Assessment of Customer Credit with Efficient Multivariate Classifier

HUNG-YI LIN¹ & KUANG-YU HUANG²
¹Department of Distribution Management
National Taichung University of Science and Technology
129, Sanmin Rd., Sec. 3, Taichung
Taiwan (R.O.C.)
linhy@nutc.edu.tw

²Department of Information Management
Ling Tung University
1, Ling Tung Rd., Taichung
Taiwan (R.O.C.)
kyhuang@teamail.ltu.edu.tw

Abstract: - The assessment of customer credit is important for financial institutions. Many techniques are developed for customer credit classification. Traditional methods suffer from the problems of inaccurate prediction and/or inefficient data analysis. In this paper, we adopt the entropy-based evaluation method and PCA-based classifier to improve these two problems. A new feature evaluation criterion collaborated with a novel feature selection are proposed to identify the critical factors of classification. By means of multivariate analysis, not only a reduced subset of relevant features is achieved but also the efficient classifier is generated. This paper aims at assessing customer credit with effectiveness and efficiency. From the experimental results of German credit case, it shows that our proposed method can simultaneously improve accuracies in the classes of high-credit and bad-credit when competing with the traditional C4.5 scheme.

Key-Words: - Customer credit assessment, feature selection, multi-class classification, PCA, multivariate analysis, C4.5

1 Introduction
After the subprime mortgage crisis emerged from United States, worldwide enterprises and commercial communities have intensified their efforts to assess customer credits. This is because the inaccurate customer credit assessment may cause financial institutions run into financial difficulty and lead to heavy costs in afterwards management. Customer credit assessing is the important basis for developing the lending strategies and becomes one of the most challenging tasks and a research topic in CRM [21].

The ubiquitous information tools and technology allows us to collect and store a variety of questionnaire and personal data about credit and financial statuses with ease. The ripe development of multi-dimensional database management, artificial intelligence, expert system, and data mining techniques make easy implementation of various investigation, studies, or analyses on the stored data. Decision makers have gained more momentum to tackle difficult problems due to modern computer technologies and computational techniques.

Classification is a problem frequently encountered when a categorical dependent variable needs to be predicted according to a small subset of independent variables. Many classification problems including web page classification [22], web spam detection [3, 6], intrusion detection [7], mobile commerce behavior [19], fraud detection [5], bankruptcy prediction [24], medical diagnosis [9, 13], and crime activity analysis [11], have attracted many research attentions. Recent studies [12, 20, 28] have developed multivariate classification methods to improve accuracy or promote classification performance. We note that high performance is especially important since modern data amount is increasingly growing with a high speed.

Complexity, efficiency, and accuracy are three principles in appraising classifiers. The complexity of one classifier relies on the amount of involved variables and the processing of data analyses. Generally, classifier complexity is closely related to classification efficiency. In other words, one classifier with simple handle can complete the classification task with high efficiency. However, it is a general case that losing classification efficiency in return can gain classification accuracy and vice
versa. This trade-off is a well-known challenge in classification problems. There is no doubt that the approach to simultaneously improving efficiency and accuracy is highly expected.

On the one hand, in order to collect sufficient information, every instance usually includes a lot of categorical and/or numerical variables (i.e., features or factors). These variables generally reflect independent data elements. For clustering problem, all variables are considered independently. For classification problem, one or few variables are taken as dependent for prediction purpose. On the other hand, in order to induce data trend and then deduce future data, it is also required to have an enough amount of instances for analyses. In this scenario, overabundance becomes the bottleneck when filtering and condensing the great amount of data. Note that feature selection and typical instance recognition are two essential issues to remedy such problems. We only focus on the first issue in this paper.

Feature selection aims at exploring the effective variables in data mining process. Many statistical and artificial intelligence techniques are devoted to identify a set of discriminative features. These techniques include genetic algorithm [8, 27], support vector machine [4, 26], neural network and fuzzy [18, 23, 25], logistic regression [10, 15], and principal component analysis [16]. The objective of feature selection is threefold. First is to eliminate the redundant information so that the analytical time of mining process can be reduced. Second, the selection of a small subset of low correlated features will facilitate data mining process since it prevents similar factors from being repeatedly involved. Thirdly, for classification problems, the relevant features to the target feature are more effective than the irrelevant ones when connecting their relationships.

