follows. Firstly, we define four concepts. TP (True Positive): The number of Bad records classified correctly. FP (False Positive): The number of Normal records classified as Bad. TN(True Negative): The number of Normal records classified correctly. FN (False Negative): The number of Bad records classified as Normal. In the whole paper, ”Normal” equals to ”Good”. Then, $Accuracy = \frac{TN+TP}{TP+FP+FN+TN}$, $Specificity = \frac{TN}{FP+TN}$,$Sensitivity = \frac{TP}{TP+FN}$. Another metric, the two-sample KS value, is used to measure how far apart the distribution functions of the scores of the ”good” and ”bad” are: the higher values of the metrics, the better the classification methods. Thus, another contribution of this paper is that through example and experiments based on various datasets, we analyze the above four metrics as well as the time and space complexity to prove the proposed algorithm outperforms three counterparts: support vector machine (SVM), decision tree and neural network methods.

The rest of this paper is organized as follows: Section 2 introduces an LP approach to credit scoring based on previous work. Section 3 presents details of our domain-driven classification algorithm based on MC2 for achieving intelligent credit scoring. In Section 4, we evaluate the performance of our proposed algorithm by experiments with various datasets. Our conclusion is provided in Section 5.

2 Linear Programming Approach

2.1 Basic Concepts of LP

A basic framework of two-class problems can be presented as follows: Given $A_iX \leq b$, $A \in B$ and $A_iX \geq b$, $A \in G$, where r is a set of variables (attributes) for a data set $a = (a_1, ..., a_r)$, $A_i = (A_{i1}, ..., A_{ir})$ is the sample of data for the variables, $i = 1, ..., n$ and n is the sample size, the aim is to determine the best coefficients of the variables, denoted by $X = (x_1, ..., x_r)^T$, and a boundary (cutoff) value b (a scalar) to separate two classes: G and B. For credit practice, B means a group of ”bad” customers, G means a group of ”good” customers.

In our previous work [22][26], to accurately measure the separation of G and B, we defined four parameters for the criteria and constraints as follows:

α_i: the overlapping of two-class boundary for case A_i (external measurement);

α: the maximum overlapping of two-class boundary for all cases A_i ($\alpha_i < \alpha$);

β_i: the distance of case A_i from its adjusted boundary (internal measurement);

β: the minimum distance for all cases A_i from its adjusted boundary ($\beta_i > \beta$).

To achieve the separation, the minimal distances of observations from the critical value are maximized (MMD). The second separates the observations by Minimizing the Sum of the Deviations (MSD) of the observations. This deviation is also called "overlapping". A simple version of Freed and Glover's [19] model which seeks MSD can be written as:

$$\begin{aligned} \text{(M1)} \qquad & \text{Min} \quad \Sigma_i \alpha_i, \\ s.t. \quad & A_i X \leq b + \alpha_i, A_i \in B \\ & A_i X > b - \alpha_i, A_i \in G \end{aligned} \tag{1}$$

where A_i, b are given, X is unrestricted and $\alpha_i \geq 0$.

The alternative of the above model is to find MMD:

$$\begin{aligned} \text{(M2)} \qquad & \text{Max} \quad \Sigma_i \beta_i, \\ s.t. \quad & A_i X \geq b - \beta_i, A_i \in B \\ & A_i X < b + \beta_i, A_i \in G \end{aligned} \tag{2}$$

where A_i, b are given, X is unrestricted and $\beta_i \geq 0$.

A graphical representation of these models in terms of α and β is shown in Fig. 2. We note that the key point of the two-class linear classification model is to use a linear combination of the minimization of the sum of α_i or maximization of the sum of β_i. The advantage of this conversion is that it allows easy utilization of all techniques of LP for separation, while the disadvantage is that it may miss the scenario of trade-offs between these two separation criteria [26].

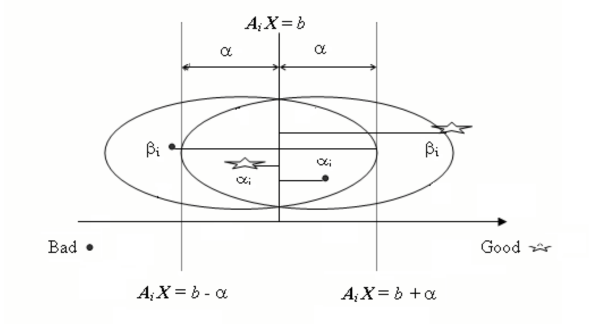


Fig. 2. Overlapping Case in Two-class Separation of LP Model.

A hybrid model (M3) in the format of multiple criteria linear programming (MC) that combines models of (M1) and (M2) is given by [26][22]:

$$\begin{aligned} \text{(M3)} \qquad & \text{Minimize} \quad \Sigma_i \alpha_i, \\ & \text{Maximize} \quad \Sigma_i \beta_i, \\ & \mathbf{A}_i \mathbf{X} = b + \alpha_i - \beta_i, \quad \mathbf{A}_i \in B, \\ & \mathbf{A}_i \mathbf{X} = b - \alpha_i + \beta_i, \quad \mathbf{A}_i \in G \end{aligned} \tag{3}$$

where A_i, b are given, X is unrestricted, $\alpha_i \geq 0$ and $\beta_i \geq 0$.

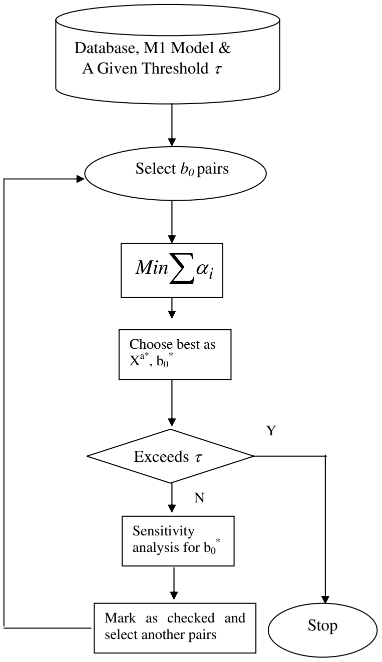

Database, M1 Model &
A Given Threshold τ
Select b_0 pairs
$Min \sum \alpha_i$
Choose best as X^{a*}, b_0^*
Exceeds τ
Y
N
Sensitivity analysis for b_0^*
Mark as checked and select another pairs
Stop

Fig. 3. A Flowchart of Choosing the b candidate for M1.

2.2 Finding the cutoff for Linear Programming

2.2.1 Sensitive analysis

Instead of finding the suitable b randomly, this paper puts forward a sensitivity analysis based approach to find and test b for (M1) shown in Fig. 3. Cutoff b for (M2) can be easily extended in a straightforward way.

First, we select b_0 and $-b_0$, as the original pair for the boundary. After (M1), we choose a better value, b_0^*, between them. In order to avoid duplicate checking for some intervals, we find the changing interval $\triangle b^*$ for b^* to keep the same optimal solution by sensitivity analysis [31]. (Users can see the detailed algorithm and proof in Appendix 1).

In order to reduce computation complexity, when we continue to check the next cutoff for (M1), it is better to avoid $b^* + \triangle b^*$ interval. We label the interval as checked and the optimal solution for b^* as the best classification parameters in the latest iteration. Similarly, we can get the interval for $-b^*$ and it is obvious that if the optimal solution is not the better classification parameters, $-b^* + \triangle(-b)^*$ is labelled as checked, too. After another iteration, we can gain the different optimal results for another $X^{a*'}$ from another pair. Note that the next $X^{a*'}$ may not be better than X^{a*}; if it is worse, we just keep X^{a*} as the original one and choose another pair for b in the next iteration.

Basically, the sensitivity analysis can be viewed as a data separation through the process of selecting the better b. However, it is hard to be sure whether the optimal solution always results in the best data separation. Therefore, the main research problem in credit scoring analysis is to seek a method for producing higher

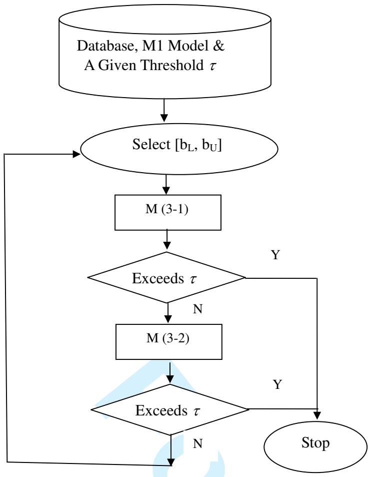


Fig. 4. Interval Programming to Choose better b*.

prediction accuracy (like sensitivity for testing ongoing customers). Such methods may provide a near-optimal result, but lead a better data separation result. That is why we use a Man-Machine Interaction as shown in Fig. 3; this is a typical Human Cooperated Mining in domain driven process.

