Customer credit quality assessments using data mining methods for banking industries

Shian-Chang Huang¹ and Cheng-Feng Wu²

¹Department of Business Administration, National Changhua University of Education, No.2, Shi-Da Rd., Changhua, Taiwan.
²Department of Business Administration, National Taiwan University, Taipei, Taiwan.

Accepted 19 April, 2011

Personal credit scoring on credit cards has been a critical issue in the banking industry. The bank with the most accurate estimation of its customer credit quality will be the most profitable. The study aims to compare quality prediction models from data mining methods, and improve traditional models by using boosting and genetic algorithms (GA). The predicting models used are instant-based classifiers (such as k-nearest neighbors), Bayesian networks, decision trees, decision tables, logistic regressions, radial basis function neural networks, and support vector machines. Three boosting (or ensemble) algorithms used for performance enhancement are AdaBoost, LogitBoost, and MultiBoost. The mentioned algorithms are optimized by GA for input features. Empirical results indicated that GA substantially improves the performance of underlying classifiers. Considering robustness and reliability, combining GA with ensemble classifiers is better than traditional models. Especially, integrating GA with LogitBoost (C4.5) is the most effective and compact model for credit quality evaluations.

Key words: Decision support system, credit risk assessment, genetic algorithm (GA), ensemble classifier, data mining.

INTRODUCTION

Credit cards have become a popular individual financial tool on making payments, consumer credit, depositing, and withdrawal of cash or bank transfer. For banking industries, credit cards are an important source of income, but credit card fraud also incurs great losses for them. With the wide spread use of credit card, personal credit scoring is getting increasingly important as more and more attention has been paid to the topic. Accurate credit quality estimation systems will substantially improve the profitability of the banking institutions (Thomas et al., 2002). Consequently, the objective of this paper is to develop a reliable and accurate model from data mining (Witten and Frank, 2005) methods for automatic credit quality assessments.

Many useful techniques, known as the credit scoring models, have been developed by the banks and researchers in order to solve the problems involved during the evaluation process. The objective of credit scoring models is to assign credit applicants to either a “good credit” group that is likely to repay financial obligation or a “bad credit” group with high possibility of defaulting. Therefore, credit scoring problems are basically a classification problem (Johnson and Wichern, 2002).

Data mining (or knowledge discovery) is a systematic approach to find underlying patterns, trend, and relationships buried in data. Data mining has drawn serious attention from both researchers and practitioners due to its wide applications in crucial business decisions. According to Curt (1995), the methodologies of data mining consist of data visualization, machine learning,
statistical techniques, and deductive database systems. The related applications using these methodologies can be summarized as classification, clustering, regression, summarization, dependency modeling, linkage analysis, and sequential analysis (Fayyad et al., 1996). The techniques to realize these applications include statistical methods, neural networks, decision trees, genetic algorithms, soft computing, and many non-parametric methods.

Recently, many approaches from data mining have been developing to support the credit admission decision. These methods include neural networks (Desai et al., 1996; Coakley and Brown, 2000; Tang and Chi, 2005; Abdou et al., 2008), support vector machine (Van Gestel et al., 2003; Huang et al., 2004; Chen and Shih, 2006), genetic algorithm (Varetto, 1998) and genetic programming (Ong et al., 2005). Owing to the high dimensionality of financial input data, some input variables or features may be irrelevant. Avoiding irrelevant features is important because they generally degenerate the performance of a classifier. This study attempts to enhance conventional credit scoring models by genetic algorithms (GA) (Mitchell, 1998). GA works by selection, crossover, and mutation can solve combinatorial optimization problems (feature selection in this study case) in very high dimensional space. The optimized feature subset for every classifier prevents its performance degeneracy from noisy and irrelevant input data.

Using two credit card data sets from UCI repository (Murphy and Aha, 2001), the study examines the performance of eight basic and three ensemble classifiers (based on three types of boosting algorithms). In the second stage, the study employs genetic algorithms to optimize the input features for every classifier.