In order to explore the useful data attributes in assessing customer credit, an entropy-based criterion is proposed in evaluating all input variables in this paper. This criterion takes the data distribution and data variation of the target feature into consideration. This criterion is suitable to evaluate categorical and numerical input features. After a small subset of discriminative features is identified, a multivariate classifier is then generated according to the statistical tool of principal component analysis. Customers with different attribute values can be efficiently classified into several distinct credit levels.

This paper is organized as follows. Section 2 first proposes the new learning model and then addresses the novelties in this model. Section 3 explains the issues for multiple classes and data granularity. The detail procedures for generating the multivariate classifier are given in Section 4. Section 5 describes the real case of German credit and the experimental results are presented as well. The conclusion is provided in Section 6.

2 New Inductive Learning Model

Figure 1 shows a new inductive learning model constituted with three stages: 1) feature selection, 2) multivariate analysis, and 3) class matching. After a collection of training data is inputted to the first stage, data preprocessing such as standardizing, z-scoring, or normalizing is first applied for overcoming the problem attributed to different measure scales. Then, all features are evaluated by using the processed data in classifying the target classification variable. All evaluation values are sorted and the features with high ranks are selected for the subsequent handles. Principal component

![Fig. 1 Inductive learning model.](image-url)
The main concept of Shannon’s information theory quantifies the uncertainty involved in predicting the value of a random variable. Information theory is developed from probability theory and statistics. The most important quantities of information are entropy, which measures the amount of uncertainty associated with a discrete random variable. Information Gain is an entropy-based way of comparing two distributions in a manner that assumes \( q(X) \) to be the distribution underlying some data and \( p(X) \) to be the correct distribution. It is thus defined as:

\[
IG(p(X)|q(X)) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)}.
\]

Information Gain Ratio adds the factor of population information amount. These entropy-based criteria all focus on evaluating the random variables with two classes (i.e., yes vs. no, or high vs. low, or good vs. bad). Although the analyses of data distribution (i.e., the statistical dispersion of data) can help distinguish the quality of classification effect, classification problems involving multiple classes tend to have large data variation inside every subset. The following example is used to illustrate this concern.

As shown in Fig. 2(a), a small dataset has two independent variables \((x, y)\) and four classes in its target classification variable \(c\). After classifying all instances using \(x\) and \(y\), the resulting subsets are depicted in Figs. 2(b) and 2(c), respectively. Intuitively, we vote \(x\) as the better feature than \(y\) since the entropy gained from \(x\) (i.e., 0.92) is better than that from \(y\) (i.e., 1). However, the average data variance of \(S_1\) and \(S_2\) (i.e., 0.89) is greater than that of \(S_3\) and \(S_4\) (i.e., 0.25). Or, the data distribution in \(S_1\) has the probabilities of 0.67 \((c = 1)\) and 0.33 \((c = 3)\), which have higher consistency than that in \(S_3\), the probabilities of 0.5 \((c = 1)\) and 0.5 \((c = 2)\). The same condition happens to \(S_2\) and \(S_4\). However, data
variation in $S_1$ (and $S_2$) is greater than that in $S_3$ (and $S_4$). As is well known, feature evaluation methods using the criterion of data distribution concern only the impurity inside each subset. Multiple classes make proximity another concern, which is ignored by many conventional evaluation criteria. Criteria such as entropy-based methods and Gini index are insufficient to evaluate the features in multi-class problems. Our new evaluation criteria will give consideration to both impurity and proximity. The detailed design is given in Section 4.

Data granularity refers to the fineness with which data fields can be sub-divided. For example, census data can be recorded, with low granularity or coarse-grained description, as a single statement: [one has 6 children at the age of 45 and his education level and standard of living are high], or with high granularity or fine-grained description, as multiple statements:

1. [Age = 45]
2. [Number of children = 6]
3. [Education level = high]
4. [Standard of living = high]

Conventional decision trees adopt the concept of fine granularity in every single decision node, which classifies instances level by level. The initial classification triggered by the root node determines the subsequent handling. An adequate high-level decision can simplify the classification tasks assigned at low levels. In the case of multi-class classification, every single variable pays close attention to its own data dimension. It is difficult to vote the most adequate variable and locate it in the root node for generating the optimal classification effect for all subsequent handling. As a result, different types of decision trees can build different classification models and achieve different effectiveness. In fact, the exploration of an optimal decision tree for multi-class problem is a NP-complete problem.