2.2.2 *Interval programming*

Using the interval coefficients for the right hand side vector b in MSD and MMD can serve the purpose of finding a better cutoff for the classification problem. Interval coefficients include not only interval numbers but also α-cuts, fuzzy numbers [32]. The general solution for interval linear programming can be found in [33]. We assume that the entries of the right hand side vector b are not fixed, they can be any value within prescribed intervals: ${b_i}^l \leq b \leq {b_i}^u$, then the interval MSD (IMSD) can be stated as:

$$\text{(M3-1)} \qquad \text{Min} \quad \Sigma_i \alpha_i,$$
$$s.t. \quad A_i X \leq [b_l, b_u] + \alpha_i, A_i \in B \qquad (4)$$
$$A_i X > [b_l, b_u] - \alpha_i, A_i \in G$$

where A_i, b_l, b_u are given, X is unrestricted and $\alpha_i \geq 0$.

The interval MMD (IMMD) can be stated as:

$$\text{(M3-2)} \qquad \text{Max} \quad \Sigma_i \beta_i,$$
$$s.t. \quad A_i X \geq [b_l, b_u] - \beta_i, A_i \in B \qquad (5)$$
$$A_i X < [b_l, b_u] + \beta_i, A_i \in G$$

where A_i, b_l, b_u are given, X is unrestricted and $\alpha_i \geq 0$.

Similar to Fig.3, IMSD and IMMD process can be stated in Fig. 4 which can be adjusted dynamically to suit user preference.

3 Proposed MC2 Methods

To improve the precision (defined by four metrics in Section 1) of the LP approach, we develop a new approach in this paper. Firstly, we provide Domain Driven MC2-based (DDS-MC2) Classification, which models domain knowledge of banking into multiple criteria and multiple constraint-level programming. Then we develop Domain Driven Fuzzy MC2-based (DDF-MC2) Classification, which is expected to improve the precision of DDS-MC2 further by adding human manual intervention into the credit scoring process and to reduce the computation complexity.

3.1 Domain Driven MC2 based Classification

3.1.1 *MC2 Algorithm for Two-Classes Classification*

Although sensitivity analysis or interval programming can be implemented into the process of finding a better cutoff for linear programming, it still has some inherent shortcomings. Theoretically speaking, classification models like (M1), (M2), (M3-1) and (M3-2) only reflect local properties, not global or entire space of changes. This paper proposes multiple criteria and multiple constraint-level programming to find the true meaning of the cutoff and an unrestricted variable.

A boundary value b (cutoff) is often used to separate two groups (recall (M1)), where b is unrestricted. Efforts to promote the accuracy rate have been confined largely to the unrestricted characteristics of b. A given b is put into calculation to find coefficients X according to the user's experience related to the real time data set. In such a procedure, the goal of finding the optimal solution for the classification question is replaced by the task of testing boundary b. If b is given, we can find a classifier using an optimal solution. The fixed cutoff value causes another problem in that the probability of those cases that can achieve the ideal cutoff score would be zero. Formally, this means that the solutions obtained by linear programming are not invariant under linear transformations of the data. An alternative approach to solve this problem is to add a constant, ζ, to all the values, but it will affect the weight results and performance of its classification, and unfortunately, it cannot be used for the method in [1]. Adding a gap between the two regions may overcome the above problem. However, if the score is falling into this gap, we must determine which class it should belong to [1].

To simplify the problem, we use a linear combination of b^λ to replace b to get the best classifier as $X^*(\lambda)$. Suppose we now have the upper boundary b_u and lower boundary b_l. Instead of finding the best boundary b randomly, we find the best linear combination for the best classifier. That is, in addition to considering the criteria space that contains the tradeoffs of multiple criteria in (M1) and (M2), the structure of MC2 linear programming has a constraint-level space that shows all possible tradeoffs of resource availability levels (i.e. the tradeoff of upper boundary b_u and lower boundary of b_l). We can test the interval value for both b_u and b_l by using a classic interpolation method such as those of Lagrange, Newton, Hermite, and Golden Section in real numbers $[-\infty, +\infty]$. It is not necessary to set negative and positive values for b_l and b_u separately but it is better to set the initial value of b_l as the minimal value and the initial value of b_u as the maximum value. So the internal between $[b_l, b_u]$ is narrowed down.

With the adjusting boundary, MSD (M1) and MMD (M2) can be changed from standard linear programming to linear programming with multiple constraints.

$$\begin{aligned} \text{(M1-1)} \quad & \text{Min} \quad \Sigma_i \alpha_i, \\ s.t. \quad & A_i X \leq \lambda_1 b_l + \lambda_2 b_u + \alpha_i, A_i \in B \\ & A_i X > \lambda_1 b_l + \lambda_2 b_u - \alpha_i, A_i \in G \\ & \lambda_1 + \lambda_2 = 1 \\ & 0 \leq \lambda_1, \lambda_2 \leq 1 \end{aligned} \tag{6}$$

where A_i, b_u, b_l are given, X is unrestricted and $\alpha_i \geq 0$.

$$\begin{aligned} \text{(M2-1)} \quad & \text{Max} \quad \Sigma_i \beta_i, \\ s.t. \quad & A_i X \geq \lambda_1 b_l + \lambda_2 b_u - \beta_i, A_i \in B \\ & A_i X < \lambda_1 b_l + \lambda_2 b_u + \beta_i, A_i \in G, \\ & \lambda_1 + \lambda_2 = 1, \\ & 0 \leq \lambda_1, \lambda_2 \leq 1 \end{aligned} \tag{7}$$

where A_i, b_u, b_l are given, X is unrestricted and $\beta_i \geq 0$.

The above two programmings are LP with multiple constraints. This formulation of the problem always gives a nontrivial solution.

The combination of the above two models is with the objective space showing all tradeoffs of MSD and MMD:

$$\begin{aligned} \text{(M4)} \quad & \text{Min} \quad \Sigma_i \alpha_i, \\ & \text{Max} \quad \Sigma_i \beta_i, \\ s.t. \quad & A_i X = \lambda_1 b_l + \lambda_2 b_r + \alpha_i - \beta_i, A_i \in B \\ & A_i X = \lambda_1 b_l + \lambda_2 b_r - \alpha_i + \beta_i, A_i \in G \\ & \lambda_1 + \lambda_2 = 1, \\ & 0 \leq \lambda_1, \lambda_2 \leq 1 \end{aligned} \tag{8}$$

where A_i, b_u, b_l are given, X is unrestricted, $\alpha_i \geq 0$ and $\beta_i \geq 0$.

A graphical representation of MC2 models in terms of α and β is shown in Fig. 5.

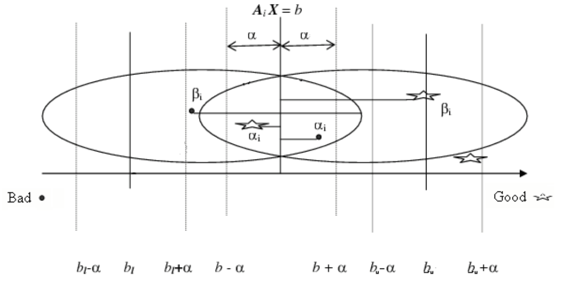


Fig. 5. Overlapping Case in Two-class Separation of MC2 model.

In (M4), theoretically, finding the ideal solution that simultaneously represents the maximal and the minimal is almost impossible. However, the theory of MC linear programming allows us to study the tradeoffs of the criteria space. In this case, the criteria space is a two dimensional plane consisting of MSD and MMD. We use a compromised solution of multiple criteria and multiple constraint-level linear programming to minimize the sum of α_i and maximize the sum of βi simultaneously. Then the model can be rewritten as:

$$\begin{aligned} \text{(M5)} \quad & \text{Max} \quad -\gamma_1 \Sigma_i \alpha_i + \gamma_2 \Sigma_i \beta_i, \\ s.t. \quad & A_i X = \lambda_1 b_l + \lambda_2 b_r + \alpha_i - \beta_i, A_i \in B \\ & A_i X = \lambda_1 b_l + \lambda_2 b_r - \alpha_i + \beta_i, A_i \in G \\ & \gamma_1 + \gamma_2 = 1 \\ & \lambda_1 + \lambda_2 = 1 \\ & 0 \leq \lambda_1, \lambda_2 \leq 1 \\ & 0 \leq \gamma_1, \gamma_2 \leq 1 \end{aligned} \tag{9}$$

where $\mathbf{A}_i$, b_u, b_l is given, $\mathbf{X}$ is unrestricted, $\alpha_i \geq 0$, and $\beta_i \geq 0$, $i = 1, 2, ..., n$. Separating the MSD and MMD, (M4) is simplified to LP problem with multiple constraints like (M3-1) and (M3-2). Replacing the combination of b_l and b_u with the fixed b, (M4) becomes an MC problem like (M3).

