Available online at www.sciencedirect.com
ScienceDirect
Expert Systems with Applications 33 (2007) 67–74
Expert Systems with Applications
www.elsevier.com/locate/eswa

# Application of support vector machines to corporate credit rating prediction

Young-Chan Lee *

*College of Commerce and Economics, Dongguk University, Gyeongju, Gyeongbuk 780-714, South Korea*

## Abstract

Corporate credit rating analysis has drawn a lot of research interests in previous studies, and recent studies have shown that machine learning techniques achieved better performance than traditional statistical ones. This paper applies support vector machines (SVMs) to the corporate credit rating problem in an attempt to suggest a new model with better explanatory power and stability. To serve this purpose, the researcher uses a grid-search technique using 5-fold cross-validation to find out the optimal parameter values of RBF kernel function of SVM. In addition, to evaluate the prediction accuracy of SVM, the researcher compares its performance with those of multiple discriminant analysis (MDA), case-based reasoning (CBR), and three-layer fully connected back-propagation neural networks (BPNs). The experiment results show that SVM outperforms the other methods.

*Keywords:* Credit rating; SVM; BPN; MDA; CBR

## 1. Introduction

Credit ratings have been extensively used by bond investors, debt issuers, and governmental officials as a surrogate measure of riskiness of the companies and bonds. They are important determinants of risk premiums and even the marketability of bonds (Huang, Chen, Hsu, Chen, & Wu, 2004). The development of the corporate credit rating prediction model has attracted lots of research interests in academic and business community. Although of interests in accurate quantitative prediction of corporate bond rating, due to lack of scientific credit rating methodology and sufficient data accumulation to construct the model, the traditional approach produce an internal rating on the basis of credit officer's judgment to a significant extent in the real world (Shin & Han, 2001).

Several studies used statistical methods, including regression, multi-variate discriminant analysis, probit and logit models to predict bond rating (Altman & Katz, 1976; Ang & Patel, 1975; Baran, Lakonishok, & Ofer, 1980; Belkaoui, 1980; Bhandari, Soldofsky, & Boe, 1979; Martin, Henderson, Perry, & Cronan, 1984). McAdams (1980) employs the use of multiple discriminant analysis to design a statistical credit analysis model to assist portfolio managers to predict agency downgrades of electric utility bonds. Horrigan (1966) and Pogue and Soldofsky (1969) use multiple regression model to predict Moody's findings. Pinches and Mingo (1973) use factor analysis to screen variables for predicting bond ratings and then apply multiple discriminant analysis. Kamstra, Kennedy, and Suan (2001) improve the statistical predictive model by combining several forecasting methods to predict bond ratings in the transportation and industrial sectors. They use ordered logit method to combine forecasts and they find that combined forecasts outperform their input forecasts (Kim, 2005). Recently artificial intelligence approaches such as inductive learning (Shaw & Gentry, 1990), artificial neural networks (Dutta & Shekhar, 1996; Kim, 1992; Kwon, Han, & Lee, 1997; Maher & Sen, 1997; Moody & Utans, 1995; Singleton & Surkan, 1995), and case-based reasoning (Butta, 1994; Kim & Han, 2001; Shin & Han,

* Tel.: +82 54 770 2317; fax: +82 54 770 2476.
*E-mail address:* chanlee@dongguk.ac.kr

0957-4174/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.eswa.2006.04.018

1999; Shin & Han, 2001) have been applied to bond rating. Although artificial intelligence approaches have several advantages over statistical methods (Salchenberger, Cinar, & Las, 1992; Tam & Kiang, 1992), the results of these studies were less than expected because the real data in application is usually unevenly distributed among classes and these approaches are limited in dealing with the ordinal nature of bond rating.

The purpose of this study is to apply support vector machines (SVMs), a relatively new machine learning technique, to corporate credit rating prediction problem and to provide a new model improving its prediction accuracy. Developed by Vapnik (1998), SVM is gaining popularity due to many attractive features and excellent generalization performance on a wide range of problems. In addition, bearing in mind that the optimal parameter search plays a crucial role to build a credit rating prediction model with high prediction accuracy and stability, this study employs a grid-search technique using 5-fold cross-validation to find out the optimal parameter values of RBF kernel function of SVM. To evaluate the prediction accuracy of SVM, this study also compares its performance with those of multiple discriminant analysis (MDA), case-based reasoning (CBR), and three-layer fully connected back-propagation neural networks (BPNs).

## 2. Support vector machines

Support vector machines (SVMs) use a linear model to implement nonlinear class boundaries through some nonlinear mapping input vectors into a high-dimensional feature space. The linear model constructed in the new space can represent a nonlinear decision boundary in the original space. In the new space, an optimal separating hyperplane (OSH) is constructed. Thus, SVM is known as the algorithm that finds a special kind of linear model, the *maximum margin hyperplane*. The maximum margin hyperplane gives the maximum separation between decision classes. The training examples that are closest to the maximum margin hyperplane are called *support vectors*. All other training examples are irrelevant for defining the binary class boundaries (Cristianini & Shawe-Taylor, 2000; Gunn, 1998; Hearst, Dumais, Osman, Platt, & Scholkopf, 1998; Vapnik, 1998).