Now that a thoughtful classification is usually accomplished by the cooperation of a collection of relevant variables, data classification according to coarse granularity does not have a bias in favour of one single data dimension but simultaneously takes multiple data dimensions into account. That is, instead of applying univariate classification, the learning model with multivariate classifier can take more aspects into account and in turn achieve better classification quality.

### 4 Feature Selection Scheme

To enhance the classification performance, PCA is integrated into our proposed learning model. PCA is also referred to as Karhunen-Loève transform [14]. This transform is able to reproduce the total system variability and achieves high reduction in dimensionality with usually lower noise than the original patterns. The disadvantage of PCA is that the principle components are not easy to interpret. However, the gain of analytical performance is derived from this disadvantage. The mathematical background of PCA is explained as follows.

Suppose there are $p$ input variables in a dataset. And, this dataset consists $n$ measurements on these $p$ variables. In general, often much of the total system variability can be accounted for by a small number $k$ of the principal components. Namely, there is as much information in the $k$ components as there is in the original $p$ variables. The $k$ principal components can then replace the initial $p$ variables and the original data set.

Principal components are particular linear combinations of a set of random variables (respectively denoted as $x_1, x_2, \ldots, x_p$). Principal components depend solely on the covariance matrix $\Sigma$ of these random variables. Their developments do not require a multivariate normal assumption. Let the eigenvalues of the covariance matrix $\Sigma$ are $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p \geq 0$. The $i$-th component is the linear combination which is given by $y_i = a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{ip}x_p$, where $[a_{i1}, a_{i2}, \ldots, a_{ip}]$ is the eigenvector corresponding to $\lambda_i$, $i = 1, 2, \ldots, p$. Consequently, the proportion of total variance (i.e., explained variance) due to the $i$-th principal component is $\frac{\lambda_i}{\sum_{j=1}^{p} \lambda_j}$. The correlation coefficient between the component $y_i$ and the variable $x_k$ is $\rho_{i,k} = \frac{a_k \sqrt{\lambda_i}}{\sigma_i}$, where $i, k = 1, 2, \ldots, p$. If most (for instance, more than 70%) of the total population variance can be featured to the first or two components, then these components can replace the original $p$ variables without much loss of information. In this paper, only the first principal component with maximum variance is considered as an effective use to replace the original $p$ features.

Data distribution and data variation inside every class are measured to evaluate all input features. Entropy measure and statistical variance collaborate to identify classification effect. Based this design, a modest number $m$ of input features are sifted from
all. The detailed steps in determining the value of $m$ are shown in Fig. 3.

1. Evaluate all input features and rank them increasingly.

2. Calculate the communality of the first feature.

3. Take the next feature into account and recalculate the accumulated communality.

4. Judge whether the stopping criterion is met.

5. Use the latest set of features to generate the multivariate classifier.

Fig. 3 Feature selection procedure.

**Step 1. Evaluate all input features and rank them increasingly.**

As mentioned above, not all input features are useful and effective for the facilitation of classification. Many of them are clumsy in generalizing the categorization from a collection of data. A revised entropy-based evaluation method called intra-diversity is proposed as follows.

Suppose a feature $x$ with $n$ possible values (numeric, nominal, or cardinal) is employed to classify a training set $T$. Then, $T$ is subdivided into $n$ subclasses, respectively denoted as $T_1$, $T_2$, ..., $T_n$. The intra-diversity $ID(x; T)$ is formulated as

$$ID(x; T) = \sum_{i=1}^{n} \frac{\text{Ent}(T_i) \times \text{Var}(T_i)}{|T_i|}$$

(1)

, where Shannon entropy ($\text{Ent}(T_i)$) considers data distribution and variance ($\text{Var}(T_i)$) takes data variation into account. They collaborate to evaluate the data aggregation acquired in every subclass. In order to reduce the effect of the subclass size, the product is weighted by $|T_i|^{-1}$. The smaller estimation of $ID(x; T)$ indicates the less data diversity and in turn the more data aggregation inside every subclass.