This formulation of the problem always gives a nontrivial solution and is invariant under linear transformation of the data. γ_1 and γ_2 are the weight parameters for MSD and MMD. λ_1 and λ_2 are the weight parameters for b_u and b_l, respectively: they serve to normalize the constraint-level and the criteria-level parameters.

A series of theorems on the relationship among the solutions of LP, multiple criteria linear programming (MC) and MC2 are used to prove the correctness of the MC2-based classification algorithm. The detailed proof for the MC2 model can be found in Appendix 2.

3.1.2 Multi-class Classification

(M5) can be extended easily to solve a multi-class problem (M6) shown in Fig. 6.

$$\begin{aligned} \text{(M6)} \quad \text{Max} \quad & -\gamma_1 \Sigma_i \alpha_i + \gamma_2 \Sigma_i \beta_i, \\ \mathbf{A}_i \mathbf{X} &= \lambda_1 b_l + \lambda_2 b_u + \alpha_i - \beta_i, \\ & \mathbf{A}_i \in G_1, \\ \lambda_1^{k-1} b_l^{k-1} + \lambda_2^{k-1} b_u^{k-1} - \alpha_i^{k-1} + \beta_i^{k-1} &= \mathbf{A}_i \mathbf{X} \\ &= \lambda_1^k b_l^k + \lambda_2^k b_u^k - \alpha_i^k + \beta_i^k, \\ & \mathbf{A}_i \in G_k, k = 2, ..., s-1 \\ \mathbf{A}_i \mathbf{X} &= \lambda_1^{s-1} b_l^{s-1} + \lambda_2^{s-1} b_u^{s-1} - \alpha_i^{s-1} + \beta_i^{s-1}, \\ & \mathbf{A}_i \in G_s \\ \lambda_1^{k-1} b_l^{k-1} + \lambda_2^{k-1} b_u^{k-1} + \alpha_i^{k-1} &\leq \lambda_1^k b_l^k + \lambda_2^k b_u^k - \alpha_i^k, \\ & k = 2, ..., s-1, i = 1, ..., n \\ & \gamma_1 + \gamma_2 = 1 \qquad (10) \\ & \lambda_1 + \lambda_2 = 1 \\ & 0 \leq \lambda_1, \lambda_2 \leq 1 \qquad (11) \end{aligned}$$

where $\mathbf{A}_i$, b_u^k, b_l^k is given, $\mathbf{X}$ is unrestricted, $\alpha_i^j \geq 0$, and $\beta_i^j \geq 0$.

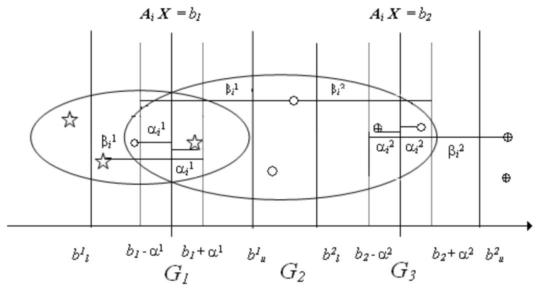


Fig. 6. Three Groups Classified by Using MC2.

3.2 Domain-Driven Fuzzy MC2-based Classification

Instead of finding the ”optimal solution” (a goal value) for the MC2 model like M4, this paper aims for a ”satisfying solution” between upper and lower aspiration levels that can be represented by the upper and lower bounds of acceptability for objective payoffs. A fuzzy MC2 approach by seeking a fuzzy (satisfying) solution obtained from a fuzzy linear program is proposed as an alternative method to identify a compromise solution for MSD (M1-1) and MMD (M2-1) with multiple constraints.

When Fuzzy MC2 programming is adopted to classify the ”good” and ”bad” customers, a fuzzy (satisfying) solution is used to meet a threshold for the accuracy rate of classifications, though the fuzzy solution is a near optimal solution for the best classifier. The advantage of this improvement of fuzzy MC2 includes two aspects: (1) reduced computation complexity. A fuzzy MC2 model can be calculated as linear programming. (2) efficient identification of the weak fuzzy potential solution/special fuzzy potential solution for a better separation. Appendix 3 show that the optimal solution X^* of M11 is a weak/special weak potential solution of (M4).

3.2.1 Fuzzy MC2

To solve the standard MC2 model like (M4), we use the idea of a fuzzy potential solution in [34] and [22]. Then the MC2 model can be converted into the linear format. We solve the following four (two maximum and two minimum) linear programming problems:

$$
\begin{aligned}
\text{(M1-2)} \quad & \text{Min} \quad \text{(Max)} \quad \Sigma_i \alpha_i, \\
s.t. \quad & A_i X \le \lambda_1 b_l + \lambda_2 b_u + \alpha_i, A_i \in B \quad (12)\\
& A_i X > \lambda_1 b_l + \lambda_2 b_u - \alpha_i, A_i \in G \\
& \lambda_1 + \lambda_2 = 1 \\
& 0 \le \lambda_1, \lambda_2 \le 1
\end{aligned}
$$

where A_i, b_u, b_l are given, X is unrestricted and $\alpha_i \ge 0$.

$$
\begin{aligned}
\text{(M2-2)} \quad & \text{Max} \quad \text{(Min)} \quad \Sigma_i \beta_i, \\
s.t. \quad & A_i X \ge \lambda_1 b_l + \lambda_2 b_u - \beta_i, A_i \in B \quad (13)\\
& A_i X < \lambda_1 b_l + \lambda_2 b_u + \beta_i, A_i \in G, \\
& \lambda_1 + \lambda_2 = 1, \\
& 0 \le \lambda_1, \lambda_2 \le 1
\end{aligned}
$$

where A_i, b_u, b_l are given, X is unrestricted and $\beta_i \ge 0$.

Let u_1^0 and l_1^0 be the upper and lower bound for the MSD criterion of (M1-2), respectively. Let u_2^0 and l_2^0 be the upper and lower bound for the MDD criterion of (M2-2), respectively. If x^* can be obtained from solving the following problems (M11), then $x^* \in X$ is a weak potential solution of (M11).

$$
\begin{aligned}
\text{(M11)} \quad & \text{Maximize} \quad \xi, \\
& \xi \le \frac{\Sigma \alpha_i - y_{1L}}{y_{1U} - y_{1L}}, \\
& \xi \le \frac{\Sigma \beta_i - y_{2L}}{y_{2U} - y_{2L}}, \\
& A_i X \le \lambda_1 b_l + \lambda_2 b_u + \alpha_i, A_i \in B \quad (14)\\
& A_i X > \lambda_1 b_l + \lambda_2 b_u - \alpha_i, A_i \in G \\
& A_i X \ge \lambda_1 b_l + \lambda_2 b_u - \beta_i, A_i \in B \\
& A_i X < \lambda_1 b_l + \lambda_2 b_u + \beta_i, A_i \in G, \\
& \lambda_1 + \lambda_2 = 1 \\
& 0 \le \lambda_1, \lambda_2 \le 1
\end{aligned}
$$

where $\mathbf{A}_i$, y_{1L}, y_{1U}, y_{2L}, y_{2U}, b_l, b_u are known, $\mathbf{X}$ is unrestricted, and α_i, β_i, λ_1, λ_2, $\xi \ge 0$, $i = 1, 2, ..., n$.

To find the solution for (M11), we can set a value of δ, and let $\lambda_1 \ge \delta$, $\lambda_2 \ge \delta$. The contents in Appendix 3 can prove that the solution with this constraint is a special fuzzy potential solution for (M11). Therefore, seeking $Maximum \quad \xi$ in the Fuzzy MC2 approach becomes the standard of determining the classifications between ”Good” and ”Bad” records in the database. Once model (M11) has been trained to meet the given threshold τ, the better classifier is identified.

(M11) can be solved as a linear programming and it exactly satisfies the goal of accurately identifying the ”good” and ”bad” customers. The proof of the fuzzy MC2 model can be found in Appendix 3.

3.2.2 Heuristic Classification Algorithm

To run the proposed algorithm below, we first create a data warehouse for credit card analysis. Then we generate a set of relevant attributes from the data warehouse, transform the scales of the data warehouse into the same numerical measurement, determine the two classes of ”good” and ”bad” customers, classify threshold τ that is selected by the user, train and verify the set.