SVM is simple enough to be analyzed mathematically since it can be shown to correspond to a linear method in a high dimensional feature space nonlinearly related to input space. In this sense, SVM may serve as a promising alternative combining the strengths of conventional statistical methods that are more theory-driven and easy to analyze, and more data-driven, distribution-free and robust machine learning methods. Recently, the SVM approach has been introduced to several financial applications such as credit rating, time series prediction, and insurance claim fraud detection (Fan & Palaniswami, 2000; Gestel et al., 2001; Huang et al., 2004; Kim, 2003; Tay & Cao, 2001; Viaene, Derrig, Baesens, & Dedene, 2002). These studies reported that SVM was comparable to and even outperformed other classifiers including ANN, CBR, MDA, and Logit in terms of generalization performance. Motivated by these previous researches, this study applies SVM to the domain of corporate credit rating prediction, and compare its prediction performance with those of MDA, CBR, and BPNs.

A simple description of the SVM algorithm is provided as follows. Given a training set $D = \{x_i, y_i\}_{i=1}^{N}$ with input vectors $x_i = (x_i^{(1)}, \ldots, x_i^{(n)})^{\mathrm{T}} \in \mathbb{R}^n$ and target labels $y_i \in \{-1, +1\}$, the support vector machine (SVM) classifier, according to Vapnik's original formulation, satisfies the following conditions:

$$\begin{cases} \mathbf{w}^{\mathrm{T}}\phi(x_i) + b \geqslant +1, & \text{if } y_i = +1 \\ \mathbf{w}^{\mathrm{T}}\phi(x_i) + b \leqslant -1, & \text{if } y_i = -1 \end{cases} \tag{1}$$

which is equivalent to

$$y_i[\mathbf{w}^{\mathrm{T}}\phi(x_i) + b] \geqslant 1, \quad i = 1, \ldots, N \tag{2}$$

where $\mathbf{w}$ represents the weight vector and $b$ the bias. Nonlinear function $\phi(\cdot) : \mathbb{R}^n \to \mathbb{R}^{n_k}$ maps input or measurement space to a high-dimensional, and possibly infinite-dimensional, feature space. Eq. (2) then comes down to the construction of two parallel bounding hyperplanes at opposite sides of a separating hyperplane $\mathbf{w}^{\mathrm{T}}\phi(x) + b = 0$ in the feature space with the margin width between both hyperplanes equal to $\frac{2}{\|\mathbf{w}\|^2}$. In primal weight space, the classifier then takes the decision function form (3):

$$\operatorname{sgn}(\mathbf{w}^{\mathrm{T}}\phi(x) + b) \tag{3}$$

Most of classification problems are, however, linearly non-separable. Therefore, it is general to find the weight vector using slack variable ($\xi_i$) to permit misclassification. One defines the primal optimization problem as

$$\underset{\mathbf{w},b,\xi}{\text{Min}} \qquad \frac{1}{2}\mathbf{w}^{\mathrm{T}}\mathbf{w} + C\sum_{i=1}^{N}\xi_i \tag{4}$$

$$\text{Subject to} \quad \begin{cases} y_i(\mathbf{w}^{\mathrm{T}}\phi(x_i) + b) \geqslant 1 - \xi_i, & i = 1, \ldots, N \\ \xi_i \geqslant 0, & i = 1, \ldots, N \end{cases} \tag{5}$$

where $\xi_i$'s are slack variables needed to allow misclassifications in the set of inequalities, and $C \in \mathbb{R}^+$ is a tuning hyperparameter, weighting the importance of classification errors vis-à-vis the margin width. The solution of the primal problem is obtained after constructing the Lagrangian. From the conditions of optimality, one obtains a quadratic programming (QP) problem with Lagrange multipliers $\alpha_i$'s. A multiplier $\alpha_i$ exists for each training data instance. Data instances corresponding to non-zero $\alpha_i$'s are called *support vectors*.

On the other hand, the above primal problem can be converted into the following dual problem with objective function (6) and constraints (7). Since the decision variables are support vector of Lagrange multipliers, it is easier to interpret the results of this dual problem than those of the primal one.

$$\underset{\boldsymbol{\alpha}}{\text{Max}} \quad \frac{1}{2}\boldsymbol{\alpha}^{\mathrm{T}}Q\boldsymbol{\alpha} - \mathbf{e}^{\mathrm{T}}\boldsymbol{\alpha} \tag{6}$$

$$\text{Subject to} \quad \begin{cases} 0 \leqslant \alpha_i \leqslant C, \quad i = 1, \ldots, N \\ \mathbf{y}^{\mathrm{T}}\boldsymbol{\alpha} = 0 \end{cases} \tag{7}$$

