Available online at www.sciencedirect.com

ScienceDirect

Expert Systems with Applications 33 (2007) 847–856

Expert Systems with Applications

www.elsevier.com/locate/eswa

Credit scoring with a data mining approach based on support vector machines

Cheng-Lung Huang [a,*], Mu-Chen Chen [b], Chieh-Jen Wang [c]

[a] *National Kaohsiung First University of Science and Technology, Department of Information Management, 2, Juoyue Road, Nantz District, Kaohsiung 811, Taiwan*

[b] *Institute of Traffic and Transportation, National Chiao Tung University, 4F, No. 118, Section 1, Chung Hsiao W. Road, Taipei 10012, Taiwan, ROC*

[c] *Department of Information Management, Huafan University, 1, Huafan Rd., Shihtin Hsiang, Taipei Hsien 223, Taiwan, ROC*

Abstract

The credit card industry has been growing rapidly recently, and thus huge numbers of consumers' credit data are collected by the credit department of the bank. The credit scoring manager often evaluates the consumer's credit with intuitive experience. However, with the support of the credit classification model, the manager can accurately evaluate the applicant's credit score. Support Vector Machine (SVM) classification is currently an active research area and successfully solves classification problems in many domains. This study used three strategies to construct the hybrid SVM-based credit scoring models to evaluate the applicant's credit score from the applicant's input features. Two credit datasets in UCI database are selected as the experimental data to demonstrate the accuracy of the SVM classifier. Compared with neural networks, genetic programming, and decision tree classifiers, the SVM classifier achieved an identical classificatory accuracy with relatively few input features. Additionally, combining genetic algorithms with SVM classifier, the proposed hybrid GA-SVM strategy can simultaneously perform feature selection task and model parameters optimization. Experimental results show that SVM is a promising addition to the existing data mining methods.

Keywords: Credit scoring; Support vector machine; Genetic programming; Neural networks; Decision tree; Data mining; Classification

1. Introduction

Recently, competition in the consumer credit market has become severe. With the rapid growth in the credit industry, credit scoring models have been extensively used for the credit admission evaluation (Thomas, 2000). In the last two decades, several quantitative methods have been developed for the credit admission decision. The credit scoring models are developed to categorize applicants as either accepted or rejected with respect to the applicants' characteristics such as age, income, and marital condition. Credit officers are faced with the problem of trying to increase credit volume without excessively increasing their exposure to default. Therefore, to screen credit applications, new techniques should be developed to help predict credits more accurately. The benefits of credit scoring involve reducing the credit analysis cost, enabling faster credit decisions, closer monitoring of existing accounts and prioritizing credit collections (Brill, 1998).

In the credit and banking area, a number of articles have been published, which herald the role of automatic approaches in helping creditors and bankers make loans, develop markets, assess creditworthiness and detect fraud. Creditors accept the credit application provided that the applicant is expected to repay the financial obligation. Creditors construct the credit classification rules (credit

[*] Corresponding author. Tel.: +886 7 6011000x4127; fax: +886 7 6011042.
E-mail address: clhuang@ccms.nkfust.edu.tw (C.-L. Huang).

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.07.007

scoring models) based on the data of the previous accepted and rejected applicants. With sizeable loan portfolios, even a slight improvement in credit scoring accuracy can reduce the creditors' risk and translate considerably into future savings.

The modern data mining techniques, which have made a significant contribution to the field of information science (Chen & Liu, 2004), can be adopted to construct the credit scoring models. Practitioners and researchers have developed a variety of traditional statistical models and data mining tools for credit scoring, which involve linear discriminant models (Reichert, Cho, & Wagner, 1983), logistic regression models (Henley, 1995), *k*-nearest neighbor models (Henley & Hand, 1996), decision tree models (Davis, Edelman, & Gammerman, 1992), neural network models (Desai, Crook, & Overstreet, 1996; Malhotra & Malhotra, 2002; West, 2000), and genetic programming models (Ong, Huang, & Tzeng, 2005). From the computational results made by Tam and Kiang (1992), the neural network is most accurate in bank failure prediction, followed by linear discriminant analysis, logistic regression, decision trees, and *k*-nearest neighbor. In comparison with other techniques, they concluded that neural network models are more accurate, adaptive and robust.

Desai et al. (1996) investigated neural networks, linear discriminant analysis and logistic regression for scoring credit decision. They concluded that neural networks outperform linear discriminant analysis in classifying loan applicants into good and bad credits, and logistic regression is comparable to neural networks. West (2000) investigated the credit scoring accuracy of several neural networks. Results were benchmarked against traditional statistical methods such as linear discriminant analysis, logistic regression, *k*-nearest neighbor and decision trees. Malhotra and Malhotra (2002) applied neuro-fuzzy models to analyze consumer loan applications and compared the advantages of neuro-fuzzy systems over traditional statistical techniques in credit-risk evaluation. Hoffmann, Baesens, Martens, Put, and Vanthienen (2002) applied a genetic fuzzy and a neuro-fuzzy classifier for credit scoring. Baesens et al. (2003) benchmarked state-of-the-art classification algorithms for credit scoring.

Recently, researchers have proposed the hybrid data mining approach in the design of an effective credit scoring model. Hsieh (2005) proposed a hybrid system based on clustering and neural network techniques; Lee and Chen (2005) proposed a two-stage hybrid modeling procedure with artificial neural networks and multivariate adaptive regression splines; Lee, Chiu, Lu, and Chen (2002) integrated the backpropagation neural networks with traditional discriminant analysis approach; Chen and Huang (2003) presents a work involving two interesting credit analysis problems and resolves them by applying neural networks and genetic algorithms techniques.