$Algorithm$:credit scoring by using MC2 in Fig. 7.

$Input$: the training samples represented by discrete-valued attributes, the set of candidate attributes, interval of cutoff b, and a given threshold by user.

$Output$:best b^* and parameters X* for credit scoring.

Method:

(3) Give a class boundary value b^u and b^l and use models (M_{1-1}) to learn and computer the overall scores $\mathbf{A}_i\mathbf{X}$ $(i = 1, 2, ..., n)$ of the relevant attributes or dimensions over all observations repeatedly.

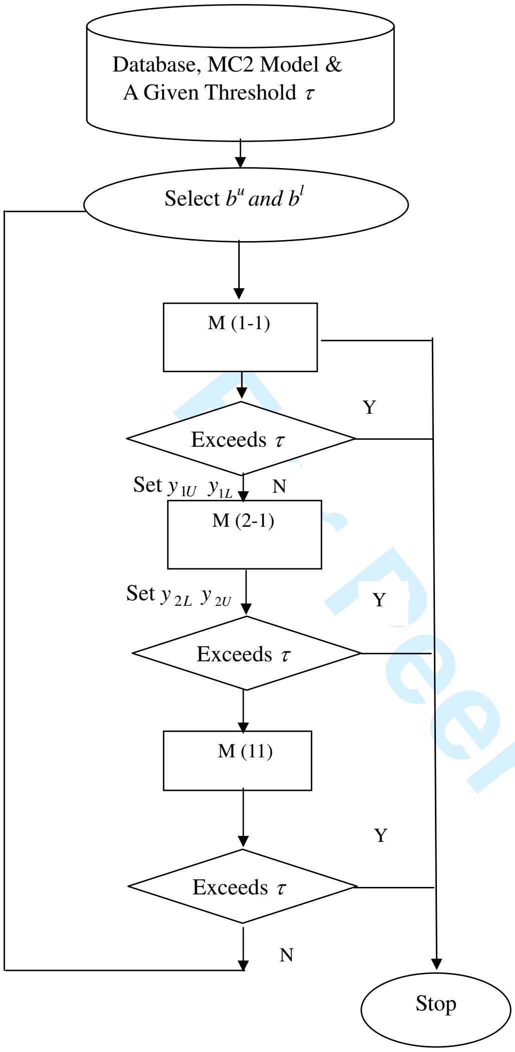


Fig. 7. A Flowchart of MC2 Classification Method.

(4) If (M_{1-1}) exceeds the threshold τ, go to (8), else go to (5).

(5) If (M_{2-1}) exceeds the threshold τ, go to (8), else go to (6).

(6) If (M_{11}) exceeds the threshold τ, go to (8), else go to (3) to consider to give another cut off pair.

(7) Apply the final learned scores $\mathbf{X}^*$ to predict the unknown data in the verifying set.

(8) Find separation.

This approach uses MSD with multiple constraints b as (M1-1) as a tool for the first loop of classification. If the result is not satisfied, then it applies MMD with multiple constraints b as (M2-1) for the second loop of classification. If this also produces an unsatisfied result, it finally produces a fuzzy (satisfying) solution of a fuzzy linear program with multiple constraints, which is constructed from the solutions of MMD and MSD with the interval cutoff, b, for the third loop of classification. The better classifier is developed heuristically from the process of computing the above three categories of classification.

Note that for the purpose of credit scoring, a better classifier must have a higher sensitivity rate and KS value with acceptable accuracy rate. Given a threshold of correct classification as a simple criterion, the better classifier can be found through the training process once the sensitivity rate and KS value of the model exceeds the threshold. Suppose that a threshold is given, with the chosen interval of cutoff b and credit data sets; this domain driven process proposes a heuristic classification method by using the fuzzy MC2 programming (FLP) for intelligent credit scoring. This process has the characteristic in-depth mining, constraints mining, human cooperated mining and loop-closed mining in terms of domain driven data mining.

4 Example and Experimental Results

4.1 An Example

To explain our model clearly, we use a data slice of the classical example in [19]. An applicant is to be classified as a ”poor” or ”fair” credit risk based on responses to two questions appearing on a standard credit application.

For the linear programming based model, users choose the cutoff b randomly. If b falls into the scalar such as (-20, -23), the user can generate an infinite solution set for X^* such as $X = (1, -4)^t$ by the mathematical programming based model like (M1), (M2), (M1-1), (M1-2) and (M11). Then the weighting scheme X^* will be produced to score the 4 customers, denoted by 4 points, and thus they can be appropriately classified by sub-dividing the scores into intervals based on cutoff $b*$ shown in Tab. 1.

Suppose the scalar of b is set as [10,1000] by users; then not all mathematical programming based model like (M2) and (M2-1) can produce the best classification model, and M11 shows its advantage.

TABLE 1
Credit Customer Data Set.

	Credit Customer	Responses		Transformed Scores X=(1, -4)t	Cutoff
		Quest 1 (a1)	Quest 2 (a2)		
Group I (Poor Risk)	1	3	4	-13	b∈ (-20,-23)
	2	4	6	-20	
Group II (Fair Risk)	3	5	7	-23	
	4	6	9	-30	

For (M11), we use a standard software package LINDO [37] to solve the following four linear programming problems instead of using MC2 software.

$$
\begin{aligned}
\text{(M1-2')} \quad \text{Min} \quad \text{(Max)} \quad & \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4, \\
s.t. \quad & A_1 X < \lambda_1 b_l + \lambda_2 b_u + \alpha_1, \\
& A_2 X < \lambda_1 b_l + \lambda_2 b_u + \alpha_2, \\
& A_3 X \geq \lambda_1 b_l + \lambda_2 b_u - \alpha_3, \\
& A_4 X \geq \lambda_1 b_l + \lambda_2 b_u - \alpha_4, \qquad (15)\\
& \lambda_1 + \lambda_2 = 1 \\
& 0 \leq \lambda_1, \lambda_2 \leq 1
\end{aligned}
$$

where A_i is given by Table 1, $b_l = 10$, $b_u = 1000$, X is unrestricted and $\alpha_i \geq 0$.

$$\begin{aligned}
\text{(M2-2}'\text{)} \quad \text{Max} \quad \text{(Min)} \quad & \beta_1 + \beta_2 + \beta_3 + \beta_4, \\
s.t. \quad & A_1X > \lambda_1 b_l + \lambda_2 b_u - \beta_1, \\
& A_2X > \lambda_1 b_l + \lambda_2 b_u - \beta_2, \\
& A_3X \leq \lambda_1 b_l + \lambda_2 b_u + \beta_3, \\
& A_4X \leq \lambda_1 b_l + \lambda_2 b_u + \beta_4, \qquad (16) \\
& \lambda_1 + \lambda_2 = 1 \\
& 0 \leq \lambda_1, \lambda_2 \leq 1
\end{aligned}$$

where A_i is given by Table 1, $b_l = 10$, $b_u = 1000$, X is unrestricted and $\beta_i \geq 0$.

We obtain u_1^0 is 0, $l_1^0 = -1000$, u_2^0 is unbounded and set as 10000, $l_2^0 = 3.33E-03$. The we solve the following fuzzy programming problem:

$$\begin{aligned}
\text{(M11}'\text{)} \qquad \text{Maximize} \quad & \xi, \\
\xi \leq & \frac{\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 + 1000}{1000}, \\
\xi \leq & \frac{\beta_1 + \beta_2 + \beta_3 + \beta_4 - 0.00333}{10000 - 0.00333}, \\
& A_1X < \lambda_1 b_l + \lambda_2 b_u + \alpha_1, \\
& A_2X < \lambda_1 b_l + \lambda_2 b_u + \alpha_2, \\
& A_3X \geq \lambda_1 b_l + \lambda_2 b_u - \alpha_3, \\
& A_4X \geq \lambda_1 b_l + \lambda_2 b_u - \alpha_4, \\
& A_1X > \lambda_1 b_l + \lambda_2 b_u - \beta_1, \\
& A_2X > \lambda_1 b_l + \lambda_2 b_u - \beta_2, \\
& A_3X \leq \lambda_1 b_l + \lambda_2 b_u + \beta_3, \\
& A_4X \leq \lambda_1 b_l + \lambda_2 b_u + \beta_4, \\
& \lambda_1 + \lambda_2 = 1 \\
& 0 \leq \lambda_1, \lambda_2 \leq 1,
\end{aligned}$$

where A_i is given by Table 1, $b_l = 10$, $b_u = 1000$, X is unrestricted, $\alpha_i \geq 0$ and $\beta_i \geq 0$. The optimum solution to the above (M11′) is $X_1 = 6.67$ and $X_2 = -3.33$. The score for the four customers are 6.67, 6.67, 10 and 10 respectively. $b^* = 10$. This is a weak potential solution for the MC2 model like (M4). The two groups have been separated.