In the dual problem above, **e** is the vector of all ones, $Q$ is a $N \times N$ positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K(x_i, x_j)$, and $K(x_i, x_j) \equiv \phi(x_i)^{\mathrm{T}}\phi(x_j)$ is the kernel. Here, training vectors $x_i$'s are mapped into a higher (maybe infinite) dimensional space by function $\phi$. As is typical for SVMs, we never calculate **w** or $\phi(x)$. This is made possible due to Mercer's condition, which relates mapping function $\phi(x)$ to kernel function $K(\cdot,\cdot)$ as follows:

$$K(x_i, x_j) = \phi(x_i)^{\mathrm{T}}\phi(x_j) \tag{8}$$

For kernel function $K(\cdot,\cdot)$, one typically has several design choices such as the linear kernel of $K(x_i, x_j) = x_i^{\mathrm{T}}x_j$, the polynomial kernel of degree $d$ of $K(x_i, x_j) = (\gamma x_i^{\mathrm{T}}x_j + r)^d$, $\gamma > 0$, the radial basis function (RBF) kernel of $K(x_i, x_j) = \exp\{-\gamma\|x_i - x_j\|^2\}$, $\gamma > 0$, and the sigmoid kernel of $K(x_i, x_j) = \tanh\{\gamma x_i^{\mathrm{T}}x_j + r\}$, where $d, r \in \mathbb{N}$ and $\gamma \in \mathbb{R}^+$ are constants. Then one constructs the final SVM classifier as

$$\operatorname{sgn}\left(\sum_i^N \alpha_i y_i K(x, x_i) + b\right) \tag{9}$$

The details of the optimization are discussed in (Chang & Lin, 2004; Cristianini & Shawe-Taylor, 2000; Gunn, 1998; Vapnik, 1998).

## 3. Research design

### 3.1. Data collection and preprocessing

The database used in this study was obtained from the Korea Information Service, Inc. that is one of the most prominent bond rating agencies in Korea. The database consists of 297 financial ratios and the corresponding bond rating of 3017 Korean companies whose commercial papers have been rated from 1997 to 2002. Credit grades are classified as five coarser rating categories (AAA, AA, A, B, C) according to credit levels. Table 1 shows the organization of the data set.

The original data are scaled into the range of $[-1, 1]$. The goal of linear scaling is to independently normalize each feature component to the specified range. It ensures the larger value input attributes do not overwhelm smaller value inputs; hence helps to reduce prediction errors (Hsu, Chang, & Lin, 2004).

Table 1
Number of companies in each rating

| Ratings | Number of companies | % |
|---|---|---|
| AAA | 203 | 6.8 |
| AA | 629 | 20.9 |
| A | 993 | 32.9 |
| B | 1057 | 34.9 |
| C | 135 | 4.5 |
| Total | 3017 | 100.0 |

Table 2
The selected financial variables

| Variables | Description |
|---|---|
| X1 | Ordinary income to total assets |
| X2 | Interest coverage ratio |
| X3 | EBIT to interest |
| X4 | Net income to stakeholders' equity |
| X5 | Stakeholders' equity to total assets |
| X6 | Current liabilities ratio |
| X7 | Fixed ratio |
| X8 | Interest expenses to sales |
| X9 | Debt ratio |
| X10 | Receivables to payables |

In choosing financial ratios, this study applies multiple discriminant analysis (MDA) stepwise regression analysis. Most previous studies using statistical methods such as discriminant analysis and logistic regression have selected independent variables through the stepwise regression analysis. In this study, several financial ratios are initially selected by the one-way ANOVA. Using a MDA stepwise method, this study also reduces the number of financial variables to a manageable set of 10. The selected variables for this study are shown in Table 2.

In cases of SVM, MDA, and CBR, each data set is split into two subsets: a training set of 80% (2413) and a holdout set of 20% (604) of the total data (3017) respectively. The holdout data is used to test the results, which is not utilized to develop the model. In case of BPN, each data set is split into three subsets: a training set of 60% (1809), a validation set of 20% (604), and a holdout set of 20% (604) of the total data (3017) respectively, where the validation data is used to check the results.

### 3.2. Support vector machines

In this study, the radial basis function (RBF) is used as the basic kernel function of SVM. There are two parameters associated with RBF kernels: $C$ and $\gamma$. The upper bound $C$ and the kernel parameter $\gamma$ play a crucial role in the performance of SVMs (Hsu et al., 2004; Tay & Cao, 2001). Nevertheless, there is little general guidance to determine the parameter values of SVM. Recently, Hsu et al. (2004) suggested a practical guideline to SVM using grid-search and cross-validation, and this study will utilize it.

The goal is to identify optimal choice of $C$ and $\gamma$ so that the classifier can accurately predict unknown data (i.e., holdout data). Note that it may not be useful to achieve high training accuracy (i.e., classifiers accurately predict training data whose class labels are indeed known). Therefore, a common way is to separate training data into two parts, of which one is considered unknown in training the classifier. Then the prediction accuracy on this set can more precisely reflect the performance on classifying unknown data. An improved version of this procedure is cross-validation.

In $v$-fold cross-validation, this study first divides the training set into $v$ subsets of equal size. Sequentially one subset is tested using the classifier trained on the remaining $(v-1)$ subsets. Thus, each instance of the whole training set is predicted once so the cross-validation accuracy is the percentage of data that are correctly classified.