Since even a fraction of improvement in credit scoring accuracy may translate into noteworthy future savings, the major issue of previous studies focused on increasing the accuracy of credit decisions. For conventional statistical classification techniques, an underlying probability model must be assumed in order to calculate the posterior probability upon which the classification decision is made. The more recently developed data mining techniques such as neural networks, genetic programming (GP) and support vector machines (SVM) can perform the classification task without this limitation. Additionally, these artificial intelligence methods also achieved better performance than traditional statistical methods.

Support vector machines (SVM) were first suggested by Vapnik (1995) and have recently been used in a range of problems including pattern recognition (Pontil & Verri, 1998), bioinformatics (Yu, Ostrouchov, Geist, & Samatova, 2003), and text categorization (Joachims, 1998). Huang, Chen, Hsu, Chen, and Wu (2004) obtained prediction accuracy around 80% for both backpropagation neural networks and SVM methods for the United States and Taiwan markets. When using SVM, two problems are confronted: how to choose the optimal input feature subset for SVM and how to set the best kernel parameters. These two problems are crucial because the feature subset choice influences the appropriate kernel parameters and vice versa (Fröhlich & Chapelle, 2003). Therefore, this study proposed hybrid SVM-based approaches to optimize the input feature subset and model parameters.

Feature selection is an important issue in building classification systems. It is advantageous to limit the number of input features in a classifier in order to have a good predictive and less computationally intensive model (Zhang, 2000). With a small feature set, the explanation of rationale for the classification decision can be easier realized. In addition to the feature selection, proper model parameters setting can improve the SVM classification accuracy. The parameters that should be optimized include penalty parameter C and the kernel function parameters such as the gamma (γ) for the radial basis function (RBF) kernel. To design a SVM, one must choose a kernel function, set the kernel parameters and determine a soft margin constant C. The grid algorithm is an alternative to finding the best C and gamma when using the RBF kernel function (Hsu & Lin, 2002). Besides the grid algorithm, other optimization tools such as genetic algorithm, which is adopted in this study, can also be applied to optimize the feature subset and model parameter. To successfully build credit scoring models, this study tried three SVM-based strategies: (1) using grid search to optimize model parameters, (2) using grid search to optimize model parameters and using F-score calculation to select input features, and (3) using genetic algorithm to simultaneously optimize model parameters and input features.

This paper is organized as follows. Section 2 describes basic SVM concepts. Section 3 describes three SVM-based strategies used in this research. Section 4 presents the experimental results from using the proposed method to classify two real world datasets. Section 5 gives remarks and provides a conclusion.

2. Basic concepts of SVM classifier

In this section we will briefly describe the basic SVM concepts for typical two-class classification problems. These concepts can also be found in Kecman (2001), Scho̸lkopf and Smola (2000), and Cristianini and Shawe-Taylor (2000).

Given a training set of instance-label pairs (x_i, y_i), $i = 1, 2, \ldots, m$ where $x_i \in R^n$ and $y_i \in \{+1, -1\}$, SVM finds an optimal separating hyperplane with the maximum margin by solving the following optimization problem:

$$\begin{aligned} &\underset{w,b}{\text{Min}} && \frac{1}{2} w^{\mathrm{T}} w \\ &\text{subject to:} && y_i(\langle w \cdot x_i \rangle + b) - 1 \geqslant 0 \end{aligned} \tag{1}$$

It is known that to solve this quadratic optimization problem one must find the saddle point of the Lagrange function:

$$L_p(w, b, \alpha) = \frac{1}{2} w^{\mathrm{T}} \cdot w - \sum_{i=1}^{m} (\alpha_i y_i (\langle w \cdot x_i \rangle + b) - 1) \tag{2}$$

where the α_i denotes Lagrange multipliers, hence $\alpha_i \geqslant 0$. The search for an optimal saddle point is necessary because the L_p must be minimized with respect to the primal variables w and b and maximized with respect to the non-negative dual variable α_i. By differentiating with respect to w and b, and introducing the Karush Kuhn–Tucker (KKT) condition for the optimum constrained function, then L_p is transformed to the dual Lagrangian $L_D(\alpha)$:

$$\begin{aligned} &\underset{\alpha}{\text{Max}} && L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \\ &\text{subject to:} && \alpha_i \geqslant 0 \quad i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{3}$$

To find the optimal hyperplane, a dual Lagrangian $L_D(\alpha)$ must be maximized with respect to non-negative α_i. The solution α_i for the dual optimization problem determines the parameters w^* and b^* of the optimal hyperplane. Thus, the optimal hyperplane decision function $f(x) = \text{sgn}(\langle w^* \cdot x \rangle + b^*)$ can be written as

$$f(x) = \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle x_i, x \rangle + b^* \right) \tag{4}$$

In a typical classification task, only a small subset of the Lagrange multipliers α_i usually tends to be greater than zero. Geometrically, these vectors are the closest to the optimal hyperplane. The respective training vectors having nonzero α_i are called support vectors, as the optimal decision hyperplane $f(x, \alpha^*, b^*)$ depends on them exclusively.

The above concepts can also be extended to the non-separable case (linear generalized SVM). In terms of these introduced slack variables, the problem of finding the hyperplane that provides the minimum number of training errors (i.e., to keep the constraint violation as small as possible) has the formal expression as follows:

$$\begin{aligned} &\underset{w,b,\xi}{\text{Min}} && \frac{1}{2} w^{\mathrm{T}} w + C \sum_{i=1}^{m} \xi_i \\ &\text{subject to:} && y_i(\langle w \cdot x_i \rangle + b) + \xi_i - 1 \geqslant 0 \\ & && \xi_i \geqslant 0 \end{aligned} \tag{5}$$

where C is a penalty parameter on the training error, and ξ_i is the non-negative slack variable. SVM finds the hyperplane that provides the minimum number of training errors (i.e., to keep the constraint violation as small as possible).