The basic or underlying classifiers used in this study contain instance-based classifiers such as k-nearest neighbors (Cover and Hart, 1967), Bayesian networks (Borgelt and Kruse, 2002), decision trees (C4.5, Quinlan, 1993), and decision tables (King, 1967), logistic regressions, radial basis function neural networks (Haykin, 1999), and support vector machines (Vapnik, 1999). The ensemble (or combining) classifiers used are based on three types of boosting algorithms (Freund and Schapire 1999), the Adaboost (Freund and Schapire, 1996, 1997), Logitboost (Friedman et al., 1998), and Multiboost (Webb, 2000).

Boosting is a general method for improving the accuracy of any given learning algorithm (Freund and Schapire, 1999). It uses a function of the performance of base classifiers as a weight for voting to produce a powerful decision committee, and archives better performance.

Empirical results of this study demonstrate that the accuracies of GA optimized classifiers are better than pure classifiers, and combining classifiers are more robust than single classifiers.

Considering the robustness and reliability in performance, GA-LogitBoost (C4.5) model integrating GA, LogitBoost algorithms and C4.5 (base learner) is the most effective and compact system for credit assessments.

CONVENTIONAL CLASSIFIER AND FEATURE SELECTION METHODS

Conventional classifiers

Logistic regressions

Logistic regression is a model used for prediction of the probability of occurrence of an event by fitting data to a logistic curve. It is a generalized linear model used for binomial regression. It makes use of several predictor variables that may be either numerical or categorical. The formulation of logistic regression could be written as:

\[ f(x) = \frac{1}{1 + e^{-z}} \]

\[ z = x_1 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n \]

where \( x_1, x_2, \ldots, x_n \) are predictor variables for the regression. The classification output is type A if \( f(z) \geq 0 \), otherwise, type B.

Neural networks: RBF Networks

Neural networks (NNs) are mathematical models or computational models based on biological neural networks. It consists of an interconnected group of artificial neurons and processes information using a connectionist approach to computation. Practically, NNs are powerful non-linear data modeling tools for complex relationships between inputs and outputs or to find patterns in data. RBF network stands for Radial Basis Function (RBF) neural network. It has an input layer, a hidden layer and an output layer. The neurons in the hidden layer contain Gaussian transfer functions whose outputs are inversely proportional to the distance from the center of the neuron. Mathematically, the RBF network, including a linear part, produces an output \( y \) given by:

\[ y(\theta, X) = \sum_{j=1}^{nb} \alpha_j e^{-r^2(x-c)^2} + \beta X + b \]

where \( X \) is input vector, \( nb \) the number of neurons, each containing a basis function. The parameters of the RBF network consist of the positions of the basis functions \( C_i \), the inverse of the width of the basis functions \( \gamma \), the weights in output sum \( \alpha_i \), and the parameters of the linear part \( \beta \).

RBF networks are similar to K-means clustering in the first stage and general neural networks in the second stage. It combines the advantages of both compact representation of the input space and not suffering from local minima in the learning process.

Support vector machines

Support vector machines (SVMs) (Vapnik, 1999) are a set of related
supervised learning methods used for classification and regression. The formulation of SVM embodies the structural risk minimization principle (a maximum margin classifier), and thus combines excellent generalization properties with a sparse model representation. Viewing input data as vectors in a high dimensional transformed space, SVM seeks to construct a separating hyperplane in that space, which maximizes the margin between the two data sets. The SVM classification function is formulated as follows:

\[
y = \text{sign}(W^T \phi(X) + b), \quad y \in \{-1,1\}
\]  

(3)

where \( y \) is output (1 for type A, -1 for type B); \( \phi(X) \) is a nonlinear mapping from the input space to the high dimensional transformed space. Based on the structured risk minimization principle, SVMs seek to minimize the following function:

\[
\min_{w,b} R(w,\xi) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^{m} \xi_i
\]  

(4)

with

\[
y_i(W^T \phi(X) + b) \geq 1 + \xi_i, \quad i = 1,\ldots,m.
\]

\[
\xi_i \geq 0,
\]

where \( C \) is a prescribed parameter to evaluates the trade-off between the empirical risk and the smoothness of the model (structured risk).

Conventional feature selection methods

Chi-squared statistics

This method measures the importance of a feature by computing the value of the \( \chi^2 \)-statistic with respect to the class label (Witten and Frank, 2005).