Suppose there are $p$ input features ($x_1, x_2, \ldots, x_p$) in the database. After the $ID$ of all input features are estimated, they are sorted into a new list based on an increasing order ($x^{(1)}, x^{(2)}, \ldots, x^{(p)}$). The ones ranked on the front side of the list are more effective than those ranked on the rear side. Hence, a modest number $m$ of features are taken from the front side of the list as the representatives for further processing. We suggest $m$ to be $(\lfloor \sqrt{p} \rfloor + 1)$ in this paper.

**Step 2. Calculate the communality of the first feature.**

After applying PCA on the $m$ features extracted from Step 1, the correlation coefficient between component $y_1$ and variable $x^{(i)}$ is

$$\rho_{y_1,x^{(i)}} = \frac{a_{ik}\sqrt{\Lambda_j}}{\sigma_k}, \text{ where } k \in [1,2,\ldots,m]$$

. The communality of $x^{(1)}$ for component $y_1$ stands for the proportion of variance of $x^{(1)}$ that is due to the component $y_1$. It is measured as $\rho_{y_1,x^{(1)}} = \frac{a_{1k}^2\Lambda_1}{\sigma_k^2}$.

**Step 3. Take the next feature into account and recalculate the accumulated communality.**

The communality accumulated from $x^{(1)}$ and $x^{(2)}$ for component $y_1$ is evaluated as $A_1 = \rho_{y_1,x^{(1)}}^2 + \rho_{y_1,x^{(2)}}^2$.

Generally, the communality accumulated from $x^{(1)}$ to $x^{(k)}$ for the component $y_1$ is formulated as $A_k = \sum_{j=1}^{k} \rho_{y_1,x^{(j)}}^2$, $k \in [2,3,\ldots,m]$.

**Step 4. Judge whether the stopping criterion is met.**

In case of $k$ features are considered, PCA is applied again on these $k$ features and then the proportion of total variance accounted for by the first component is evaluated as $B_k = \frac{\Lambda_1}{\sum_{i=1}^{k} \Lambda_i}$, where $\Lambda_i$ is the $i$-th eigenvalue of the covariance matrix $\Sigma$ formed by these $k$ features. The product $A_k \times B_k$ approximates the absolute communality of the $k$ features (denoted as $C_k$) for the first component. Steps 3 and 4 will execute repeatedly if the sequence of absolute communalities appears an increasing trend. Otherwise, the stopping criterion is met.

**Step 5. Use the latest subset of features to generate the multivariate classifier.**
In this step, the first component obtained from the latest subset of features is outputted as the multivariate classifier of our inductive learning model.

5 Case Study

5.1 Dataset and Feature Evaluation

The data were taken from the German credit dataset preserved in the UCI repository [1]. There are 21 features ($a_1$–$a_{21}$) and 1000 instances in the dataset. These features are classified into three major categories: “Basic data”, “Credit & financial grading”, and “Job related rating”. In the category of “Basic data”, personal status & sex, present residence in years, telephone, age in years, housing status, purpose of loan, and other installment plans were included (as shown in Table 1). Personal status & sex, telephone, purpose of loan, and other installment plans were cardinal measurements with nominal scales. Present residence and age in years were measured with numerical scales, and housing status with an ordinal scale (1: rent, 2: own, 3: for free). In the category of “Credit & financial grading”, except $a_{18}$, most features were measured with ordinal or numerical scales. In the category of “Job related rating”, three features were measured with ordinal or cardinal scales.

To reduce computational complexity, numerical features are converted into discretized or nominal attributes. For comparison studies, the German credit dataset of preprocessed instances are saved into individual files so that they can be reused for building distinct decision models. All decision models were implemented in C and Matlab programming languages executed on a workstation with an Intel Core 2 dual 2.4 GHz processor. For verifying the accuracy of our classifier, 10-fold cross-validation was applied in the dataset. Namely, for each round of learning process, 9 folds (900 instances) were picked as the training data and the left (100 instances) were used as the test data. The training data were classified into four groups: High-credit, Good-credit, Normal-credit, and Bad-credit. Table 2 shows their definitions based on the features of $a_8$ and $a_{18}$. The data corresponding to $a_8$ were so objective that they are obtained from applicants’ account. However, the data corresponding to $a_{18}$ were more subjective since they were judged by the bank.