4.2 Experimental Evaluation

In this section, we evaluate the performance of our proposed classification methods using various datasets that represent the credit behaviors of people from different countries, shown in Table 2, which includes one benchmark German dataset, two real-world datasets from the UK and US, and one massive dataset randomly generated by a public tool. All these datasets classify people as ”good” or ”bad” credit risks based on a set of attributes/variables.

The first dataset contains a German Credit card records from UCI Machine learning databases [38]. The set contains 1000 records (700 normal and 300 bad) and 24 variables. The second is a UK bankruptcy dataset [1], which collects 307 bad and 918 normal customers with 14 variables. The third dataset is from a major US bank, which contains data for 5000 credit cards (815 bad and 4185 normal) [26][39] with 64 variables that describe cardholders’ behaviors. The fourth dataset includes 500,000 records (305,339 normal and 194,661 bad) with 32 variables is generated by a multivariate Normally Distributed Clustered program (NDC) [40].

Data	Number	Type
German datasets	1,000	Public data
UK datasets	1,225	Real world data
US datasets	5,000	Real world data
NDC massive datasets	500,000	Generated by Public Tool

TABLE 2
Datasets

We also compare our proposed method with three classic methods: support vector machine (SVM light) [41], Decision Tree (C5.0 [27]) and Neural Network. All experiments were carried out on a windows XP P4 3GHZ cpu with 1.48 gigabyte of RAM. Table 3 shows the implementation software that was explored to run the four datasets. We implemented both LP methods (M1, M2) in Section 2 and the proposed DDF-MC2 methods (M1-1, M2-1 and Eq.M11) in Section 3 by using Lindo/Lingo [37] then used Clementine 8.0 to run both C5.0 and neural network algorithms, and at last compiled an open source SVM light in [41] to generate SVM results.

Method	Software	Type
SVMlight	SVMlight	Open source
Decision Tree C5.0	Clementine 8.0	Commercial
Neural Network		
Proposed Methods	Lindo/Lingo	Commercial

TABLE 3
Employed Software.

We now briefly describe our experimental setup. Firstly, we studied both LP methods (M1 and M2) and the proposed DDF-MC2 methods (M1-1, M2-2 and M11) by analyzing real-world datasets and then chose the best one to compare with three counterparts: SVMlight, C5.0 and Neural Network. We validated our proposed best method in massive datasets to study scalability, as well as time efficiency and memory usage. All of the four methods - our proposed method and three counterparts - are evaluated by four metrics (e.g. accuracy, specificity, sensitivity and K-S value) that are defined in Section 1. We used Matlab [42] and SAS [43] to help compute K-S value. Our goal is to see if the best one of our methods performs better than the three counterparts, especially in terms of sensitivity and K-S value, without reducing accuracy and specificity.

4.2.1 Experimental Study of Proposed Methods

We chose real-world datasets from the US and UK to evaluate our proposed methods, because our methods are derived from a real world application project. From both Table 4 (US dataset) and Table 5 (UK dataset), overall, we can see that both the training values and test values of M11 exceed those of the other four methods in terms of sensitivity (except M1) and K-S value. As previously mentioned, these two metrics are the most important ones for helping users in the banking industry to make decisions. We also observe that the accuracy of M11 is slightly lower than for M1 and M1-2, but the specificity and K-S value of M11 are far better than those of M1 and M1-2. Therefore, M11 is the best method.

		Accuracy	Specificity	Sensitivity	KS Value
M1	Training	0.8435	0.0458	0.9989	11.07
	Test	0.8810	0.4768	0.9597	14.45
M2	Training	0.6920	0.2841	0.7714	33.07
	Test	0.5239	0.1447	0.5978	33.03
M1-2	Training	0.8387	0.079	0.9867	8.26
	Test	0.8261	0.0523	0.9768	4.25
M2-2	Training	0.47	0.6413	0.4367	34.87
	Test	0.4283	0.6835	0.3786	34.31
M11	Training	0.7625	0.8601	0.7435	62.34
	Test	0.7976	0.8034	0.7965	60.1

TABLE 4
10-fold cross-validation result of US dataset.

		Accuracy	Specificity	Sensitivity	KS Value
M1	Training	0.8985	0.3943	0.9967	40.1
	Test	0.8808	0.48	0.9589	44.89
M2	Training	0.7277	0.42	0.7876	33.07
	Test	0.5579	0.3356	0.6012	33.03
M1-1	Training	0.9054	0.483	0.9876	47.06
	Test	0.9001	0.4548	0.9868	45.67
M2-1	Training	0.7619	0.8913	0.7367	62.87
	Test	0.704	0.8345	0.6786	51.31
M11	Training	0.8306	0.921	0.813	73.44
	Test	0.819	0.8878	0.8056	69.35

TABLE 5
10-fold cross-validation result of UK dataset.

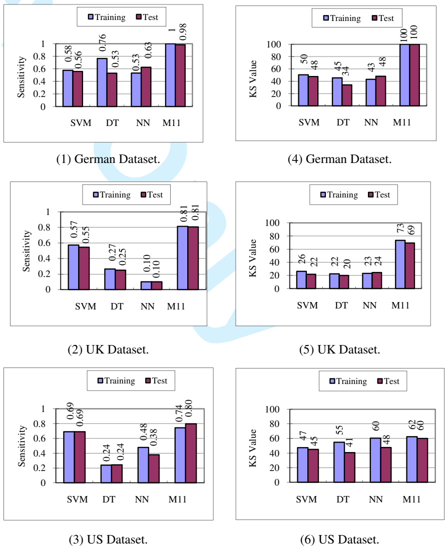
The MC2 method in this paper is the best for predicting "Bad" accounts with a satisfied acceptable sensitivity rate of 0.7965 and the highest KS value (60.1). If the threshold on catching "Bad" accounts is chosen based on a priciple in [2], then the MC2 method should be implemented to conduct the data-mining project. The MC2 method demonstrated its advantages over other mathematical programming methods like M1, M2, M1-1 and M2-1 and is thus an alternative tool to the other well-known classification techniques.

4.2.2 Comparison to Three Counterparts

In this section, the best of our proposed methods, M11, is compared with three classic classification methods: SVMlight (SVM), Decision Tree (DT) and Neural Network (NN). In addition to two real world datasets (US and UK datasets), we adopt a public benchmark dataset (German dataset) to evaluate our method.

We can see from Fig. 8 (1-6), both the sensitivity and KS values of our method M11 excel those of SVMlight, Decision Tree and Neural Network in both training and test datasets of German, UK and US. Notably, the best sensitivity values of M11 for training and test benchmark datasets of German are nearly 1.0 and around 0.98 respectively, which almost double the values that are achieved by other three counterparts in Fig. 8 (1). The same trend is shown in Fig. 8 (4) that the best KS values of M11 are twice others' values, nearly reaching 100. The sensitivity values of the SVM method rank second in all three datasets. The KS values of M11 are trible those of its counterparts in UK dataset. Also, in the US dataset, the KS values of M11 are slightly greater than for other three methods. In a summary, our method M11 outperforms other three methods in terms of sensitivity and KS value: it has better ability to distinguish "bads".

Fig. 8 shows various performances of M11 due to different datasets that have their own distinct properties reflecting the credit behavior of peoples from three countries. These difference also ensure that the evaluation is objective. Overall, the common observation of the results based on three countries' datasets is that the performance of M11 is always better than those of other three counterparts.



(1) German Dataset.

(4) German Dataset.

(2) UK Dataset.

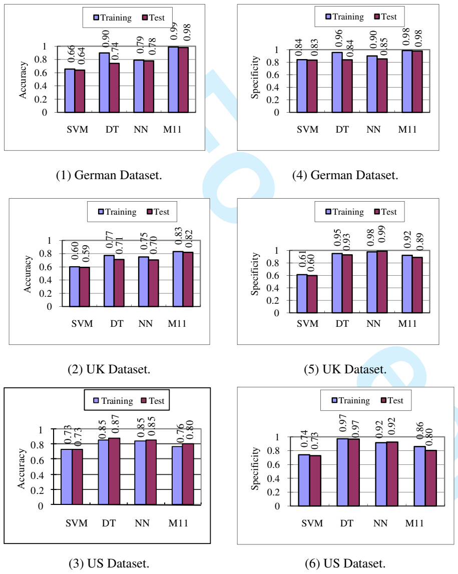
(5) UK Dataset.