Information gain

This method measures the importance of a feature by measuring the information gain with respect to the class. Information gain is given by:

\[
\text{InfoGain} = H(Y) - H(Y \mid X)
\]

(5)

where \( X \) and \( Y \) are features and

\[
H(Y) = \sum_{y \in Y} p(y) \log_2 p(y)
\]

(6)

\[
H(Y \mid X) = \sum_{x \in X} \sum_{y \in Y} p(y \mid x) \log_2 p(y \mid x)
\]

(7)

is the change in information entropy from a prior state to a state that takes some information as given. Both the information gain and the \( \chi^2 \)-statistic, are biased in favor of features with higher dispersion.

ReliefF

This method is feature weighting algorithm that is sensitive to feature interaction. The key idea of ReliefF is to rate features according to how well their values distinguish among instances of different classes and how well they cluster instances of the same class. To this end, ReliefF repeatedly chooses a single instance at random from data, and then locates the nearest instances of the same class and the nearest instances pertaining to different classes. The feature values of these instances are used to update the scores for each feature.

Symmetrical uncertainty

This method measures the importance of a feature by measuring the symmetrical uncertainty with respect to the class, and the balances for the information gain’s bias. It’s defined as

\[
SU = 2 \frac{\text{InfoGain}}{H(X) + H(Y)}
\]

COMBINING CLASSIFIERS: BOOSTING ALGORITHMS

Boosting (Freund, 1990) is a general way to improve the overall performance of basic or underlying learning algorithms. By combining weak classifiers, boosting methods could produce a powerful committee. The basic idea of boosting is applying a sequence of classifiers to the classification, where later classifiers focus on examples that are misclassified by earlier classifiers, weights the predictions of the classifiers with their errors, and finally combines all classifiers for test prediction.

AdaBoost (Freund and Schapire, 1996) is the abbreviation of adaptive boosting, which is an algorithm for constructing a strong classifier from linear combination of simple weak classifiers. AdaBoost has the following advantages: (1) it is very simple to implement; (2) it is a linear classifier with all desirable properties; (3) its output converges to the logarithm of likelihood ratio; (4) it can be combined with any weak learner. To solve the over fitting problem encountered by AdaBoost, Friedman et al. (1998) developed LogitBoost which could reduce training errors linearly and hence yield better generalization. LogitBoost is featured by introducing a log-likelihood loss function to reduce the sensitivity to noise and outliers, as well as by performing classification via combining many weak classifiers together to build up a very strong and robust classifier. MultiBoosting (Webb, 2000) is an extension to the highly successful AdaBoost technique for forming decision committees. MultiBoosting can be considered as wagging (Webb, 2000) committees formed by AdaBoost. It offers the further advantage over AdaBoost of suiting parallel execution.

GENETIC ALGORITHMS

Genetic algorithms (Mitchell, 1998) is an evolutionary algorithm that use techniques inspired by evolutionary biology such as selection, crossover and mutation, and used as simple models of evolutionary processes for solving complex optimization problems.

The evolution usually starts from a population of randomly generated individuals (parents). The fitness of every individual in the population is evaluated, and modified (randomly crossover and mutation) to form a new population (offspring). The new population is then used in the next iteration of the algorithm. The algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population, and then, an optimal solution is generated for the problem. GA is well adapted for feature selection problems with a large number of features. This study presents here, the main characteristics and adaptations made to deal with the particular
feature selection problem.

Encoding

A chromosome in our study is a string of binary bits whose size corresponds to the number of features. A 0 or 1, at position i, indicates whether the feature i is selected (1) or not (0).

Fitness function

Fitness function is an objective function that optimal solution should maximize in a genetic algorithm, so that the particular chromosome may be better ranked than all the other chromosomes. This fitness value reflects how well the chromosomes that represent, solve the given problem. This study used the accuracy of a classifier from a 10-fold (leave-one-out) cross-validation as the fitness function to select good chromosomes.

Genetic operators

After the assignments of binary encoding scheme and fitness function, our GA follows traditional genetic operators for binary chromosomes to search the whole solution space (Mitchell, 1998).

Selection

Chromosomes are selected from the population to crossover. Chromosomes are selected according to their fitness values, and the best ones should survive and create new offspring. This study used rank selection in this stage.