### Table 1. Description of German credit dataset.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Scale</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic data</td>
<td></td>
</tr>
</tbody>
</table>

### Table 2. Four classes in German credit dataset.

<table>
<thead>
<tr>
<th>Feature</th>
<th>$a_{18} = 1$ (good)</th>
<th>$a_{18} = 2$ (bad)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_8 = 1$ or $2$ (higher or normal amount)</td>
<td>High-credit</td>
<td>Normal-credit</td>
</tr>
<tr>
<td>$a_8 = 3$ or $4$ (lower amount or none)</td>
<td>Good-credit</td>
<td>Bad-credit</td>
</tr>
</tbody>
</table>

The scales and ranges of the remaining features varied from case to case. Take $a_1$ as an example. $a_1 = 1$ denoted male who is divorced or separated. $a_1 = 2$ denoted female who is divorced, separated, or married. $a_1 = 3$ denoted single male while $a_1 = 4$ denoted male who is married or widowed. $a_1 = 5$ denoted single female. To measure the intra-diversity $ID(a_1, T)$ for the remaining features, the formula presented at the step 1 of Section 3 is applied 19 times. Table 3 displays the first six ($= \sqrt{21} + 1$) features with the better $ID$ measurements. The first four features belong to the “Credit & financial grading” category. As these measurements show, feature $a_{16}$ is identified as the most representative variable of all and it truly makes sense that the number of people being liable to provide maintenance (guarantee) for the applicant is significant in assessing his credit...
status. Property ownership status (i.e., $a_{17}$) and housing status (i.e., $a_{6}$) provide the real estate related information and are also identified as the import factors in assessing one’s credit status. Installment rate in percentage of disposable income (i.e. $a_{14}$) has the ID measurement similar to $a_{17}$ and $a_{5}$. Features $a_{2}$ and $a_{19}$ are respectively numerical and ordinal variables whose ID measurements are significantly worst than the first four variables. We note that no cardinal feature is taken in the initial round of feature selection.

<table>
<thead>
<tr>
<th>Rank</th>
<th>Feature</th>
<th>ID($x, T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$a_{16}$</td>
<td>17.55</td>
</tr>
<tr>
<td>2</td>
<td>$a_{17}$</td>
<td>45.50</td>
</tr>
<tr>
<td>3</td>
<td>$a_{8}$</td>
<td>46.30</td>
</tr>
<tr>
<td>4</td>
<td>$a_{14}$</td>
<td>46.75</td>
</tr>
<tr>
<td>5</td>
<td>$a_{5}$</td>
<td>56.81</td>
</tr>
<tr>
<td>6</td>
<td>$a_{19}$</td>
<td>80.73</td>
</tr>
</tbody>
</table>

5.2 Multivariate Classifier

After a small subset of effective features is ensured, the multivariate analysis was applied again to synthesize them. Suppose $n$ features $a^{(1)}, a^{(2)}, \ldots, a^{(n)}$ are inputted to the process of PCA. And then the first component $y_{1}^{(n)}$ is outputted as the multivariate classifier. Alternatively and precisely, $y_{1}^{(n)}$ is denoted as $PCA[a^{(1)}, a^{(2)}, \ldots, a^{(n)}]$. Proceeding with the German credit data, $a^{(1)}=a_{16}, a^{(2)}=a_{17}, a^{(3)}=a_{8}, a^{(4)}=a_{14}, a^{(5)}=a_{2}, a^{(6)}=a_{19}$ forms the base for the subsequent processing. Intuitively, $y_{1}^{(1)}=a_{16}, y_{1}^{(2)}=PCA[a_{16}, a_{17}], y_{1}^{(3)}=PCA[a_{16}, a_{17}, a_{8}], y_{1}^{(4)}=PCA[a_{16}, a_{17}, a_{8}, a_{14}], y_{1}^{(5)}=PCA[a_{16}, a_{17}, a_{8}, a_{14}, a_{2}], y_{1}^{(6)}=PCA[a_{16}, a_{17}, a_{8}, a_{14}, a_{2}, a_{19}]$, and so on. Table 4 reveals the linear combination of variables and intra-diversity for each classifier. Notably, the absolute communality $C_k$ ’s keep a growing trend from $y_{1}^{(1)}$ to $y_{1}^{(5)}$ and thus $y_{1}^{(3)}=0.242a_{16} - 0.667a_{5} - 0.705a_{8}$ is concluded as the optimal classifier for the German credit data. The measurements of intra-diversity are also listed in the last column of Table 4 and they verify that $y_{1}^{(3)}$ owns the best evaluation of all. Since no cardinal feature is included in the classifier, the quantitative property can be completely reserved and then the computational results are meaningful.