(3) US Dataset.

(6) US Dataset.

Fig. 8. Sensitivity and KS Value.

Although accuracy and specificity are not as important as sensitivity and K-S value in the banking industry, we also simply analyze them as follows. Overall, we can see from Fig. 9 (1-6) that the performance of M11 waves slightly above or below the other three methods in terms of both accuracy and sensitivity in three datasets. For example, the accuracy of M11 is slightly greater in the German and UK datasets and slightly lower in the US dataset than the other three methods. Also, the specificity of M11 is slightly greater in the German dataset and slightly less in the UK and US datasets than other three methods.



(1) German Dataset.

(4) German Dataset.

(2) UK Dataset.

(5) UK Dataset.

(3) US Dataset.

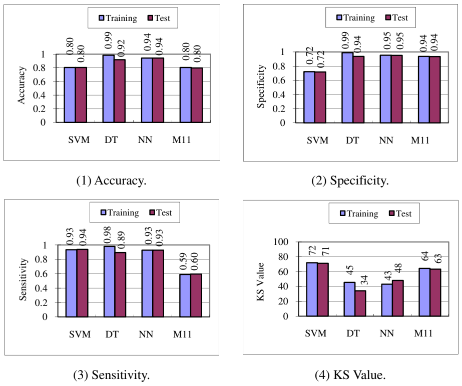
(6) US Dataset.

Fig. 9. Accuracy and Specificity.

4.2.3 *Massive Data Evaluation*

We used NDC in [40] to generate massive datasets as it is difficult to find massive data in real life. According to [48], we explain the NDC as follows: NDC firstly generates a series of random centres for multivariate normal distributions and then randomly generates a fraction of data for each centre. Also, NDC randomly generates a separating plane, based on which classes are chosen for each centre and then randomly generates the points from the distributions. NDC can increase inseparability by increasing variances of distributions. Since the separating plane is generated randomly, we may not control directly the degree of linear separability of the massive data, but this ensures an objective evaluation of our proposed M11. The experimental study of a massive dataset has two goals: the first is to study the performance of our proposed M11 in the four precision metrics; the second is to observe the time efficiency and memory usage of our method.

In the massive dataset, in Fig. 10 (3), M11 performs below the other three methods in terms of sensitivity. However, we can see from Fig. 10 (4) that the KS values of M11 are better than those of Decision Tree and Neural Network. Although it performs best, SVMlight consumes too much time (Table 6). SVMlight took more than 5 hours - nearly 3.6 times of the time spent by M11 -, to process 500,000 records. Thus, the time efficiency of the other three methods including M11 is more reasonable. For transaction banking data, it is not efficient to use SVMlight, because it is very important to predict the credit card holders' behavior in advance, and the sooner we achieve the credit scores the more loss can be avoided. Therefore, from this perspective, the performance of M11 is the best among the actionable methods (the others being decision tree and neural network). We also can observe from Fig. 10 (1-2) that the accuracy and specificity values of M11 are slightly less than those of decision tree and neural network and slightly greater than SVM.



(1) Accuracy.

(2) Specificity.

(3) Sensitivity.

(4) KS Value.

Fig. 10. Massive Data.

Method	Time	Memory
SVM light	18451s	279M
Decision Tree C5.0	1649s	45M
Neural Network	1480s	45M
M11	5098s	67M

TABLE 6
Time and Memory.

Empirical experiments for both Sensitivity rate to predict the bankruptcy and KS value on various datasets showed that the proposed DDF-MC2 classification performed better than decision tree, neural network and SVM with respect to predicting the future spending behavior of credit cardholders.

5 Conclusions

A key difference between our domain-driven MC2-based classification and those that are based on data-driven methods (e.g. SVM, decision tree and neural network) is that in our algorithm, domain knowledge of banking can be built into multiple criteria and multiple constraint functions and also humans can intervene manually to help recalibrate and remodel programming during credit scoring process. This difference ensures that we can achieve a better solution that satisfies the user. Our example and experimental studies based on benchmark datasets and real world datasets, as well as massive datasets, confirm that our proposed method outperforms the abovementioned three data-driven counterparts in terms of sensitivity and KS value while maintaining time efficiency and acceptable accuracy and specificity.

6 ACKNOWLEDGEMENTS

This work was partially supported by an Australian Research Council (ARC) Linkage Project (Integration and Access Services for Smarter, Collaborative and Adaptive Whole of Water Cycle Management) and ARC Discovery Project (Privacy Protection in Distributed Data Mining), as well as the National Natural Science Foundation of China (Grant No.70621001, 70531040, 70840010), 973 Project of Chinese Ministry of Science and Technology (Grant No.2004CB720103), and BHP Billiton.

REFERENCES

[1] Lyn C. Thomas, David B. Edelman, Jonathan N. Crook, Credit Scoring and Its Applications, SIAM, Philadelphia, 2002.

[2] Longbing Cao, Chengqi Zhang: Domain-Driven Actionable Knowledge Discovery in the Real World. PAKDD 2006:821-830.

[3] L. Cao and C. Zhang: Knowledge Actionability: Satisfying Technical and Business Interestingness, International Journal of Business Intelligence and Data Mining 2(4): 496-514 (2007).

[4] J. R. Quilan, Induction of Decision Trees, Machine Learning, 1, 81-106, 1986.

[5] Ron Rymon, An SE-tree Based Characterization of the Induction Problem, Proceedings of the International Conference on Machine Learning, 1993.

[6] D. Martens, M. De Backer, R. Haesen, J. Vanthienen, M. Snoeck and B. Baesens, Classification with Ant Colony Optimization, *IEEE Transactions on Evolutionary Computation*, 11(5):651-665, 2007.

[7] D. Martens, B. B. Baesens and T. Van Gestel, Decomposition Rule Extraction from Support Vector Machines by Active Learning, *IEEE Transactions on Knowledge and Data Engineering*, 21(2): 178-191, 2009.

[8] Domingos Pedro, Michael Pazzani, On the Optimality of the Simple Bayesian Classifier under Zero-one Loss, Machine Learning, 29:103-137, 1997.

[9] Belur V. Dasarathy, editor Nearest Neighbor (NN) Norms: NN Pattern Classification Techniques, ISBN 0-8186-8930-7, 1991.

[10] G. Guo, CR Dyer, Learning from Examples in the Smaple Case: Fase Expression Recognition, *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, 35(3): 477-488, 2005.

[11] Wanpracha Art Chaovalitwong Se, Ya-Ju Fan, Rajesh C. Sachdeo, Support Feature Machine for Classification of Abnormal Brain Activity, *ACM SIGMOD*, 113-122, 2007.

[12] Berger Rachel P., Ta'asan Shlomo, Rand Alex, Lokshin Anna and Kochanek. Patrick, Multiplex Assessment of Serum Biomarker Concentrations in Well-Appearing Children with Inflicted Traumatic Brain Injury, *Pediatric Research*, 65(1): 97-102, 2009.

[13] S. Olafsson, X. Li and S. Wu, Operations Research and Data Mining, *European Journal of Operational Research*, 187(3): 1429-1448, 2008.

[14] A. Benos, G. Papanastasopoulos, Extending the Merton Model: a Hybrid Approach to Assessing Credit Quality, *Mathematical and Computer Modelling*, 2007.

[15] H. Guo and S. B. Gelfand, Classification trees with Neural Network Feature Extraction, IEEE Transaction on Neural Networks, 3, 923-933, 1992.

[16] Lean Yu, Shouyang Wang and Kinkeung Lai, Credit Risk Assessment with a Mutistage Neural Network Ensemble Learning Approach, *Expert Systems with Applications*, 34(2): 1434-1444, 2008.

[17] T. Joachims, Making Large-scale SVM learning practical, In: B. Schol Lkopf, C. Burges, A. Smola (Eds.), Advances in Kernel Methods- Support Vector Learning, MIT-Press, Cambridge, Massachusetts, USA, 1999.

[18] Yongqiao Wang, Shouyang Wang and K. K. Lai, A New Fuzzy Support Vector Machine to Evaluate Cedit Risk, *IEEE Transactions on Fuzzy Systems*, 13(6): 820-831, 2005.

[19] N. Freed, F. Glover, Simple but powerful goal programming models for discriminant problems, European Journal of Operational Research, 7, 44-60, 1981.

[20] Joaquin Pacheco, Silvia Casado and Laura Nunez, Use of VNS and TS in Classification: Variable Selection and Determination of the Linear Discrimination Function Coefficients, *IMA Journal of Management Mathematics*, 18(2): 191-206, 2007.

[21] J. Zhang, Y. Shi and P. Zhang, Several Multi-criteria Programming Methods for Classification, *Computers and Operations Research*, 36(3): 823-836, 2009.

[22] Yong Shi, Multiple Criteria and Multiple Constraint Levels Linear Programming: Concepts, Techniques and Applications, World Scientific, Singapore, 2001.

[23] http://www.fico.com/en/Pages/default.aspx

[24] http://www.firstdata.com/

[25] Han, J., Kamber, M.: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers, 2003.

[26] J. He, X. Liu, Y. Shi, W. Xu, N. Yan: Classifications Of Credit Cardholder Behavior by Using Fuzzy Linear Programming. International Journal of Information Technology and Decision Making, 3(4): 633-650, 2004.

[27] J. Quinlan, See5.0., 2004, available at:http://www.rulequest.com/see5-info.html.

[28] Peng, Y., G. Kou, Y. Shi, and Z. Chen, A Multi-Criteria Convex Quadratic Programming Model for Credit Data Analysis, Decision Support Systems, Vol. 44, 1016-1030, 2008.

[29] N. Freed, F. Glover, Evaluating Alternative Linear Programming Models to Solve the Two-group Discriminant Analysis Formulations, Decision Sciences, 17, 151-162, 1986.

[30] N. Freed, F. Glover, Resolving Certain Difficulties and Improving Classfciation Power of LP Discriminant Analysis Formulations, Decision Sciences, 17, 589-595, 1986.

[31] S. I. Gass, Linear Programming, McGraw-Hill, New York, 1985.

[32] H.J.Zimmermann, Fuzzy Set Theory and its Application, Boston: Kluwer Academic Pub., 1991.

[33] Atanu Sengupta, Tapan Kumar Pal, Debjani Chakraborty, Interpretation of inequality constraints involving interval coefficients and a solution to interval linear programming, Fuzzy Sets and Systems, Vol. 119, pp. 129-138, 2001.

[34] H. J. Zimmermann, Fuzzy Programming and Linear Programming with Several Objective, Fuzzy Sets and Systems, Vol. 13, No. 1, pp. 45-55, 1978.

[35] F. Gyetvan and Y. Shi, Weak Duality Theorem and Complementary Slackness Theorem for Linear Matrix Programming Problems, Operations Research Letters 11 (1992) 249-252.

[36] C. Y. Chianglin, T. Lai and P. Yu, Linear Programming Models with Changeable Parameters - Theoretical Analysis on "Taking Loss at the Ordering Time and Making Profit at the Delivery Time", International Journal of Information Technology & Decision Making, Vol. 6, No. 4 (2007) 577-598.

[37] http://www.lindo.com/

[38] P.M. Murphy, D. W. Aha, UCI repository of machine learning databases, 1992, www.ics.uci.edu/ melearn/MLRepository.html.

[39] Y. Shi, J. He, and etc., Computer-based Algorithms for Multiple Criteria and Multiple Constraint Level Integer Linear Programming, Computers and Mathematics with Applications, 49(5): 903-921, 2005.

[40] D. R. Musicant, NDC: normally distributed clustered datasets, 1998, www.cs.wise.edu/ musicant/data/ndc/.

[41] T. Joachims, SVM-light: Support Vector Machine, 2004, available at http://svmlight.joachims.org/.

[42] http://www.mathworks.com/

[43] SAS, http://www.sas.com.

[44] Hamdy A. Taha: Operations Research: An Introduction, 8th ed., Prentice Hall, ISBN 0-13-188923-0, 2007.

[45] L. Seiford and P. L. Yu, Potential Solutions of Linear Systems: the Multi-criteria Multiple Constriant Level Program, Journal of Mathematical Analysis and Application 69, 283-303, 1979.

[46] J. Philip, Algorithms for the Vector Maximization Problem, Mathematical Programming 2, 207-229, 1972.

[47] P. L. Yu and M. Zeleny, The Set of all Nondominated Solutions in the Linear Cases and a Muticriteria Simplex Method, Journal of Mathematical Analysis and Applications 49, 430-458, 1975.

[48] D. R. Musicant, NDC Description, http://www.cs.wisc.edu /dmi/svm/ndc/content.html.

Appendix

Appendix 1 Sensitivity Analysis for selecting candidate Cutoff b.

This appendix explains the general solution for sensitivity analysis for the right hand side of (M1) and (M2).

Suppose that X^{a*} is identified as the optimal solution of the LP problem (M1); we may be interested in what perturbation of known $\triangle b$ allows X^{a*} to remain optimal. We assume that b is changed to $b^{'} = b^{*} + \triangle b$, then according to the basic idea of the Simplex Method for LP provided by G. B. Dantzig [44], let B be the matrix composed of the basic columns of A^a and Let N be the matrix composed of the nonbasic columns of A^a. The new constraint level vector is $b^{'} = b + \triangle b$, and the new basic variables are $X_B^{a^{'}} = B^{-1}b^{'} = B^{-1}(b + \triangle b)$. Then $X_B^a = B^{-1}b \rightarrow X_B^{a^{'}} = B^{-1}\ (b + \triangle b)$, and $-Z = -C_B^a B^{-1}b \rightarrow -Z^{'} = -C_B^a B^{-1}(b + \triangle b)$. In this case, although the objective coefficient of the optimal solution X^{a*} is independent of the change, the solution X^{a*} may become unfeasible. If $X_B^{a^{'}}$ is feasible, then $B^{-1}b + B^{-1}\triangle b \geq X_B^a + \triangle b B^{-1}$. Therefore, we can determine $\triangle b$ by solving $X_B^a + \triangle b B^{-1} \geq 0$ for $\triangle b$ in the given optimal simplex tableau. Even if $X_B^a + \triangle b B^{-1} < 0$, but the following equation

$$\left[\begin{array}{c} X_B^{'} \\ X_N^{'} \end{array}\right] = \left[\begin{array}{c} B^{-1}(b + \triangle b) \\ 0 \end{array}\right] \tag{17}$$

is still a regular solution [31]. It is easy to use the dual simplex models for iteration until the optimal solutions are found; this is called Sensitivity Analysis [44]. Sensitivity analysis can be implemented to find the cutoff candidate efficiently by using the proposed method shown in Fig. 3.

Appendix 2 Proof for MC2 model of (M4).

This appendix contains the proof that the proposed classification (M4) can find a better solution than MC based (M3), LP based (M1) and (M2) clarification models.

MC2 of (M4) in the matrix format is given as follows:

$$\begin{aligned} \text{(M7)} \qquad \text{Max} \quad & Z = \gamma^t CX, \\ s.t. \quad & AX^a = D\lambda \\ & X \geq 0 \end{aligned} \tag{18}$$

For any given value of (λ_1, λ_2), and (γ_1, γ_2), there is a potential solution of a classification. The structure of potential solutions provides a comprehensive scenario of all possible tradeoffs or compromises between the cutoffs of MSD and MMD in the classification problem.

Using the terminology of Gale, Kuhn and Tucker, the initial formulation of (M4) can be transferred into the following primal-dual pair [45]:

$$\begin{aligned} &\{Max \quad \gamma^t Cx \mid Ax \leq D\lambda, x \geq 0, \lambda \geq 0, \gamma \geq 0\} \\ &\{Min \quad u^t D\lambda \mid u^t A \geq \gamma^t C, u \geq 0, \lambda \geq 0, \gamma \geq 0\} \end{aligned} \tag{19}$$

There is one definition to solve (M4): *Definition* 1: A basis (index set) J is called a potential solution for the MC2 problem if and only if there exist $\lambda > 0$ and $\gamma > 0$ such that J is an optimal basis for

$$\begin{aligned} \text{(M8)} \qquad \text{Max} \quad & Z = (\gamma^0)^t CX, \\ s.t. \quad & AX^a = D\lambda \\ & X \geq 0 \end{aligned} \tag{20}$$

Remark 1.1 If D is replaced by the vector d, then the MC2 problem like (M4) is changed to the MC problem like (M3) . Furthermore, if C is replaced by the vector c, the MC problem becomes the LP problem like (M1) and (M2).

The following theorem shows the relationship between MC and LP programming [46], [47]:

Theorem 1 [35]: The vector x^* is a nondominated solution for (M4) if and only if there is a vector γ^0 for which x^* is an optimal solution for the following scalar LP problem:

$$\{Max(\gamma^0)^t Cx \mid Ax \leq D\lambda^0, x \geq 0\} \tag{21}$$

The following theorem shows the relationship between MC2 and MC programming [46], [47]:

Theorem 2 [35]: The vector x^* is a nonlaminated solution for the MC2 problem (M7), then (x^*, γ^0), $\gamma^0 > 0$, is a nondominated solution for the MC problem. Furthermore, $(x^*, \lambda^0, \gamma^0)$, $\gamma^0 > 0$, $\lambda^0 > 0$ is an optimal solution for the LP problem (theorem 1).

Then theoretically, with the solution relationship among LP, MC, MC2, We can prove the MC2 model of (M4) can find a potential solution better than the MC model (M3) and LP models ((M1) and (M2)).

Appendix 3 Proof for Fuzzy MC2 model.

This appendix contains the proof that the weak potential solution and special weak potential of (M11) is the efficient solution for (M4).

According to [22], a special MC2 problem like (M4) is built as (M9):

$$\begin{aligned} \text{(M9)} \qquad \text{Max} \quad & Z = \Sigma_i \gamma_i C^i x, \\ s.t. \quad & A_i X \leq \Sigma_i \lambda_i b_k, \\ & \Sigma_k \lambda_k = 1 \end{aligned} \tag{22}$$

Let u_i^0 be the upper bound and l_i^0 be the lower bound for the ith criterion $C^i X$ of (M9) if x^* can be obtained from solving the following problem (M10):

$$\begin{aligned} \text{(M10)} \qquad & \text{Maximize} \quad \xi, \\ & \xi \leq \frac{C^i X - l_i^0}{u_i^0 - l_i^0}, i = 1, 2, q \\ & A_i X \leq \Sigma_{i=1}^p \lambda_i b_k, \\ & \Sigma_{k=1}^p \lambda_k = 1, k = 1, 2, ..., p \end{aligned} \tag{23}$$

then $x^* \in X$ is a weak potential solution of (M9) [22][35]. If $\lambda_k \geq \theta > 0$ for given θ, $k = 1, 2, ..., p$ is a special potential solution of (M9). According to this, in formulating a FLP problem, the objectives (Min $\Sigma_{i=1}^n \alpha_i$, Max $\Sigma_{i=1}^n \beta_i$) and constraints ($\mathbf{A}_i \mathbf{X} = b +$

$\alpha_i - \beta_i,\ \mathbf{A}_i \in G, \mathbf{A}_i\mathbf{X} = b - \alpha_i + \beta_i,\ \mathbf{A}_i \in B)$ of (M4) are redefined as fuzzy sets **F** and **X** with corresponding membership functions $\mu_{\mathbf{F}}(x)$ and $\mu_{\mathbf{X}}(x)$, respectively. In this case the fuzzy decision set **D** is defined as $\mathbf{D} = \mathbf{F} \cap \mathbf{X}$, and the membership function is defined as $\mu_{\mathbf{D}}(x) = \{\mu_{\mathbf{F}}(x), \mu_{\mathbf{X}}(x)\}$. In a maximization problem, x_1 is a "better" decision than x_2 if $\mu_{\mathbf{D}}(x_1) \geq \mu_{\mathbf{D}}(x_2)$. Thus, it can be considered appropriate to select x^* such as [26]:

$$\begin{aligned} \max_x \mu_{\mathbf{D}}(x) &= \max_x \min\{\mu_{\mathbf{F}}(x), \mu_{\mathbf{X}}(x)\} \qquad (24) \\ &= \min\{\mu_{\mathbf{F}}(x^*), \mu_{\mathbf{X}}(x^*)\} \end{aligned}$$

where $\max_x \mu_{\mathbf{D}}(x)$ is the maximized solution.

Let y_{1L} be MSD and y_{2U} be MMD, the value of Max $\Sigma_{i=1}^n \alpha_i$ is y_{1U} and the value of Min $\Sigma_{i=1}^n \beta_i$ is y_{2L}. Let $\mathbf{F}_1 = \{x : y_{1L} \leq \Sigma_{i=1}^n \alpha_i \leq y_{1U}\}$ and $\mathbf{F}_2 = \{x : y_{2L} \leq \Sigma_{i=1}^n \beta_i \leq y_{2U}\}$. Their membership functions can be expressed respectively by [26]:

$$\mu_{\mathbf{F}_1}(x) = \begin{cases} 1, & \Sigma_{i=1}^n \alpha_i \geq y_{1U} \\ \dfrac{\Sigma_{i=1}^n \alpha_i - y_{1L}}{y_{1U} - y_{1L}}, & y_{1L} \leq \Sigma_{i=1}^n \alpha_i \leq y_{1U} \\ 0, & \Sigma_{i=1}^n \alpha_i \leq y_{1L} \end{cases}$$

and

$$\mu_{\mathbf{F}_2}(x) = \begin{cases} 1, & \Sigma_{i=1}^n \beta_i \geq y_{2U} \\ \dfrac{\Sigma_{i=1}^n \beta_i - y_{2L}}{y_{2U} - y_{2L}}, & y_{2L} \leq \Sigma_{i=1}^n \beta_i \leq y_{2U} \\ 0, & \Sigma_{i=1}^n \beta_i \leq y_{2L}. \end{cases}$$

Then the fuzzy set of the objective functions is $\mathbf{F} = \mathbf{F}_1 \cap \mathbf{F}_2$ and its membership function is $\mu_{\mathbf{F}}(x) = \min\{\mu_{\mathbf{F}_1}(x), \mu_{\mathbf{F}_2}(x)\}$. Using the crisp constraint set $\mathbf{X} = \{x : \mathbf{A}_i\mathbf{X} = b + \alpha_i - \beta_i, \mathbf{A}_i \in G, \mathbf{A}_i\mathbf{X} = b - \alpha_i + \beta_i, \mathbf{A}_i \in B\}$, the fuzzy set of the decision problem is $\mathbf{D} = \mathbf{F}_1 \cap \mathbf{F}_2 \cap \mathbf{X}$, and its membership function is $\mu_{\mathbf{D}}(x) = \mu_{\mathbf{F}_1 \cap \mathbf{F}_2 \cap \mathbf{X}}(x)$. [34] has shown that the "optimal solution" of

$$\max_x \mu_{\mathbf{D}}(x) = \max_x \min\{\mu_{\mathbf{F}_1}(x), \mu_{\mathbf{F}_2}(x), \mu_{\mathbf{X}}(x)\}$$

is an efficient solution of (M4).

Jing He received her PhD degree from Academy of Mathematics and System Science, Chinese Academy of Sciences, in 2006. She is currently a research fellow at School of Engineering and Science, Victoria University, Australia. Her current research interests include data mining, database systems, and operational research. She has published more than 30 research papers.

Yanchun Zhang is a full Professor and Director of Centre for Applied Informatics at Victoria University. He obtained a PhD degree in Computer Science from the University of Queensland in 1991. He was a research fellow at UQ and CRC Distributed Systems Technology from 1991 to 1993, and an academic member (lecturer, senior lecturer and then associate professor) in the Department of Mathematics and Computing at the University of Southern Queensland from 1994 to June 2003. He has been active in areas of database and information systems, distributed databases, Web and internet technologies, Web information systems, Web data management, Web mining, Web search, Web services, and e-Research. He has published over 200 research papers in refereed international journals and conference proceedings, and authored/edited over a dozen books/proceedings and journal special issues. He is Editor-In-Chief of World Wide Web: Internet and Web Information Systems, the Editor of the Web Information Systems Engineering and Internet Technologies Book Series by Springer, and Chairman of Web Information Systems Engineering (WISE) Society. He is currently a member of the Australian Research Council (ARC) College of Experts.

Yong Shi is a full Professor and the Executive Deputy Director, of Chinese Academy of Sciences Research Center on Fictitious Economy & Data Science since 2007. He has been the Charles W. and Margre H. Durham Distinguished Professor of Information Technology, College of Information Science and Technology, University of Nebraska, USA since 1999. He received his B.S. in mathematics from the Southwestern Petroleum Institute (SWPI), China and a Ph.D in management science from the University of Kansas, USA. His current research interests are data mining, information overload, optimal system designs, multiple criteria decision making, decision support systems, petroleum engineering management, and telecommunication management. He has published 15 books and more than 150 papers. Prof. Shi is the Founding Editor and Editor-in-Chief of the International Journal of Information Technology and Decision Making, an Area Editor of the International Journal of Operations and Quantitative Management, and a member of the Editorial Broad of International Journal of Business Intelligence and Data Mining.

Guangyan Huang received her PhD degree from Institute of Computing Technology, Chinese Academy of Sciences in 2007. She has been an assistant professor at Institute of Software, Chinese Academy of Sciences since April 2007. Her current research interests include data mining, wireless sensor networks and software testing. She has published 15 international conference and journal papers.