Crossover

Crossover combines two chromosomes to produce a new chromosome (offspring). It randomly selects a crossover point within a chromosome, and then interchanges the two parent chromosomes at this point to produce two new offspring. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents. Crossover occurs during evolution according to a user-definable crossover probability. This study used one point crossover and the point is randomly selected in this stage.

Mutation

The purpose of mutation in GA is to allow the algorithm to avoid local minima by preventing the population of chromosomes from becoming too similar to each other, thus slowing or even stopping evolution. This study used typical approach in binary chromosomes that selects a single bit in a chromosome and flip it.

EMPIRICAL RESULTS

Data sources and sampling

Two data sets are used in this study. The first one is Australian credit data set available from the UCI repository of machine learning databases (Murphy and Aha, 2001), in which 307 samples are creditworthy applicants and the other 383 samples with high possibility of default should be denied. This data set includes eight categorical attributes and six numerical attributes. Attributes of the credit set were converted into symbolic data set to maintain the confidentiality of the data source. The second data set, also from UCI is Japan credit data set which includes nine categorical attributes and six numerical attributes, has 690 samples, consisting of 307 creditworthy applicants and 383 bad customers.

Feature subset selection

The study proposes an applicant quality predicting model based on genetic algorithms and basic classifiers from data mining methods. The basic predicting models used are k-nearest neighbors (KNN, with k = 1 and 3), Bayesian network (BN), decision tree (C4.5), decision table (DT), logistic regression (LR), RBF neural network (RBFNN), support vector machine (SVM), and three combining classifiers (AdaBoost, LogitBoost, and MultiBoost with C4.5 as base learner for its compactness and effectiveness). In SVM, a grid search is used to optimize the model parameters. The kernel used is a polynomial kernel of degree 2, and the C value is set to be 2500 owing to its good performance. The details of all these algorithms may be found in Witten and Frank (2005). First, the performances of these classifiers are compared, and then they are optimized by genetic algorithms for input features in the second stage to develop reliable decision support systems. The parameters utilized in the GA algorithm are set as follows: the population size is 200; the chromosome crossover probability is 0.6; the chromosome mutation probability is 0.033, the number of iterations is 20. This study also compares the performance of GA with traditional feature selection methods including Chi-squared statistics, Information gain, ReliefF, symmetrical uncertainty.

Performance comparison

Empirical results (accuracy under ten-fold cross-validation) of Australian dataset are summarized in Table 1. In the first stage, logistic regression, AdaBoost (C4.5), and LogitBoost (C4.5) achieve higher accuracy than other classifiers. All their accuracies are above 86%. The second group is DT (85.80%), MultiBoost (C4.5) (85.51%), SVM (85.51%), and C4.5 (85.22%). Their accuracies are all above 85%. Empirical results of Japan
dataset are summarized in Table 2. In original classifiers, LogitBoost (C4.5) achieves best accuracy of 86.52%, followed by MultiBoost (C4.5) with 86.37%, SVM with 86.22%, LR and BN with 86.06%. Overall, in the two data sets, boosting classifiers are better than other classifiers.

**Performance improvement by feature selection**

In the second stage, GA is integrated with eleven models to optimize their input features. This study also performs traditional feature selection methods including Chi-squared statistics (X2), information gain (IG), RELIEFF (RF), and symmetrical uncertainty (SU) for comparison. Test results are also shown in Tables 1 and 2. In two dataset, the performance improvement owing to feature selection is significant. In Australian dataset, the performance enhancement of GA ranges from 0 to 7.11% in accuracy. Improvement on KNN1 is the most with approximately 7.11%. Compared with X2, IG, RF, SU, and GA, optimized features achieves best performance. In Japan dataset, the results are similar. The maximum performance improvement is 7.04% on KNN3.

Table 1 shows that GA-RBFNN, GA-LR, GA-LogitBoost (C4.5), GA-KNN1, GA-BN, and GA-C4.5 achieves
Table 3. The confusion matrix of every classifier for Australian data set.