This study uses a grid-search on $C$ and $\gamma$ using 5-fold cross-validation. Basically, all the pairs of $(C,\gamma)$ are tried and the one with the best cross-validation accuracy is selected. This study realizes that trying exponentially growing sequences of $C$ and $\gamma$ is a practical method to identify optimal parameters (for example, $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$, $\gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$).

In this study, *LIBSVM* software system (Chang & Lin, 2004) is used to perform SVM experiments.

### 3.3. Back-propagation neural networks

In this study, MDA, CBR and a three-layer fully connected back-propagation neural network (BPN) are used as benchmarks. In BPN, this study varies the number of nodes in the hidden layer and stopping criteria for training. In particular, 10, 15, 20, 25 hidden nodes are used for each stopping criterion because BPN does not have a general rule for determining the optimal number of hidden nodes (Kim, 2003). For the stopping criteria of BPN, this study allows 1000, 2000, 3000 learning epochs per one training example since there is little general knowledge for selecting the number of epochs. The learning rate is set to 0.1 and the momentum term is to 0.6. The hidden nodes use the sigmoid transfer function and the output node uses the same transfer function. This study uses *XLMiner 2.4.1* to perform the BPN experiments.

### 3.4. Multiple discriminant analysis

Multiple discriminant analysis (MDA) tries to derive a linear combination of two or more independent variables that best discriminates among a priori defined groups, which in our case are bankruptcy and non-bankruptcy companies. This is achieved by the statistical decision rule of maximizing the between-group variance relative to the within-group variance. This relationship is expressed as the ratio of the between-group to the within-group variance. The MDA derives the linear combinations from an equation that takes the following form:

$$Z = w_1x_1 + w_2x_2 + \cdots + w_nx_n \tag{10}$$

where $Z$ is a discriminant score, $w_i (i = 1,2,\ldots,n)$ are discriminant weights, and $x_i (i = 1,2,\ldots,n)$ are independent variables, the financial ratios. Thus, each firm receives a single composite discriminant score which is then compared to a cut-off value, and with this information, we can determine to which group the firm belongs.

MDA does very well provided that the variables in every group follow a multivariate normal distribution and the covariance matrix for every group is equal. However, empirical experiments have shown that especially failed firms violate the normality condition.[1] In addition, the equal group variance condition is often violated. Moreover, multi-collinearity among independent variables may cause a serious problem, especially when the stepwise procedures are employed (Hair, Anderson, Tatham, & Black, 1998).

### 3.5. Case-based reasoning

Case-based reasoning (CBR) is a problem solving technique by re-using past cases and experiences to find a solution to the given new problems. This study uses a nearest neighbor method to extract similar cases in CBR. The nearest neighbor method can be easily used to numerical data such as financial ratios. This study varies the number of nearest neighbor from 1 to 10. To extract similarity of cases with a nearest neighbor method, this study uses Euclidian distance as follows:

$$D_{ab} = \sqrt{\sum_{i=1}^{n} w_i \times (f_{ai} - f_{bi})^2} \tag{11}$$

where $D_{ab}$ is a distance between $f_{ai}$ and $f_{bi}$ that are values of attribute $f_{\cdot i}$ of input $a$ and retrieved case $b$, $n$ is a number of attribute, and $w_i$ is a weight of attribute $f_{\cdot i}$.

## 4. Empirical analysis

### 4.1. SVM models

In SVM, each data set is split into two subsets: a training set of 80% (2413) and a holdout set of 20% (604) of the total data (3017) respectively.

We must first decide which kernels to select for implementing SVM; and then the penalty parameter $C$ and kernel parameters are chosen. One of the advantages of the linear kernel SVM is that there are no parameters to tune except for constant $C$. But the upper bound $C$ on coefficient $\alpha_i$ affects the prediction performance for the cases where the training data is not separable by a linear SVM (Drucker, Wu, & Vapnik, 1999). For the nonlinear SVM, there is an additional parameter, the kernel parameter, to tune. There are three kernel functions for nonlinear SVM including the radial basis function (RBF), the polynomial, and the sigmoid.

The RBF kernel nonlinearly maps the samples into a higher dimensional space unlike the linear kernel, so it can handle the case when the relation between class labels and attributes is nonlinear. The sigmoid kernel behaves like

[1] Nevertheless, empirical studies have shown that the problems concerning normality assumptions do not weaken its classification capability, but its prediction ability.

the RBF for certain parameters; however, it is not valid under some parameters (Vapnik, 1998). The polynomial function takes a longer time in the training stage of SVM, and it is reported to provide worse results than the RBF function in the previous studies (Huang et al., 2004; Kim, 2003; Tay & Cao, 2001). In addition, the polynomial kernel has more hyperparameters than the RBF kernel and may go to infinity or zero while the degree is large. Thus, this study uses the RBF kernel SVM as the default model.