This optimization model can be solved using the Lagrangian method, which is almost equivalent to the method for solving the optimization problem in the separable case. One must maximize the dual variables Lagrangian:

$$\begin{aligned} &\underset{\alpha}{\text{Max}} && L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i \cdot x_j \rangle \\ &\text{subject to:} && 0 \leqslant \alpha_i \leqslant C \quad i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0 \end{aligned} \tag{6}$$

To find the optimal hyperplane, a dual Lagrangian $L_D(\alpha)$ must be maximized with respect to non-negative α_i under the constrains $\sum_{i=1}^{m} \alpha_i y_i = 0$ and $0 \leqslant \alpha_i \leqslant C$. The penalty parameter C, which is now the upper bound on α_i, is determined by the user. Finally, the form of optimal hyperplane decision function is the same as (4).

The nonlinear SVM maps the training samples from the input space into a higher-dimensional feature space via a mapping function Φ. In the dual Lagrange (6), the inner products are replaced by the kernel function (7), and the nonlinear SVM dual Lagrangian $L_D(\alpha)$ (8) is similar with that in the linear generalized case

$$(\Phi(x_i) \cdot \Phi(x_j)) := k(x_i, x_j) \tag{7}$$

$$L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j k(x_i \cdot x_j) \tag{8}$$

Subject to: $0 \leqslant \alpha_i \leqslant C$, $i = 1, \ldots, m$ and $\sum_{i=1}^{m} \alpha_i y_i = 0$.

Followed by the steps described in the linear generalized case, we obtain decision function of the following form:

$$\begin{aligned} f(x) &= \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle \Phi(x), \Phi(x_i) \rangle + b^* \right) \\ &= \text{sgn}\left(\sum_{i=1}^{m} y_i \alpha_i^* \langle k(x, x_i) \rangle + b^* \right) \end{aligned} \tag{9}$$

Radial basis function (RBF) is a common kernel function as follows:

$$k(x_i, x_j) = \exp\left(-\gamma \|x_i - x_j\|^2\right) \tag{10}$$

3. Strategies for building SVM classifier

3.1. Setting model parameters using grid search

Proper parameters setting can improve the SVM classification accuracy. With the RBF kernel, there are two parameters to be determined in the SVM model: C and gamma (γ). The grid search approach (Hsu, Chang, & Lin, 2003) is an alternative to finding the best C and gamma when using the RBF kernel function.

To guarantee that the present results are valid and can be generalized for making predictions regarding new data, the data set is further randomly partitioned into training and independent testing sets via a k-fold cross validation. Each of the k subsets acts as an independent holdout test set for the model trained with the rest of $k - 1$ subsets. The advantages of cross validation are that the impact of data dependency is minimized and the reliability of the results can be improved (Salzberg, 1997).

In the grid search approach, pairs of (C,γ) are tried and the one with the best cross-validation accuracy is chosen. After identifying a "better" region on the grid, a finer grid search on that region can be conducted. To get good generalization ability, grid search approach uses a validation process to decide parameters. That is, for each of the k subsets of the data set D, create a training set $T = D - k$, then run a cross-validation process as follows (Chen & Lin, 2005; Hsu et al., 2003):

Step 1. Consider a grid space of (C,γ) with $\log_2 C \in \{-5, -4,\ldots,12\}$ and $\log_2 \gamma \in \{-12,-13,\ldots,5\}$.
Step 2. For each hyperparameter pair (C,γ) in the search space, conduct 5-fold cross validation on the training set.
Step 3. Choose the parameter (C,γ) that leads to the lowest CV error classification rate.
Step 4. Use the best parameter to create a model as the predictor.

Overall accuracy is averaged across all k partitions. These k accuracy values also give an estimate of the accuracy variance of the algorithms.

3.2. Setting model parameters using grid search and selecting input features using F-score

In addition to the proper parameters setting, feature subset selection can improve the SVM classification accuracy. F-score (Chen & Lin, 2005) is a simple technique which measures the discrimination of two sets of real numbers. Given training vectors x_k, $k = 1,2,\ldots,m$, if the number of positive and negative instances are n_+ and n_-, respectively, then the F-score of the ith feature is defined as follows (Chen & Lin, 2005):

$$F(i) \equiv \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n_+ - 1}\sum_{k=1}^{n_+}(x_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n_- - 1}\sum_{k=1}^{n_-}(x_{k,i}^{(-)} - \bar{x}_i^{(-)})^2} \tag{11}$$

where $\bar{x}_i$, $\bar{x}_i^{(+)}$, and $\bar{x}_i^{(-)}$ are the averages of the ith feature of the whole, positive, and negative data sets, respectively; $x_{k,i}^{(+)}$ is the ith feature of the kth positive instance, and $x_{k,i}^{(-)}$ is the ith feature of the kth negative instance. The numerator indicates the discrimination between the positive and negative sets, and the denominator indicates the one within each of the two sets. The larger the F-score is, the more likely this feature is more discriminative (Chen & Lin, 2005). One can select the features manually; however, this study follows the following procedure. For each of the k subsets of the data set D, create a training set $T = D - k$, then run a cross-validation process, and the overall accuracy is averaged across all k partitions. The procedure is as follows (Chen & Lin, 2005):

Step 1. Calculate F-score for every feature.
Step 2. Sort F-score, and set possible number of features by $f = [n/2^i]$, $i \in \{0,1,2,\ldots,m\}$, where m is an integer with $n/2^m \geqslant 1$.
Step 3. For each f (threshold), do the following:
(a) Keep the first f features according to the F-score.
(b) Randomly split the training data into D_{training} and $D_{\text{validation}}$ using 5-fold cross validation. Do the following step for each fold:
(c) Let D_{training} be the new training data. Use the SVM procedure in Section 3.1 to obtain a predictor; use the predictor to predict $D_{\text{validation}}$.
(d) Calculate the average validation error of the 5-fold cross validation.
Step 4. Choose the f (threshold) with the lowest average validation error.
Step 5. Drop features with F-score below the selected threshold. Rerun SVM training in Section 3.1 on the larger set T. And measure classification accuracy on test set k.

3.3. Optimizing model parameter and feature subset using GA-based approach

When using SVM, obtaining the optimal feature subset and SVM parameters must occur simultaneously. In the literature, only a few algorithms have been proposed for SVM feature selection (Fröhlich & Chapelle, 2003; Guyon, Weston, Barnhill, & Bapnik, 2002; Mao, 2004; Weston et al., 2001). Somol, Baesens, Pudil, and Vanthienen (2005) studied filter and wrapper-based feature selection for credit scoring. Fröhlich and Chapelle (2003) proposed a GA-based feature selection approach that used the theoretical bounds on the generalization error for SVMs. However, previous research neither deals with parameters optimization for the SVM classifier nor focuses on building a credit scoring model based on SVM model.

Genetic algorithms (Goldberg, 1989; Holland, 1975) have the potential to generate both the optimal feature subset and SVM parameters at the same time. This paper used

GA-based approach to optimize the parameters and feature subset simultaneously, without degrading the SVM classification accuracy. The proposed method performs feature selection and parameters setting in an evolutionary way.

When the RBF kernel is selected, the parameters (C and γ) and features used as input attributes must be optimized using our proposed GA-based system. Therefore, the chromosome is comprised of three parts, C, γ, and the feature mask. The binary coding system was used to represent the chromosome. Note that we can choose the length of bit strings representing C and γ according to the calculation precision required; meanwhile, the number of features varies from the different datasets. The bit strings representing the genotype of parameter C and γ should be transformed into phenotype by converting binary into decimal representation. For the chromosome representing the feature mask, the bit with value "1" means the feature is selected, and "0" indicates the feature is not selected.

We used the same 10-cross validation procedure used in the previous section. During the evolutionary process of GA, we conduct 5-fold cross validation on the training set for each pair of (C, γ) with the selected features. The fitness is defined as the average accuracy of five folds cross validation, and GA finds the evolutionary direction based on the fitness.

Since the training of GA is a stochastic evolutionary process, the accuracy rates for each data set partitions are themselves averages of five repetitions. That is, five repetitions are conducted for each of the 10 partitions to reduce the stochastic variability of model training process in GA-based SVM.

For each of the k subsets of the data set D, create a training set $T = D - k$, then run a cross-validation process as follows:

Step 1. For each of the k folds, run the following steps:
 Step 1.1. Initialization: Generate initial population which individually is comprised of C, γ, and selected features.
 Step 1.2. Fitness Evaluation: For each individual population, do the following:
  (a) Select the features that represent the chromosome bit string, and use C and gamma to represent the bit string.
  (b) For the (C, γ) with selected features, randomly split the training data into D_{training} and $D_{\text{validation}}$ using 5-fold cross validation, conduct 5-fold cross validation on the training set. Then calculate the average validation accuracy of the 5-fold cross validation.
  (c) Set the fitness to the average validation accuracy obtained in the above step.
 Step 1.3. Genetic operations: Perform genetic operation: selection, recombination, mutation, and replacement, and form a new population.
 Step 1.4. Stop condition: Go to step 1.2 unless stopping criteria are met.
 Step 1.5. Once the parameters are optimized via the above GA process, rerun SVM training on the larger set T to obtain a trained SVM classifier. Based on the trained SVM model, measure classification accuracy on the test set k.
Step 2. Overall accuracy is averaged across all k partitions. These k values also give an estimate of the variance of the algorithms.

Based on whether feature selection is performed independently of the learning algorithm that constructs the classifier, feature subset selection algorithms can be classified into two categories: the filter approach and the wrapper approach (John, Kohavi, & Peger, 1994; Kohavi & John, 1997; Liu & Motoda, 1998). The filter approach selects important features first and then SVM is applied for classification. On the other hand, the wrapper approach either modifies SVM to choose important features as well as conducts training/testing or combines SVM with other optimization tools to perform feature selection. By the definition of Liu and Motoda (1998), the F-score + SVM approach in Section 3.2 is a filter approach as shown in Fig. 1, while GA + SVM proposed in this section is a wrapper approach as shown in Fig. 2.

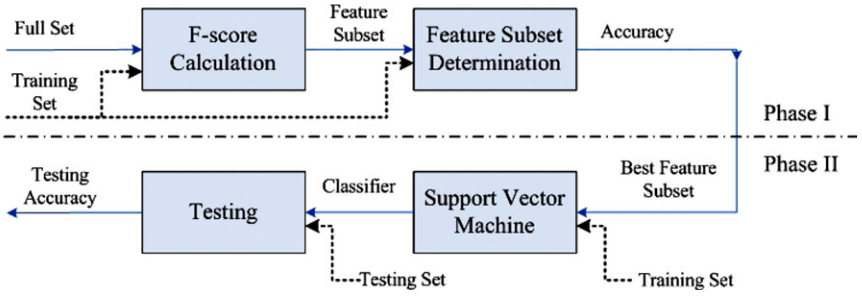

Fig. 1. A filter model of feature selection modified from Liu and Motoda (1998).