<table>
<thead>
<tr>
<th>Classifier</th>
<th>TP Rate (%)</th>
<th>TN Rate (%)</th>
<th>Overall correct rate (%)</th>
<th>Type I error rate (%)</th>
<th>Type II error rate (%)</th>
<th>Overall incorrect rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN1</td>
<td>81.50</td>
<td>78.20</td>
<td>80.00</td>
<td>18.50</td>
<td>21.80</td>
<td>20.00</td>
</tr>
<tr>
<td>KNN3</td>
<td>81.50</td>
<td>78.20</td>
<td>80.00</td>
<td>18.50</td>
<td>21.80</td>
<td>20.00</td>
</tr>
<tr>
<td>BN</td>
<td>89.00</td>
<td>79.80</td>
<td>84.93</td>
<td>11.00</td>
<td>20.20</td>
<td>15.07</td>
</tr>
<tr>
<td>C4.5</td>
<td>88.00</td>
<td>81.80</td>
<td>85.22</td>
<td>12.00</td>
<td>18.20</td>
<td>14.78</td>
</tr>
<tr>
<td>DT</td>
<td>85.90</td>
<td>85.70</td>
<td>85.80</td>
<td>14.10</td>
<td>14.30</td>
<td>14.20</td>
</tr>
<tr>
<td>LR</td>
<td>86.20</td>
<td>87.90</td>
<td>86.96</td>
<td>13.80</td>
<td>12.10</td>
<td>13.04</td>
</tr>
<tr>
<td>SVM</td>
<td>79.90</td>
<td>92.50</td>
<td>85.51</td>
<td>20.10</td>
<td>7.50</td>
<td>14.49</td>
</tr>
<tr>
<td>RBFNN</td>
<td>90.30</td>
<td>73.30</td>
<td>82.75</td>
<td>9.70</td>
<td>26.70</td>
<td>17.25</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>87.20</td>
<td>85.00</td>
<td>86.23</td>
<td>12.80</td>
<td>15.00</td>
<td>13.77</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>86.40</td>
<td>85.70</td>
<td>86.09</td>
<td>13.60</td>
<td>14.30</td>
<td>13.91</td>
</tr>
<tr>
<td>MultiBoost</td>
<td>79.90</td>
<td>92.50</td>
<td>85.51</td>
<td>20.10</td>
<td>7.50</td>
<td>14.49</td>
</tr>
<tr>
<td>GA-KNN1</td>
<td>90.10</td>
<td>83.40</td>
<td>87.10</td>
<td>9.90</td>
<td>16.60</td>
<td>12.90</td>
</tr>
<tr>
<td>GA-KNN3</td>
<td>84.60</td>
<td>89.30</td>
<td>86.67</td>
<td>15.40</td>
<td>10.70</td>
<td>13.33</td>
</tr>
<tr>
<td>GA-BN</td>
<td>88.30</td>
<td>85.70</td>
<td>87.10</td>
<td>11.70</td>
<td>14.30</td>
<td>12.90</td>
</tr>
<tr>
<td>GA-C4.5</td>
<td>88.80</td>
<td>85.00</td>
<td>87.10</td>
<td>11.20</td>
<td>15.00</td>
<td>12.90</td>
</tr>
<tr>
<td>GA-DT</td>
<td>90.30</td>
<td>82.70</td>
<td>86.96</td>
<td>9.70</td>
<td>17.30</td>
<td>13.04</td>
</tr>
<tr>
<td>GA-LR</td>
<td>84.60</td>
<td>90.20</td>
<td>87.10</td>
<td>15.40</td>
<td>9.80</td>
<td>12.90</td>
</tr>
<tr>
<td>GA-SVM</td>
<td>79.90</td>
<td>92.50</td>
<td>85.51</td>
<td>20.10</td>
<td>7.50</td>
<td>14.49</td>
</tr>
<tr>
<td>GA-RBFNN</td>
<td>86.20</td>
<td>89.30</td>
<td>87.54</td>
<td>13.80</td>
<td>10.70</td>
<td>12.46</td>
</tr>
<tr>
<td>GA-AdaBoost</td>
<td>86.20</td>
<td>87.90</td>
<td>86.96</td>
<td>13.80</td>
<td>12.10</td>
<td>13.04</td>
</tr>
<tr>
<td>GA-LogitBoost</td>
<td>86.20</td>
<td>88.30</td>
<td>87.10</td>
<td>13.80</td>
<td>11.70</td>
<td>12.90</td>
</tr>
<tr>
<td>GA-MultiBoost</td>
<td>79.90</td>
<td>92.50</td>
<td>85.51</td>
<td>20.10</td>
<td>7.50</td>
<td>14.49</td>
</tr>
</tbody>
</table>