There are two parameters associated with the RBF kernels: $C$ and $\gamma$. It is not known beforehand which values of $C$ and $\gamma$ are the best for one problem; consequently, some kind of model selection (parameter search) approach must be employed (Hsu et al., 2004). This study conducts a grid-search to find the best values of $C$ and $\gamma$ using 5-fold cross-validation. Pairs of $(C,\gamma)$ are tried and the one with the best cross-validation accuracy is picked. After conducting the grid-search for training data, we found that the optimal $(C,\gamma)$ was $(2^{11},2^{-3})$ with the cross-validation rate of 67.55% (see Fig. 1). Table 3 summarizes the results of the grid-search using 5-fold cross-validation.

After the optimal $(C,\gamma)$ was found, the whole training data was trained again to generate the final classifier. The overall classification accuracy of the holdout data turned out to be 67.22%, where the prediction accuracy of the training data was 77.62%. The classification confusion matrix is shown in Table 4.

As shown in Table 4, the classification accuracy of each class is acceptable level except 5 (rating: C).

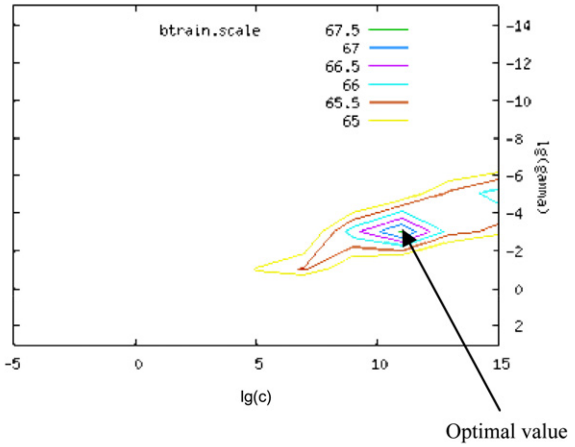

Fig. 1. Grid-search on $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}$ and $\gamma = 2^{-15}, 2^{-13}, \ldots, 2$.

Table 4
The classification confusion matrix of SVM

| Actual | Predicted | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | Total |
| *Training set* | | | | | | |
| 1 | 137 | 24 | 1 | 0 | 0 | 162 |
| 2 | 17 | 368 | 114 | 4 | 0 | 503 |
| 3 | 1 | 59 | 599 | 135 | 0 | 794 |
| 4 | 0 | 8 | 124 | 706 | 8 | 846 |
| 5 | 0 | 0 | 1 | 44 | 63 | 108 |
| 1 | 84.57% | 14.81% | 0.62% | 0.00% | 0.00% | 100 |
| 2 | 3.38% | 73.16% | 22.66% | 0.80% | 0.00% | 100 |
| 3 | 0.13% | 7.43% | 75.44% | 17.00% | 0.00% | 100 |
| 4 | 0.00% | 0.95% | 14.66% | 83.45% | 0.95% | 100 |
| 5 | 0.00% | 0.00% | 0.93% | 40.74% | 58.33% | 100 |
| *Holdout set* | | | | | | |
| 1 | 28 | 13 | 0 | 0 | 0 | 41 |
| 2 | 10 | 78 | 37 | 1 | 0 | 126 |
| 3 | 4 | 31 | 139 | 25 | 0 | 199 |
| 4 | 1 | 2 | 53 | 150 | 5 | 211 |
| 5 | 0 | 0 | 0 | 16 | 11 | 27 |
| 1 | 68.29% | 31.71% | 0.00% | 0.00% | 0.00% | 100 |
| 2 | 7.94% | 61.90% | 29.37% | 0.79% | 0.00% | 100 |
| 3 | 2.01% | 15.58% | 69.85% | 12.56% | 0.00% | 100 |
| 4 | 0.47% | 0.95% | 25.12% | 71.09% | 2.37% | 100 |
| 5 | 0.00% | 0.00% | 0.00% | 59.26% | 40.74% | 100 |

## 4.2. BPN models

In BPN, each data set is split into three subsets: a training set of 60% (1809), a holdout set of 20% (604), and a validation set of 20% (604) of the total data (3017) respectively. The results of three-layer BPN according to parameter adjustment are summarized in Table 5.

Table 3
The result of grid-search using 5-fold cross-validation

| $C$ | $\gamma$ | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | $2^{3}$ | $2^{1}$ | $2^{-1}$ | $2^{-3}$ | $2^{-5}$ | $2^{-7}$ | $2^{-9}$ | $2^{-11}$ | $2^{-13}$ | $2^{-15}$ |
| $2^{-5}$ | 35.06 | 47.00 | 51.31 | 49.81 | 44.26 | 35.06 | 35.06 | 35.06 | 35.06 | 35.06 |
| $2^{-3}$ | 47.29 | 55.66 | 56.69 | 54.58 | 51.06 | 47.00 | 35.06 | 35.06 | 35.06 | 35.06 |
| $2^{-1}$ | 57.61 | 58.81 | 58.77 | 58.35 | 54.70 | 51.22 | 47.29 | 35.06 | 35.06 | 35.06 |
| $2^{1}$ | 60.67 | 60.63 | 60.30 | 59.35 | 58.14 | 54.62 | 51.43 | 47.29 | 35.06 | 35.06 |
| $2^{3}$ | 60.13 | 63.03 | 62.95 | 60.88 | 59.68 | 57.85 | 54.62 | 51.47 | 47.33 | 35.06 |
| $2^{5}$ | 58.60 | 62.70 | *65.11* | 62.00 | 61.33 | 59.55 | 57.77 | 54.58 | 51.43 | 47.33 |
| $2^{7}$ | 58.31 | 61.29 | *65.56* | *64.19* | 61.75 | 60.63 | 59.76 | 57.85 | 54.66 | 51.43 |
| $2^{9}$ | 58.06 | 59.30 | *64.36* | *66.31* | *63.49* | 61.71 | 60.55 | 59.35 | 57.90 | 54.70 |
| $2^{11}$ | 57.90 | 58.56 | *63.53* | **67.55** | *64.48* | 62.70 | 61.92 | 60.59 | 59.72 | 57.90 |
| $2^{13}$ | 57.85 | 58.23 | 62.66 | *65.77* | *65.60* | *63.82* | 62.74 | 62.58 | 60.59 | 59.72 |
| $2^{15}$ | 57.85 | 58.06 | 61.13 | *65.31* | *66.27* | *64.19* | 63.37 | 63.24 | 62.70 | 60.42 |

Table 5
The performance of BPN

| Learning epoch | Hidden nodes | Classification accuracy (%) | |
|---|---|---|---|
| | | Training set | Holdout set |
| 1000 | 10 | 60.59 | 57.45 |
| | 15 | 62.54 | 57.95 |
| | 20 | 65.64 | 56.29 |
| | 25 | 66.89 | 55.13 |
| 2000 | 10 | 62.45 | 59.27 |
| | 15 | 64.53 | 57.62 |
| | 20 | 68.50 | 57.62 |
| | 25 | 68.84 | 58.61 |
| 3000 | 10 | 62.95 | **59.93** |
| | 15 | 64.90 | 57.45 |
| | 20 | **69.75** | 58.61 |
| | 25 | 72.19 | 58.77 |

Table 6
The classification confusion matrix of BPN

| Actual | Predicted | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | Total |
| 1 | 21 | 17 | 3 | 0 | 0 | 41 |
| 2 | 14 | 39 | 70 | 3 | 0 | 126 |
| 3 | 12 | 14 | 109 | 64 | 0 | 199 |
| 4 | 2 | 2 | 23 | 177 | 7 | 211 |
| 5 | 0 | 0 | 0 | 15 | 12 | 27 |
| 1 | 51.22% | 41.46% | 7.32% | 0.00% | 0.00% | 100 |
| 2 | 11.11% | 30.95% | 55.56% | 2.38% | 0.00% | 100 |
| 3 | 6.03% | 7.04% | 54.77% | 32.16% | 0.00% | 100 |
| 4 | 0.95% | 0.95% | 10.90% | 83.89% | 3.32% | 100 |
| 5 | 0.00% | 0.00% | 0.00% | 55.56% | 44.44% | 100 |

We can see that the classification accuracy of training data tends to be higher as the learning epoch increases. The best classification accuracy for the holdout data was found when the epoch was 3000 and the number of hidden nodes was 10. The classification accuracy of the holdout data turned out to be 59.93%, and that of the training data was 62.95%.

As shown in Table 5, the best classification accuracy of BPN for both training and holdout data is lower than that of SVM. The classification confusion matrix of BPN is shown in Table 6.

From the results of the empirical experiment, we can conclude that SVM shows better performance than BPN in corporate credit rating prediction while avoiding overfitting problem and exhaustive parameter search.

### 4.3. Multiple discriminant analysis

In MDA, each data set is split into two subsets: a training set of 80% and a holdout set of 20% of the total data (3017) respectively. The overall classification accuracy of the holdout data turned out to be 58.77%, where the prediction accuracy of the training data was 58.72%. The classification confusion matrix is shown in Table 7.

Table 7
The classification confusion matrix of MDA

| Actual | Predicted | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | Total |
| 1 | 22 | 17 | 3 | 0 | 0 | 41 |
| 2 | 8 | 83 | 32 | 3 | 0 | 126 |
| 3 | 2 | 63 | 105 | 29 | 0 | 199 |
| 4 | 1 | 7 | 57 | 129 | 17 | 211 |
| 5 | 0 | 0 | 0 | 11 | 16 | 27 |
| 1 | 52.38% | 40.48% | 7.14% | 0.00% | 0.00% | 100 |
| 2 | 6.35% | 61.87% | 25.40% | 2.38% | 0.00% | 100 |
| 3 | 1.01% | 31.66% | 52.76% | 14.57% | 0.00% | 100 |
| 4 | 0.47% | 3.32% | 27.01% | 61.14% | 8.06% | 100 |
| 5 | 0.00% | 0.00% | 0.00% | 40.74% | 59.26% | 100 |

### 4.4. Case-based reasoning

In CBR, each data set is split into two subsets: a training set of 80% and a holdout set of 20% of the total data (3017) respectively. The overall classification accuracy of the holdout data turned out to be 63.41%. The classification confusion matrix is shown in Table 8.

### 4.5. Classification accuracy comparisons

Table 9 compares the best classification accuracy of SVM, BPN, MDA, and CBR in training and holdout data, and shows that SVM outperforms BPN, MDA and CBR by 7.29%, 8.5%, and 3.81% respectively for the holdout data.

In addition, we conducted McNemar test to examine whether SVM significantly outperformed the other three models. As a nonparametric test for two related samples, it is particularly useful for before-after measurement of the same subjects (Kim, 2003).

Table 10 shows the results of the McNemar test to statistically compare the classification accuracy for the

Table 8
The classification confusion matrix of CBR

| Actual | Predicted | | | | | |
|---|---|---|---|---|---|---|
| | 1 | 2 | 3 | 4 | 5 | Total |
| 1 | 29 | 12 | 0 | 0 | 0 | 41 |
| 2 | 3 | 85 | 31 | 7 | 0 | 126 |
| 3 | 4 | 34 | 123 | 37 | 0 | 199 |
| 4 | 1 | 10 | 56 | 136 | 8 | 211 |
| 5 | 0 | 0 | 1 | 16 | 10 | 27 |
| 1 | 70.73% | 29.27% | 0.00% | 0.00% | 0.00% | 100 |
| 2 | 2.38% | 67.46% | 24.60% | 5.56% | 0.00% | 100 |
| 3 | 2.02% | 17.17% | 62.12% | 18.69% | 0.00% | 100 |
| 4 | 0.47% | 4.74% | 26.54% | 64.45% | 3.79% | 100 |
| 5 | 0.00% | 0.00% | 3.70% | 59.26% | 37.04% | 100 |

Table 9
The best classification accuracy of SVM, BPN, MDA, and CBR (%)

| | SVM | BPN | MDA | CBR |
|---|---|---|---|---|
| Training set | 77.62 | 62.95 | 58.77 | – |
| Holdout set | 67.22 | 59.93 | 58.72 | 63.41 |

Table 10
McNemar values ($p$-values) for the pairwise comparison of performance

| | BPN | MDA | CBR |
|---|---|---|---|
| SVM | 9.630[a] (0.002)[b,**] | 14.970 (0.000)[**] | 2.766 (0.096)[*] |
| BPN | | 0.156 (0.693) | 1.747 (0.186) |
| MDA | | | 3.439 (0.064)[*] |

[a] Chi-square value.
[b] $p$-value.
[*] $p < 0.1$.
[**] $p < 0.01$.

holdout data among four models. As shown in Table 10, SVM outperforms BPN and MDA at 1% statistical significance level and outperforms CBR at 10% significance level. In addition, Table 10 also shows that the classification accuracy among BPN, MDA, and CBR do not significantly differ each other (CBR outperforms MDA at 10% significance level).

## 5. Conclusions

This study applied SVM to corporate credit rating prediction problems, and showed its attractive prediction power compared to the existing methods. Mapping input vectors into a high-dimensional feature space, SVM transforms complex problems (with complex decision surfaces) into simpler problems that can use linear discriminant functions, and it has been successfully introduced in several financial applications recently. Achieving similar to or better performance than BPN or CBR in practical applications, SVM can conduct classification learning with relatively small amount of data. Also, embodying the structural risk minimization principle (SRM), SVM may prevent the overfitting problem and makes its solution global optimum since the feasible region is convex set.

In particular, this study utilizes a grid-search technique using 5-fold cross-validation in order to choose optimal values of the upper bound $C$ and the kernel parameter $\gamma$ that are most important in SVM model selection. Selecting the optimal parameter values through the grid-search, this study could build a credit rating prediction model with high stability and classification power. To validate the prediction performance of this model, this study statistically compared its prediction accuracy with those of standard three-layer fully connected BPNs, MDA, and CBR respectively. The results of empirical analysis showed that SVM outperformed the other methods. With these results, we can claim that SVM can serve as a promising alternative for the credit rating prediction.

While this study used RBF kernel as a basic kernel function of SVM model, it should be noted that the appropriate kernel function can be problem-specific; hence it remains an interesting topic for further study to derive judicious procedures to select proper kernel functions and the corresponding parameter values according to the types of classification problems.

## References

Altman, E., & Katz, S. (1976). Statistical bond rating classification using financial and accounting data. In M. Schiff & G. Sorter (Eds.), *Topical research in accounting*. Springer.

Ang, J., & Patel, S. K. A. (1975). Bond rating methods: Comparison and validation. *Journal of Finance, 30*(2), 631–640.

Baran, A., Lakonishok, J., & Ofer, A. R. (1980). The value of general price level adjusted data to bond rating. *Journal of Business Finance and Accounting, 7*, 135–149.

Belkaoui, A. (1980). Industrial bond ratings: A new look. *Financial Management, 9*, 44–51.

Bhandari, S. B., Soldofsky, R. M., & Boe, W. J. (1979). Bond quality rating changes for electric utilities: A multivariate analysis. *Financial Management, 8*(1), 74–81.

Butta, P. (1994). Mining for financial knowledge with CBR. *AI Expert, 9*(2), 34–41.

Chang, C.-C., & Lin, C.-J. (2004). LIBSVM: A library for support vector machines. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University. Available from http://www.csie.ntu.edu.tw/~cjlin/papers/libsvm.pdf.

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

Drucker, H., Wu, D., & Vapnik, V. N. (1999). Support vector machines for spam categorization. *IEEE Transactions on Neural Networks, 10*(5), 1048–1054.

Dutta, S., & Shekhar, S. (1996). Bond rating: A non-conservative application of neural networks. In R. R. Trippi & E. Turban (Eds.), *Neural networks in finance and investing*. IRWIN.

Fan, A., & Palaniswami, M. (2000). Selecting bankruptcy predictors using a support vector machine approach. In *Proceedings of the international joint conference on neural networks*.

Gestel, T. V., Suykens, J. A. K., Baestaens, D.-E., Lambrechts, A., Lanckriet, G., Vandaele, B., et al. (2001). Financial time series prediction using least squares support vector machines within the evidence framework. *IEEE Transactions on Neural Networks, 12*(4), 809–821.

Gunn, S. R. (1998). Support vector machines for classification and regression. Technical Report, University of Southampton.

Hair, J. F., Anderson, R. E., Tatham, R. E., & Black, W. C. (1998). *Multivariate data analysis with readings*. Prentice-Hall.

Hearst, M. A., Dumais, S. T., Osman, E., Platt, J., & Scholkopf, B. (1998). Support vector machines. *IEEE Intelligent System, 13*(4), 18–28.

Horrigan, J. L. (1966). The determination of long term credit standing with financial ratios, empirical research in accounting: Selected studies. Supplement to V.4. *Journal of Accounting Research*.

Hsu, C.-W., Chang, C.-C., & Lin, C.-J. (2004). A practical guide to support vector classification. Technical Report, Department of Computer Science and Information Engineering, National Taiwan University. Available from http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf.

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

Kamstra, M., Kennedy, P., & Suan, T.-K. (2001). Combining bond rating forecasts using logit. *The Financial Review, 37*, 75–96.

Kim, J. (1992). A comparative study of rule-based, neural networks, and statistical classification systems for the bond rating problem. Unpublished doctoral dissertation, Richmond, VA: Virginia Common Wealth University.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing, 55*, 307–319.

Kim, K. S. (2005). Predicting bond ratings using publicly available information. *Expert Systems with Applications, 29*, 75–81.

Kim, K. S., & Han, I. (2001). The clustering-indexing method for case-based reasoning using self-organizing maps and learning vector quantization for bond rating cases. *Expert Systems with Applications, 12*, 147–156.

Kwon, Y. S., Han, I., & Lee, C. (1997). Ordinal pairwise partitioning (OPP) approach to neural networks training in bond rating. *International Journal of Intelligent Systems in Accounting, Finance and Management, 6*(1), 23–40.

Maher, J. J., & Sen, T. K. (1997). Predicting bond ratings using neural networks: A comparison with logistic regression. *Intelligent Systems in Accounting, Finance and Management, 6*, 59–72.

Martin, L., Henderson, G., Perry, L., & Cronan, T. (1984). Bond ratings: Predictions using rating agency criteria. Working paper 85-3, Arizona State University, Tempe, AZ.

McAdams, L. (1980). How to anticipate utility bond rating changes. *Journal of Portfolio Management, 7*(1), 56–60.

Moody, J. E., & Utans, J. (1995). Architecture selection strategies for neural networks application to corporate bond rating. In A. Refens (Ed.), *Neural networks in the capital markets*. John Wiley.

Pinches, G. E., & Mingo, K. A. (1973). A multivariate analysis of industrial bond rating. *Journal of Finance, 28*, 1–18.

Pogue, T. F., & Soldofsky, R. M. (1969). What's a bond rating? *Journal of financial and Quantitative Analysis, 4*, 201–228.

Salchenberger, L. M., Cinar, E. M., & Las, N. A. (1992). Neural networks: A new tool for predicting thrift failures. *Decision Sciences, 23*(4), 899–915.

Shaw, M., & Gentry, J. (1990). Inductive learning for risk classification. *IEEE Expert: Intelligent Systems and Their Applications, 5*(1), 47–53.

Shin, K. S., & Han, I. (1999). Case-based reasoning supported by genetic algorithms for corporate bond rating. *Expert Systems with Applications, 16*, 85–95.

Shin, K. S., & Han, I. (2001). A case-based approach using inductive indexing for corporate bond rating. *Decision Support Systems, 32*, 41–52.

Singleton, J. C., & Surkan, A. J. (1995). Bond rating with neural networks. In A. Refens (Ed.), *Neural networks in the capital markets*. John Wiley.

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

Tay, F. E. H., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega, 29*, 309–317.

Vapnik, V. (1998). *Statistical learning theory*. New York: Springer.

Viaene, S., Derrig, R. A., Baesens, B., & Dedene, G. (2002). A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. *Journal of Risk & Insurance, 69*(3), 373–421.