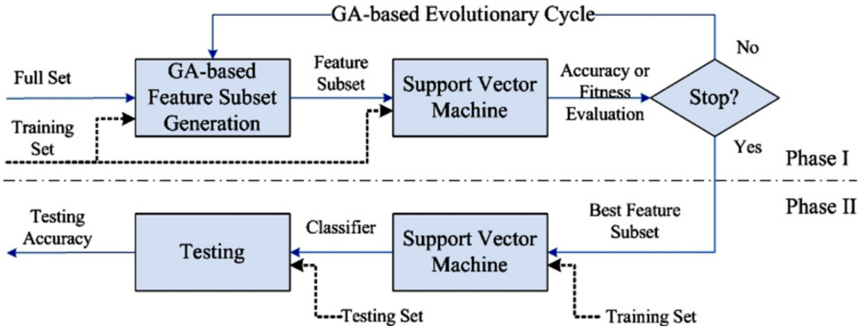

Fig. 2. A wrapper model of feature selection modified from Liu and Motoda (1998).

4. Empirical analysis

4.1. Real world credit data sets

The two real world data sets illustrated in Table 1, the Australian and German credit data sets, are available from the UCI Repository of Machine Learning Databases (Murphy & Aha, 2001) and are adopted herein to evaluate the predictive accuracy. The Australian credit data consists of 307 instances of creditworthy applicants and 383 instances where credit is not creditworthy. Each instance contains 6 nominal, 8 numeric attributes, and 1 class attribute (accepted or rejected). This dataset is interesting because there is a good mixture of attributes: continuous, nominal with small numbers of values, and nominal with larger numbers of values. There are also a few missing values. To protect the confidentiality of data, the attributes names and values have been changed to meaningless symbolic data. The German credit scoring data are more unbalanced, and it consists of 700 instances of creditworthy applicants and 300 instances where credit should not be extended. For each applicant, 24 input variables describe the credit history, account balances, loan purpose, loan amount, employment status, personal information, age, housing, and job title. This data set only consists of numeric attributes.

4.2. Experimental results

Three strategies as stated in Section 3 were used in this study, namely "SVM + Grid," "SVM + Grid + *F*-score," and "SVM + GA." The results for the two data sets were obtained by using the three strategies and are summarized in Tables 2 and 3, respectively.

For the Australian data set, the classificatory accuracy (hit rate) of the three models achieved 85.51%, 84.20%, and 86.90%. The experiments of the three strategies used identical training and testing sets for each "treatment" (three strategies); thus the test set is clearly not independent (Salzberg, 1997). To compare classification accuracy of the test set; therefore, we performed a nonparametric Friedman test—a test for the *k*-related (dependent) samples. We found no significant differences among these three strategies (with $p = 0.349$). That means the three SVM strategies achieved the same classificatory accuracy. In Table 2, the average numbers of selected features are 14.0, 7.6, and 7.3. For the number of selected features, we found no significant differences between "SVM + Grid + *F*-score" and "SVM + GA" based on the nonparametric Wilcoxon signed rank tests ($p = 0.567$).

For the German data set, as shown in Table 3, the classificatory accuracy of the three models achieved 76.00%, 77.50%, and 77.92%, and the average of the selected features are 24.0, 20.4, and 13.3, respectively. There are no significant differences among these three strategies based on the Friedman test (with $p = 0.32$) for classificatory accuracy. However, "SVM + GA" has significantly lesser number of selected features than the "SVM + Grid + *F*-score" based on the nonparametric Wilcoxon signed rank tests ($p = 0.001$).

Table 1
Datasets from the UCI repository

No.	Names	# classes	# instances	Nominal features	Numeric features	Total features
1	German	2	1000	0	24	24
2	Australian	2	690	6	8	14

Table 2
Results summary with 10-fold cross validation for Australian credit data set

	Selected features		Hit rate	
	Avg.	Std.	Avg. (%)	Std. (%)
SVM + Grid search	14.0	–	85.51	3.78
SVM + Grid search + *F*-score	7.6	1.20	84.20	4.51
SVM + GA	7.3	1.65	86.90	4.22

Table 3
Results summary with 10-fold cross validation for German credit data set

	Selected features		Hit rate	
	Avg.	Std.	Avg. (%)	Std. (%)
SVM + Grid search	24.0	–	76.00	3.86
SVM + Grid search + *F*-score	20.4	5.50	77.50	4.03
SVM + GA	13.3	1.41	77.92	3.97

According to the above Friedman test results, the classificatory accuracies of the three strategies are identical in the two data sets, although the "SVM + GA" accuracy is slightly higher than the other two strategies. Although achieving as high as the other two strategies, the "SVM + GA" strategy has lesser selected features in one of the two data sets. The above results reveal that a GA-based strategy is an acceptable alternative to optimize both the feature subset and model parameters for credit scoring.

4.3. Comparison with other methods

In the previous studies, the artificial intelligence based approaches such as neural network and GP have been successfully applied to credit analysis, and they are usually more accurate (Desai et al., 1996; Ong et al., 2005; West, 2000). Decision tree is a popular approach to build classification model. The credit scoring results of SVM, therefore, are benchmarked to those generated by the back-propagation neural network (BPN), GP (Koza, 1992) and decision tree (C4.5) (Quinlan, 1986, 1993).

In this paper of credit scoring, BPN as well as GP is used for a two-class pattern classification. A simple thresholding scheme is sufficient for the BPN and GP to divide the feature space into two categories in a two-class classification problem. A threshold value of 0.5 is used to distinguish between credit groups, good credit and bad credit. If the output result of BPN or GP is greater than or equal to 0.5, the input sample is assigned to one class (good, accepted); otherwise it is assigned to the other class (bad, rejected). Hence,

$$\begin{cases} \text{If } O_{\text{model}}(X_i) \geqslant 0.5, \quad O_i = 1 \text{ and } X_i \in \text{Class of good credit;} \\ \text{If } O_{\text{model}}(X_i) < 0.5, \quad O_i = 0 \text{ and } X_i \in \text{Class of bad credit.} \end{cases} \tag{12}$$

O_{model} is the output value of BPN, X_i is the input feature set of the ith sample, and O_i is the output credit decision associated with X_i.

These two credit scoring data sets are partitioned into training and independent test sets by the same 10-fold cross validation procedure used in SVM-based approaches. Therefore, the results are averages of the accuracy rates determined for each of the 10 independent holdout data set partitions (testing accuracy). Since the training of GP and BPN is a stochastic process, the accuracy rate for each data set partition is the average of five repetitions.

The GP specific parameters for these two credit data sets are as follows: population size is 250, reproduction rate is 0.2, crossover rate is 0.7, mutation rate is 0.08, and maximum number of generations is 2000–3000. For the BPN model, several options of the neural network configurations are tested, in which 14-32-1 and 24-43-1 respectively for the Australian data and German data are selected to obtain better results. Additionally, the learning rate and momentum are set to 0.8 and 0.2, respectively. For C4.5, we choose its default settings.

4.3.1. Comparison of the accuracy

The results for the Australian credit data set and German credit data set were obtained by using the three methods of GP, BPN and C4.5 and are summarized in Tables 4 and 5, respectively.

For the Australian data set, there are no significant differences among the SVM (SVM + GA strategy), GP, BPN, and C4.5 based on the Friedman test ($p = 0.37$). For the German data set, however, we found SVM, GP and BPN were identical (no significant differences among these approaches), but C4.5 model was significantly inferior to the other three approaches based on the nonparametric Wilcoxon signed rank tests ($p = 0.04$). That is, for the cases of the German data set, SVM, GP and BPN are more appropriate than C4.5 given the criteria of maximizing prediction accuracy.

Regarding the computation time, however, the CPU time of C4.5 is very short compared to the "SVM + GA," "SVM + Grid + *F*-score," GP and BPN approaches whose average CPU times for running one fold of the Australian data set are 19.3, 15.4, 9.5, and 6.3 min, respectively. The CPU time is based on an IBM compatible PC with an Intel Pentium IV CPU running at 1.6 GHz with 256 MB RAM. The average running time is affected by the software environment. The GP and BPN are developed by using the C language, while the SVM was implemented by the C language—libsvm (Chang & Lin, 2001). Grid search and *F*-score feature selection developed by Chen and Lin (2005) were performed under the Python environment, and the GA evolutionary process was performed under the Matlab environment. Generally, compared with other systems, the running time is much longer when using the Matlab. Basically, we do not intend to compare their running times in this study.

Table 4
Predictive credit accuracy of GP, BPN, C4.5, and "SVM + GA" for the Australian data

	Selected features		Hit rate	
	Avg.	Std.	Avg. (%)	Std. (%)
BPN	–	–	86.83	3.86
GP	8.20	1.60	87.00	4.03
C4.5	12.20	0.80	85.90	3.47
SVM + GA	7.3	1.65	86.90	4.22

Table 5
Results summary with 10-fold cross validation for German credit data set

	Selected features		Hit rate	
	Avg.	Std.	Avg. (%)	Std. (%)
BPN	–	–	77.83	3.21
GP	13.20	2.10	78.10	4.12
C4.5	20.30	1.90	73.60	3.41
SVM + GA	13.3	1.41	77.92	3.97

4.3.2. Comparison of the selected features

A key deficiency of neural network models for credit scoring applications is the difficulty in selecting the discriminative features and explaining the rationale for the credit decision (West, 2000). We apply Garson's index (Garson, 1991) to estimate the relative contributions of input features to the credit class.

After performing the 10-fold cross validation, for each attribute, we calculated its average *F*-score ("SVM + Grid + *F*-score"), average Garson's index (BPN), and the frequency of selected features ("SVM + GA" and BP). Figs. 3 and 4 illustrate the relative importance of each feature with the form of its relative percentage for the two data sets respectively.

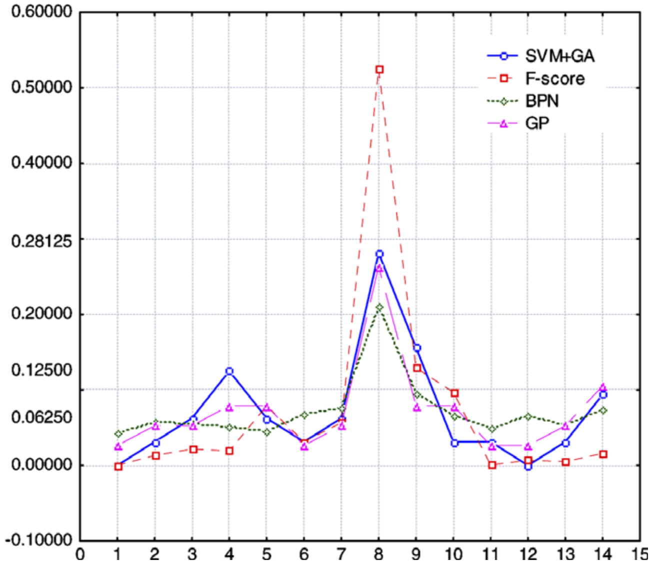

Fig. 3. Relative importance of features for Australian dataset.

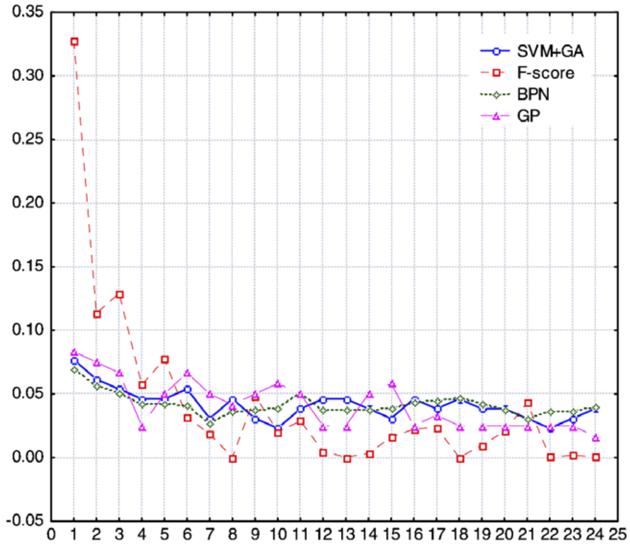

Fig. 4. Relative importance of features for German dataset.

In this study, GP and "SVM + GA" credit scoring models have a smaller feature set. For the case of the Australian data, only about 8 of the 14 input variables are finally present in the credit scoring model; for the results of the German data, only about 13 of the 24 input features appear in the models.

From Fig. 3, in the Australian credit example, the 8th input feature is the most important one to credit classification. Some attributes (e.g., attributes 1, 11, and 12) do not contribute to the "SVM + Grid + *F*-score" model. For GP, BPN, and "SVM + GA" models, all input variables seem to contribute to the output decision variable. For C 4.5, almost all input features evenly contribute to the credit decision, but for simplicity, this result is not shown in Figs. 3 and 4.

In the case of the German data set, for *F*-score, some attributes are not selected (e.g., attributes 8, 13, 18, 22, 23, and 24), but all input features contribute to the credit decision for all other models.

5. Conclusions

Credit scoring is a widely used technique that helps banks decide whether to grant credit to consumers who submit an application. Constructing the credit scoring models from a credit database can be taken as a task of data mining. The statistical classification models perform favorably only when the essential assumptions are satisfied. In contrast to traditional statistical techniques, the artificial intelligence techniques (such as SVM, GP, BPN or decision tree) do not require the knowledge of the underlying relationships between input and output variables.

This paper investigates the three strategies of the SVM credit scoring models and benchmarks their performance against neural network, genetic programming, and C4.5 models under concern for commercial applications. We make the following conclusions:

(1) The SVM-based approach credit scoring model can properly classify the applications as either accepted or rejected, thereby minimizing the creditors' risk and translating considerably into future savings.
(2) It is evident that the SVM-based model is very competitive to BPN and GP in terms of classification accuracy. Compared with GP and BPN, SVM-based credit scoring model can achieve identical classificatory accuracy.
(3) The SVM-based models also have similar accuracies reported in the literature. Ong et al. (2005) reported that the accuracies of GP, BPN and C4.5 are 88.27%, 87.93%, and 87.06%, respectively, for the Australian data set and are 77.34%, 75.51%, and 73.17%, respectively, for the German data set.

To adopt the SVM-based credit scoring model, this study recommend combining SVM with a mechanism to search the optimal model parameters and feature subset. *F*-score is a simple way to determine important features, but it does not reveal mutual information among features (Chen & Lin, 2005). According to our study, a hybrid SVM-GA system is a good alternative for optimizing the parameters and feature subset. With a small feature subset, a hybrid SVM-GA system can obtain a good classification performance. However, when using SVM-GA strategy (as well as GP and BPN), one should avoid over-training. This study recommends using a separate validation set to tune the model parameters and determine appropriate training iterations.

The drawback of the SVM-GA (as well as GP-based) credit scoring model is its long training time. It is a well-known fact that many applications of KDD require the capability of efficient processing of large databases. In such cases, algorithms that offer very good classification accuracy at the cost of high computational complexity cannot be applied. Fortunately, GA-based systems are well suited for parallel architecture. Another practical obstacle of the SVM-based (as well as neural networks) credit scoring model is its black-box nature. A possible solution for this issue is the use of SVM rule extraction techniques or the use of hybrid-SVM model combining with other more interpretable models.

References

Baesens, B., Van Gestel, T., Viaene, S., Stepanova, M., Suykens, J., & Vanthienen, J. (2003). Benchmarking state-of-the-art classification algorithms for credit scoring. *Journal of the Operational Research Society, 54*(6), 627–635.

Brill, J. (1998). The importance of credit scoring models in improving cash flow and collection. *Business Credit, 100*(1), 16–17.

Chang, C. C., & Lin, C. J. (2001) LIBSVM: a library for support vector machines. Available from http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chen, M. C., & Huang, S. H. (2003). Credit scoring and rejected instances reassigning through evolutionary computation techniques. *Expert Systems with Applications, 24*(4), 433–441.

Chen, S. Y., & Liu, X. (2004). The contribution of data mining to information science. *Journal of Information Science, 30*(6), 550–558.

Chen, Y.-W., & Lin, C.-J. (2005). Combining SVMs with various feature selection strategies. Available from http://www.csie.ntu.edu.tw/~cjlin/papers/features.pdf.

Cristianini, N., & Shawe-Taylor, J. (2000). *An introduction to support vector machines*. Cambridge: Cambridge University Press.

Davis, R. H., Edelman, D. B., & Gammerman, A. J. (1992). Machine learning algorithms for credit-card applications. *Journal of Mathematics Applied in Business and Industry, 4*, 43–51.

Desai, V. S., Crook, J. N., & Overstreet, G. A. (1996). A comparison of neural networks and linear scoring models in the credit union environment. *European Journal of Operational Research, 95*(1), 24–37.

Fröhlich, H., & Chapelle, O. (2003). Feature selection for support vector machines by means of genetic algorithms. In *Proceedings of the 15th IEEE international conference on tools with artificial intelligence, Sacramento, California, USA*, pp. 142–148.

Garson, G. D. (1991). Interpreting neural-network connection weights. *AI Expert, 6*(4), 47–51.

Goldberg, D. E. (1989). *Genetic algorithms in search optimization and machine learning*. MA: Addison-Wesley.

Guyon, I., Weston, J., Barnhill, S., & Bapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine Learning, 46*(1–3), 389–422.

Henley, W. E. (1995). Statistical aspects of credit scoring. Dissertation, The Open University, Milton Keynes, UK.

Henley, W. E., & Hand, D. J. (1996). A *k*-nearest neighbor classifier for assessing consumer credit risk. *Statistician, 44*(1), 77–95.

Hoffmann, F., Baesens, B., Martens, J., Put, F., & Vanthienen, J. (2002). Comparing a genetic fuzzy and a neurofuzzy classifier for credit scoring. *International Journal of Intelligent Systems, 17*(11), 1067–1083.

Holland, J. H. (1975). *Adaption in natural and artificial systems*. Ann Arbor, MI: The University of Michigan Press.

Hsieh, N.-C. (2005). Hybrid mining approach in the design of credit scoring models. *Expert Systems with Applications, 28*(4), 655–665.

Hsu, C. W., Chang, C. C., & Lin, C. J. (2003). A practical guide to support vector classification. Available from http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

Hsu, C. W., & Lin, C. J. (2002). A simple decomposition method for support vector machine. *Machine Learning, 46*(1–3), 219–314.

Huang, Z., Chen, H., Hsu, C.-J., Chen, W.-H., & Wu, S. (2004). Credit rating analysis with support vector machines and neural networks: a market comparative study. *Decision Support Systems, 37*(4), 543–558.

Joachims, T. (1998). Text categorization with support vector machines. In *Proceedings of European conference on machine learning (ECML), Chemintz, DE*, pp.137–142.

John, G., Kohavi, R., & Peger, K. (1994). Irrelevant features and the subset selection problem. In *Proceedings of the eleventh international conference on machine learning, San Mateo, CA, USA*, pp. 121–129.

Kecman, V. (2001). *Learning and soft computing*. Cambridge, MA: The MIT Press.

Kohavi, R., & John, G. (1997). Wrappers for feature subset selection. *Artificial Intelligence, 97*(1–2), 273–324.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA: The MIT Press.

Lee, T.-S., & Chen, I.-F. (2005). A two-stage hybrid credit scoring model using artificial neural networks and multivariate adaptive regression splines. *Expert Systems with Applications, 28*(4), 743–752.

Lee, T.-S., Chiu, C.-C., Lu, C.-J., & Chen, I.-F. (2002). Credit scoring using the hybrid neural discriminant technique. *Expert Systems with Applications, 23*(3), 245–254.

Liu, H., & Motoda, H. (1998). *Feature selection for knowledge discovery and data mining*. Norwell, MA: Kluwer Academic.

Malhotra, R., & Malhotra, D. K. (2002). Differentiating between good credits and bad credits using neuro-fuzzy systems. *European Journal of Operational Research, 136*(1), 190–211.

Mao, K. Z. (2004). Feature subset selection for support vector machines through discriminative function pruning analysis. *IEEE Transactions on Systems, Man, and Cybernetics, 34*(1), 60–67.

Murphy, P. M., Aha, D. W. (2001). UCI repository of machine learning databases. Department of Information and Computer Science, University of California Irvine, CA. Available from http://www.ics.uci.edu/mlearn/MLRepository.htmlurlhttp://www.ics.uci.edu/mlearn/MLRepository.html.

Ong, C.-S., Huang, J.-J., & Tzeng, G.-H. (2005). Building credit scoring models using genetic programming. *Expert Systems with Applications, 29*(1), 41–47.

Pontil, M., & Verri, A. (1998). Support vector machines for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(6), 637–646.

Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning, 1*(1), 81–106.

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Francisco, CA: Morgan Kaufman.

Reichert, A. K., Cho, C. C., & Wagner, G. M. (1983). An examination of the conceptual issues involved in developing credit-scoring models. *Journal of Business and Economic Statistics, 1*(2), 101–114.

Salzberg, S. L. (1997). On comparing classifiers: pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery, 1*, 317–327.

Scho̎lkopf, B., & Smola, A. J. (2000). *Statistical learning and kernel methods*. Cambridge, MA: MIT Press.

Somol, P., Baesens, B., Pudil, P., & Vanthienen, J. (2005). Filter-versus wrapper-based feature selection for credit scoring. *International Journal of Intelligent Systems, 20*(10), 985–999.

Tam, K. Y., & Kiang, M. Y. (1992). Managerial applications of the neural networks: the case of bank failure prediction. *Management Science, 38*(7), 926–947.

Thomas, L. C. (2000). A survey of credit and behavioural scoring: forecasting financial risk of lending to consumers. *International Journal of Forecasting, 16*(2), 149–172.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer-Verlag.

West, D. (2000). Neural network credit scoring models. *Computers and Operations Research, 27*(11–12), 1131–1152.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., & Vapnik, V. (2001). Feature selection for SVM. In S. A. Solla, T. K. Leen, & K.-R. Muller (Eds.). *Advances in neural information processing systems* (Vol. 13, pp. 668–674). Cambridge, MA: MIT Press.

Yu, G. X., Ostrouchov, G., Geist, A., & Samatova, N. F. (2003). An SVM-based algorithm for identification of photosynthesis-specific genome features. In *2nd IEEE computer society bioinformatics conference, CA, USA*, pp. 235–243.

Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, 30*(4), 451–462.