The confusion matrices of our predicting models are shown on Tables 3 and 4. The true positive (TP) and true negative (TN) are correct classifications. A false negative (FN) means the outcome is incorrectly predicted as negative when it is actually positive. The false negative is also known as a Type 1 error. A false positive (FP) means the outcome is incorrectly predicted as positive when it is actually negative. The false positive is also known as a Type 2 error.

In the case, Type 1 error refers to a situation where a creditworthy customer is incorrectly classified as bad customer. Type 2 error refers to the contrary case. In this study, Type 2 error is more important than Type 1 error, because Type 2 error refers to a bad applicant, incorrectly classified as a creditworthy applicant. The institution auditor and the department of credit management may under-estimate the risk and suffer serious bad debt losses. In GA optimized models, Type 2 errors are significantly lower than Type 1 errors. Consequently, it's helpful for investors in controlling their risk. As Table 3 shows, the TP or TN rates of GA optimized models have observably risen; and the FP or FN rates have observably reduced. The lowest Type 2 error is achieved by SVM, GA-SVM, and GA-MultiBoost(C4.5). In Japan dataset, the results are similar.

Comparison of the selected features

The study finds that the feature rankings of four with the
Table 4. The confusion matrix of every classifier for Japan data set.

<table>
<thead>
<tr>
<th></th>
<th>TP rate (%)</th>
<th>TN rate (%)</th>
<th>Overall correct rate (%)</th>
<th>Type I error rate (%)</th>
<th>Type II error rate (%)</th>
<th>Overall incorrect rate (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>KNN1</td>
<td>77.70</td>
<td>84.00</td>
<td>81.16</td>
<td>22.30</td>
<td>16.00</td>
<td>18.84</td>
</tr>
<tr>
<td>KNN3</td>
<td>77.70</td>
<td>84.00</td>
<td>81.16</td>
<td>22.30</td>
<td>16.00</td>
<td>18.84</td>
</tr>
<tr>
<td>BN</td>
<td>81.40</td>
<td>89.90</td>
<td>86.06</td>
<td>18.60</td>
<td>10.10</td>
<td>13.94</td>
</tr>
<tr>
<td>C4.5</td>
<td>83.40</td>
<td>86.80</td>
<td>85.30</td>
<td>16.60</td>
<td>13.20</td>
<td>14.70</td>
</tr>
<tr>
<td>DT</td>
<td>86.50</td>
<td>84.60</td>
<td>85.45</td>
<td>13.50</td>
<td>15.40</td>
<td>14.55</td>
</tr>
<tr>
<td>LR</td>
<td>88.20</td>
<td>84.30</td>
<td>86.06</td>
<td>11.80</td>
<td>15.70</td>
<td>13.94</td>
</tr>
<tr>
<td>SVM</td>
<td>93.60</td>
<td>80.10</td>
<td>86.22</td>
<td>6.40</td>
<td>19.90</td>
<td>13.78</td>
</tr>
<tr>
<td>RBFNN</td>
<td>66.60</td>
<td>91.90</td>
<td>80.40</td>
<td>33.40</td>
<td>8.10</td>
<td>19.60</td>
</tr>
<tr>
<td>AdaBoost</td>
<td>86.10</td>
<td>85.70</td>
<td>85.91</td>
<td>13.90</td>
<td>14.30</td>
<td>14.09</td>
</tr>
<tr>
<td>LogitBoost</td>
<td>85.80</td>
<td>87.10</td>
<td>86.52</td>
<td>14.20</td>
<td>12.90</td>
<td>13.48</td>
</tr>
<tr>
<td>MultiBoost</td>
<td>93.90</td>
<td>80.10</td>
<td>86.37</td>
<td>6.10</td>
<td>19.90</td>
<td>13.63</td>
</tr>
<tr>
<td>GA-KNN1</td>
<td>81.10</td>
<td>86.30</td>
<td>83.92</td>
<td>18.90</td>
<td>13.70</td>
<td>16.08</td>
</tr>
<tr>
<td>GA-KNN3</td>
<td>89.20</td>
<td>87.40</td>
<td>88.21</td>
<td>10.80</td>
<td>12.60</td>
<td>11.79</td>
</tr>
<tr>
<td>GA-BN</td>
<td>86.10</td>
<td>88.20</td>
<td>87.29</td>
<td>13.90</td>
<td>11.80</td>
<td>12.71</td>
</tr>
<tr>
<td>GA-C4.5</td>
<td>87.80</td>
<td>84.00</td>
<td>85.76</td>
<td>12.20</td>
<td>16.00</td>
<td>14.24</td>
</tr>
<tr>
<td>GA-DT</td>
<td>90.20</td>
<td>84.90</td>
<td>87.29</td>
<td>9.80</td>
<td>15.10</td>
<td>12.71</td>
</tr>
<tr>
<td>GA-LR</td>
<td>91.20</td>
<td>84.60</td>
<td>87.60</td>
<td>8.80</td>
<td>15.40</td>
<td>12.40</td>
</tr>
<tr>
<td>GA-SVM</td>
<td>94.60</td>
<td>80.10</td>
<td>86.68</td>
<td>5.40</td>
<td>19.90</td>
<td>13.32</td>
</tr>
<tr>
<td>GA-RBFNN</td>
<td>88.20</td>
<td>85.40</td>
<td>86.68</td>
<td>11.80</td>
<td>14.60</td>
<td>13.32</td>
</tr>
<tr>
<td>GA-AdaBoost</td>
<td>88.20</td>
<td>85.20</td>
<td>86.52</td>
<td>11.80</td>
<td>14.80</td>
<td>13.48</td>
</tr>
<tr>
<td>GA-LogitBoost</td>
<td>92.60</td>
<td>84.30</td>
<td>88.06</td>
<td>7.40</td>
<td>15.70</td>
<td>11.94</td>
</tr>
<tr>
<td>GA-MultiBoost</td>
<td>93.90</td>
<td>80.10</td>
<td>86.37</td>
<td>6.10</td>
<td>19.90</td>
<td>13.63</td>
</tr>
</tbody>
</table>

most discrimination power are completely different for different classifiers and also different from those selected by other four feature selectors. In Japan dataset, X2, IG, RF, and XU all rank features 9,11,10,8,15 in high priority, but GA selects features 2,3,4,5,8,15 for LR, 5,9,10,13 for RBFNN, 7,8,9,10,12,14 for LogitBoost (C4.5), and 3,7,9,12,13,14,15 for C4.5.

The advantages of GA method are the automatic determination of the best number and best combination of features to achieve best performance of every classifier. The most important is that GA could prune irrelevant and highly correlated features, whereas, two variables could be considered good predictors individually. There could be little gain to combine them together in a subset especially when the two variables are highly multicollinear. That is, complementary information is more important than similar information.

CONCLUSIONS AND FUTURE RESEARCH

How to effectively evaluate the credit quality of an applicant is a major problem in credit management. This paper applies data mining methods to develop an automatic assessment support system. The basic classifiers used in this research include k-nearest neighbors, Bayesian networks, decision trees (C4.5), decision tables, logistic regressions, RBF neural networks, support vector machines, and three combining classifiers (AdaBoost, LogitBoost, and MultiBoost). In the second stage, this study uses genetic algorithms to optimize the input features of these classifiers for constructing a reliable and robust credit scoring system.

The empirical results of this study demonstrate that the accuracies of GA optimized classifiers are better than pure classifiers, and combining classifiers are more robust than single classifiers. Considering the robustness and reliability in performance, GA-LogitBoost (C4.5) model which integrates GA with Logitboost algorithm is the most effective and compact model for credit scoring.

The results of this study can provide references for auditors on their decision making in credit management. Future research may consider more input variables for classifiers. But including more information does not guarantee higher accuracy. A more efficient parallel GA optimization schemes should be developed for higher dimensional data.

REFERENCES


Chen WH, Shih JY (2006). A paper of Taiwan’s issuer credit rating