5.3 Results

In this study, all learning processes were implemented in Matlab programming languages executing on a workstation with Pentium Dual-core CPU 2.00GHz processor. Besides, Weka 3.6 [2] data mining software in Java was also used to generate the related results about C4.5 decision tree for competitive studies. Our proposed model and C4.5 decision tree were trained 10 times for generating 10 results and then they were averaged to produce a single estimation.

To explore whether our model would support effective classification, analysis of variance (ANOVA) is applied to test if there was any significant difference between four classes. Table 5 shows that $y_{1}^{(2)}$ and $y_{1}^{(3)}$ have the better $F$-test and $p$ values than others. In the last column of Table 5, the accumulated communalities measured from $y_{1}^{(3)}$ to $y_{1}^{(6)}$ indicate that they embrace sufficient population variances for classification task. However, for the sake of performance, $y_{1}^{(3)}$ is taken as the final classifier. So, statistical and quantitative analyses both support that $y_{1}^{(3)}$ is a fine and sound classifier in this experiment.

As to classification accuracy, two decision trees were built by C4.5 algorithm in Weka. In order to carry out a fair contest, the first decision tree only elected three features based on entropy and then constructs its classifier merely based on these features. Features $a_{6}, a_{9}$ and $a_{11}$ were employed in the first decision tree. We denoted this model as DT($a_{6}, a_{9}, a_{11}$). The second decision tree, denoted as DT(all), was built based on the scenario that all 19 features can be used. In a close look into test instances, the accuracies measured from our model and C4.5 decision tree in four classes (High-credit, Good-credit, Normal-credit, and Bad-credit) were listed in Table 6.

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Linear combination of variables</th>
<th>$C_{s}$</th>
<th>ID($y, T$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$y_{1}^{(1)}$</td>
<td>$a_{16}$</td>
<td>0.0517</td>
<td>47.55</td>
</tr>
<tr>
<td>$y_{1}^{(2)}$</td>
<td>$-0.707a_{16} + 0.707a_{17}$</td>
<td>0.2503</td>
<td>45.50</td>
</tr>
<tr>
<td>$y_{1}^{(3)}$</td>
<td>$0.242a_{16} + 0.667a_{17} + 0.705a_{8}$</td>
<td>0.4059</td>
<td>40.42</td>
</tr>
<tr>
<td>$y_{1}^{(4)}$</td>
<td>$0.158a_{16} - 0.652a_{17} - 0.686a_{5} - 0.281a_{14}$</td>
<td>0.3549</td>
<td>45.50</td>
</tr>
<tr>
<td>$y_{1}^{(5)}$</td>
<td>$-0.154a_{16} + 0.657a_{17} + 0.642a_{18} + 0.286a_{14} + 0.225a_{8}$</td>
<td>0.3448</td>
<td>46.35</td>
</tr>
<tr>
<td>$y_{1}^{(6)}$</td>
<td>$0.182a_{16} - 0.534a_{17} - 0.503a_{5} - 0.282a_{14} - 0.357a_{2} + 0.472a_{19}$</td>
<td>0.4065</td>
<td>60.20</td>
</tr>
</tbody>
</table>
The test set contains 100 instances for accuracy analysis. In the high-credit class with the largest data amount (39.5 instances in average for 10-fold cross-validation), the accuracy of our classifier is better than that of DT(all) and is nearly as good as that of DT($a_6, a_9, a_{11}$). Note that there seems to have the losing accuracy on the bad-credit side in return for gaining accuracy on the high-credit side or vice versa for two decision tree models. High accuracy in identifying high-credit customer is so critical that their commercial contributions always dominate bank profits. Nevertheless, approving a potentially bad-credit customer is more costly than denying a good-credit customer. Our classifier successfully gains a better improvement in this respect. The customers accurately identified in the Good-credit class by the three models are respectively as 0.269×28.7(=8), 0.086×28.7(=2), and 0.414×28.7(=12). For the normal-credit class, no customer is classified by three models. As to the bad-credit class, around 10, 7, and 10 customers are respectively identified by three models. The average amounts of instances corresponding to each class are listed in the last column of Table 6.

Finally, the amounts of data entries involved in the training processes of three models are counted for efficiency analysis. For each try in the 10-fold cross-validation experiment, 19 features and 900 instances have to be taken in feature evaluation. Namely, at least 19×900=17100 data entries must be involved for each model. Besides the root node, C4.5 algorithm executes several feature evaluation rounds in order to select the proper features for thesequent decision nodes. This is why Table 7 shows our PCA-based model imposes on less data than C4.5. Table 7 verifies that time efficiency is successfully achieved by our model.

### Table 5. Statistical and quantitative analyses.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>$F$-test</th>
<th>$p$ value</th>
<th>Accumulated communality</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x^{(n)}_i$</td>
<td>4.97</td>
<td>0.002</td>
<td>0.4967</td>
</tr>
<tr>
<td>$x^{(n)}_i$</td>
<td>4.97</td>
<td>0.002</td>
<td>0.8912</td>
</tr>
<tr>
<td>$x^{(n)}_i$</td>
<td>2.64</td>
<td>0.049</td>
<td>1.0153</td>
</tr>
<tr>
<td>$x^{(n)}_i$</td>
<td>1.69</td>
<td>0.168</td>
<td>1.2144</td>
</tr>
<tr>
<td>$x^{(n)}_i$</td>
<td>0.95</td>
<td>0.417</td>
<td>1.5618</td>
</tr>
</tbody>
</table>

### Table 6. The accuracies of the four classes.

<table>
<thead>
<tr>
<th>Class</th>
<th>Accuracy</th>
<th>Size of class</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PCA-based</td>
<td>DT($a_6, a_9, a_{11}$)</td>
</tr>
<tr>
<td>High-credit</td>
<td>0.844</td>
<td>0.896</td>
</tr>
<tr>
<td>Good-credit</td>
<td>0.269</td>
<td>0.086</td>
</tr>
<tr>
<td>Normal-credit</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Bad-credit</td>
<td>0.352</td>
<td>0.241</td>
</tr>
<tr>
<td>Total</td>
<td>100.00</td>
<td></td>
</tr>
</tbody>
</table>

### Table 7. Efficiency analysis.

<table>
<thead>
<tr>
<th>Model</th>
<th>Number of data entries</th>
<th>Ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA-based</td>
<td>17100</td>
<td>1</td>
</tr>
<tr>
<td>DT($a_6, a_9, a_{11}$)</td>
<td>21255</td>
<td>1.25</td>
</tr>
<tr>
<td>DT(all)</td>
<td>86890</td>
<td>5.08</td>
</tr>
</tbody>
</table>

### 6 Conclusion

Empirically, in a real data analysis, multiple class problems are frequently encountered, and a classifier which explains multiple classes simultaneously would be valuable and would exhibit higher readability. Many application domains (e.g., environmental science, actuarial science, biostatistics, population ecology, psychometrics, and quality control) highlight the usefulness in solving multi-class classification or optimization problems. When more alternative financial plans are launched for serving different customers, the decision makers of financial institutions have to promote many adequate plans to satisfy various customers’ requirements. It needs an efficient customer identification mechanism or a powerful classifier with acceptable prediction accuracy to solve such multi-class classification or optimization problems. This paper proposes an inductive learning model which captures the effective information after a more careful examination. A sounder classifier based on multivariate analysis is achieved by this model. Besides the effectiveness and efficiency verified by our experimental results, our model has flexibilities in three aspects. The first aspect is about feature evaluation method. In accordance with the practical need, data analyzers can replace the entropy-based measures with other methods such as Gini Index [17], goodness, or impurity. The second aspect regards computational performance. Our model can speed up the learning process by considering fewer features in case that analysis efficiency is more crucial than accuracy to the decision making. Finally, our model can cooperate with $k$-NN algorithm in retrieving a collection of similarity to every new query object for accurate reasoning.
References: