

# An application of support vector machines in bankruptcy prediction model

Kyung-Shik Shin<sup>\*</sup>, Taik Soo Lee<sup>1</sup>, Hyun-jung Kim<sup>2</sup>

*College of Business Administration, Ewha Womans University, 11-1 Daehyun-dong, Seodaemun-gu, Seoul 120-750, South Korea*

## Abstract

This study investigates the efficacy of applying support vector machines (SVM) to bankruptcy prediction problem. Although it is a well-known fact that the back-propagation neural network (BPN) performs well in pattern recognition tasks, the method has some limitations in that it is an art to find an appropriate model structure and optimal solution. Furthermore, loading as many of the training set as possible into the network is needed to search the weights of the network. On the other hand, since SVM captures geometric characteristics of feature space without deriving weights of networks from the training data, it is capable of extracting the optimal solution with the small training set size. In this study, we show that the proposed classifier of SVM approach outperforms BPN to the problem of corporate bankruptcy prediction.

The results demonstrate that the accuracy and generalization performance of SVM is better than that of BPN as the training set size gets smaller. We also examine the effect of the variability in performance with respect to various values of parameters in SVM. In addition, we investigate and summarize the several superior points of the SVM algorithm compared with BPN.

© 2004 Elsevier Ltd. All rights reserved.

*Keywords:* Support vector machines; Bankruptcy prediction

## 1. Introduction

The development of the bankruptcy prediction model has long been regarded as an important and widely studied issue in the academic and business community. The bankruptcy prediction can have significant impact on lending decisions and profitability of financial institutions.

Our research pertains to a bankruptcy prediction model that can provide a basis for credit rating system. Early studies of bankruptcy prediction used statistical techniques such as multiple discriminant analysis (MDA) (Altman, 1968, 1983), logit (Ohlson, 1980) and probit (Zmijewski, 1984). Recently, however, numerous studies have demonstrated that artificial intelligence such as neural networks (NNs) can be an alternative method for classification problems to which traditional statistical method have long been applied (Atiya, 2001; Barniv, Agarwal, & Leach, 1997; Bell, 1997; Boritz & Kennedy, 1995; Charalambous,

Charitous, & Kaourou, 2000; Etheridge & Sriram, 1997; Fletcher & Goss, 1993; Grice & Dugan, 2001; Jo, Han, & Lee, 1997; Lee, Han, & Kwon, 1996; Leshno & Spector, 1996; Odom & Sharda, 1990; Salchenberger, Cinar, & Lash, 1992; Shin & Han, 1998; Tam & Kiang, 1992; Wilson & Sharda, 1994; Zhang, Hu, Patuwo, & Indro, 1999).

Although numerous theoretical and experimental studies reported the usefulness of the back-propagation neural network (BPN) in classification studies, there are several limitations in building the model. First, it is an art to find an appropriate NN model, which can reflect problem characteristics because there are large numbers of controlling parameters and processing elements in the layer. Second, the gradient descent search process to compute the synaptic weights may converge to a local minimum solution that is a good fit for the training examples. Finally, the empirical risk minimization principle that seeks to minimize the training error does not guarantee good generalization performance. To determine the size of the training set is also the main issue to be resolved in the generalization because the sufficiency and efficiency of the training set is one most commonly influenced factor.

This study investigates the effectiveness of support vector machines (SVM) approach in detecting

<sup>\*</sup> Corresponding author. Tel.: +82 2 3277 2799; fax: +82 2 3277 2776.

*E-mail addresses:* ksshin@ewha.ac.kr (K.-S. Shin), leetaik@ewha.ac.kr (T.S. Lee), charitas@empal.com (H.-j. Kim).

<sup>1</sup> Tel.: +82 2 3277 2781; fax: +82 2 3277 2776.

<sup>2</sup> Tel.: +82 2 3277 3767; fax: +82 2 3277 2776.

the underlying data pattern for the corporate failure prediction tasks. SVM classification exercise finds hyperplanes in the possible space for maximizing the distance from the hyperplane to the data points, which is equivalent to solving a quadratic optimization problem. The solution of strictly convex problems for SVM is unique and global. SVM implement the structural risk minimization (SRM) principle that is known to have high generalization performance. As the complexity increases by numbers of support vectors, SVM is constructed through trading off decreasing the number of training errors and increasing the risk of overfitting the data. However, a data-dependent SRM for SVM does not rigorously support the argument that good generalization performance of SVM is attributable to SRM (Borges, 1998). Since SVM captures geometric characteristics of feature space without deriving weights of networks from the training data, it is capable of extracting the optimal solution with the small training set size.

While there are several arguments that support the observed high accuracy of SVM, as the training set size is getting smaller, the preliminary results show that the accuracy and generalization performance of SVM is better than that of the standard BPN. In addition, since choosing an appropriate value for parameters of SVM plays an important role on the performance of SVM, we also investigate the effect of the variability in prediction and generalization performance of SVM with respect to various values of parameters in SVM such as the upper bound  $C$  and the bandwidth of the kernel function according to the size of the training set.

The remainder of this paper is organized as follows. Section 2 provides a description of artificial intelligence applications for the bankruptcy prediction modeling, including a review of prior studies relevant to the research topic of this paper. Section 3 provides a brief description of SVM and demonstrates the several superior points of the SVM algorithm compared with BPN. Section 4 describes the research data and experiments. Section 5 summarizes and analyzes empirical results. Section 6 discusses the conclusions and future research issues.

## 2. A review of bankruptcy prediction studies

Prediction of corporate failure using past financial data is a well-documented topic. Beaver (1966), one of the first researchers to study bankruptcy prediction, investigated the predictability of the 14 financial ratios using 158 samples consisted of failed and non-failed firms. Beaver's study was followed by Altman's model (1968, 1983) based on the MDA to identify the companies into known categories. According to Altman, bankruptcy could be explained quite completely by using a combination of five (selected from an original list of 22) financial ratios. Altman utilized a paired sample design, which incorporated 33 pairs of manufacturing companies. The pairing criteria were predicated upon size and industrial classification. The classification of

Altman's model based on the value obtained for the Z score has a predictive power of 96% for prediction 1 year prior to bankruptcy.

These conventional statistical methods, however, have some restrictive assumptions such as the linearity, normality and independence among predictor or input variables. Considering that the violation of these assumptions for independent variables frequently occurs with financial data (Deakin, 1976), the methods can have limitations to obtain the effectiveness and validity.

Recently, a number of studies have demonstrated that artificial intelligence approaches that are less vulnerable to these assumptions, such as inductive learning, NNs can be alternative methodologies for classification problems to which traditional statistical methods have long been applied. While traditional statistical methods assume certain data distributions and focus on optimizing the likelihood of correct classification (Liang, Chandler, & Han, 1990), inductive learning is a technology that automatically extracts knowledge from training samples, in which induction algorithms such as ID3 (Quinlan, 1986) and CART (Classification and Regression Trees) generate a tree type structure to organize cases in memory. Thus, the difference between a statistical approach and an inductive learning approach is that different assumptions and algorithms are used to generate knowledge structures.

Messier and Hansen (1998) extracted bankruptcy rules using rule induction algorithm that classifies objects into specific groups based on observed characteristics ratios. They drew their data from two prior studies and began with 18 ratios. Their algorithm developed a bankruptcy prediction rule that employed five of these ratios. This method was able to correctly classify 87.5% of the holdout data set.

Shaw and Gentry (1990) applied inductive learning methods to risk classification applications and found that inductive learning's classification performance was better than probit or logit analysis. They have concluded that this result can be attributed to the fact that inductive learning is free from parametric and structural assumptions that underlie statistical methods.

Chung and Tam (1992) compared the performance of two inductive learning algorithms (ID3 and AQ) and NNs using two measures; the predictive accuracy and the representation capability. Results generated by the ID3 and AQ are more explainable yet they have less predictive accuracy than NNs. The predictive accuracy of ID3 and AQ is 79.5% while that of NN is 85.3%.

Because NNs are capable of identifying and representing non-linear relationships in the data set, they have been studied extensively in the fields of financial problems including bankruptcy prediction (Atiya, 2001; Barniv et al., 1997; Bell, 1997; Boritz & Kennedy, 1995; Charalambous et al., 2000; Etheridge & Sriram, 1997; Fletcher & Goss, 1993; Grice & Dugan, 2001; Lee

et al., 1996; Leshno & Spector, 1996; Odom & Sharda, 1990; Salchenberger et al., 1992; Shin & Han, 1998; Tam & Kiang, 1992; Wilson & Sharda, 1994; Zhang et al., 1999). NNs fundamentally differ from parametric statistical models. Parametric statistical models require the developer to specify the nature of the functional relationship such as linear or logistic between the dependent and independent variables. Once an assumption is made about the functional form, optimization techniques are used to determine a set of parameters that minimizes the measure of error. In contrast, NNs with at least one hidden layer use data to develop an internal representation of the relationship between variables so that a priori assumptions about underlying parameter distributions are not required. As a consequence, better results might be expected with NNs when the relationship between the variables does not fit the assumed model (Salchenberger et al., 1992).

The first attempt to use NNs for bankruptcy prediction is done by Odom and Sharda (1990). The model had five input variables the same as the five financial ratios used in Altman's (1968) study, and one hidden layer with five nodes and one node for the output layer. They took a research sample of 65 bankrupt firms between 1975 and 1982, and 64 non-bankrupt firms, overall 129 firms. Among these, 74 firms (38 bankrupt and 36 non-bankrupt firms) were used to form the training set, while the remaining 55 firms (27 bankrupt and 28 non-bankrupt firms) were used to make holdout sample. MDA was conducted on the same training set as a benchmark. As a result, NNs correctly classified 81.81% of the hold out sample while MDA only achieved 74.28%.

Tam and Kiang (1992) compared a NN models' performance with a linear discriminant model, a logit model, the ID3 algorithm, and the k-Nearest Neighbor approach using the commercial bank failure data. The bank data were collected for the period between 1985 and 1987 and consisted of 59 failed and 59 non-failed. Among the evaluated models, NNs showed more accurate and robust results.

Fletcher and Goss (1993) compared a NNs performance with a logit regression model. Their data were drawn from an earlier study and were limited to 36 bankrupt and non-bankrupt firms. Their model used three financial variables, and because of the very small sample size, they used a variation of the 18-fold cross-validation analysis. Although the NN models had higher prediction rates than the logit regression model for almost all risk index cutoff values, due to a very small sample size, the training effort for building NNs was much higher.

Leshno and Spector (1996) used various NN models with novel architecture containing cross terms and cosine terms, different data span and the number of iterations. And they achieved prediction accuracy for the 2-years-ahead case in the range of 74.2–76.4% (depending on the order of the network), compared with 72% for the linear perceptron network. One of their main conclusions was that

the prediction capability of NN models depended on the availability of a large data size used for training. NN models benefited from being exposed to a larger number of examples even though the samples were drawn from distinctive years before the event.

Zhang et al. (1999) also compared a NN models' performance with a logit model, and employed a five-fold cross-validation procedure, on a sample of manufacturing firms. The NNs significantly outperformed the logit regression model with accuracy of 80.46 versus 78.18% for small test set, and with accuracy of 86.64 versus 78.65% for large test set. Since the robustness and performance of the NN model improved significantly from small sets to large sets, user of NN would be well advised to use a large number of sets.

The SVMs is also applied for bankruptcy prediction (Häardle, Moro, & Schäafer, 2003) and compared with NN, MDA and learning vector quantization (LVQ) (Fan & Palaniswami, 2000). SVM obtained the best results (70.35–70.90% accuracy depending on the number of inputs used), followed by NN (66.11–68.33%), followed by LVQ (62.50–63.33%), followed by MDA (multivariate discriminant analysis) (59.79–63.68%). Van Gestel et al. (2003) also reported on the experiment with least squares SVMs, a modified version of SVMs, and showed significantly better results in bankruptcy prediction when contrasted with the classical techniques.

### 3. Support vector machines

Since SVM was introduced from statistical learning theory by Vapnik, a number of studies have been announced concerning its theory and applications. Compared with most other learning techniques, SVM lead to increase performance in pattern recognition, regression estimation, and so on; financial time-series forecasting (Kim, 2003; Mukherjee, Osuna, & Girosi, 1997; Tay & Cao, 2001), marketing (Ben-David & Lindenbaum, 1997), estimating manufacturing yields (Stoneking, 1999), text categorization (Joachims, 2002), face detection using image (Osuna, Freund, & Girosi, 1997), hand written digit recognition (Burges & Scholkopf, 1997; Cortes & Vapnik, 1995), medical diagnosis (Tarassenko, Hayton, Cerneaz, & Brady, 1995). The following brief description of SVM dwells entirely on the pattern recognition problem in classification field. The detailed explanation and proofs of SVM may be contained in the books (Vapnik, 1995, 1998).

SVM produces a binary classifier, the so-called optimal separating hyperplanes, through extremely non-linear mapping the input vectors into the high-dimensional feature space. SVM constructs linear model to estimate the decision function using non-linear class boundaries based on support vectors. If the data is linearly separated, SVM trains linear machines for an optimal hyperplane

that separates the data without error and into the maximum distance between the hyperplane and the closest training points. The training points that are closest to the optimal separating hyperplane are called support vectors. All other training examples are irrelevant for determining the binary class boundaries. In general cases where the data is not linearly separated, SVM uses non-linear machines to find a hyperplane that minimize the number of errors for the training set.

Let us define labeled training examples  $[\mathbf{x}_i, y_i]$ , an input vector  $\mathbf{x}_i \in R^n$ , a class value  $y_i \in \{-1, 1\}$ ,  $i = 1, \dots, l$ .

For the linearly separable case, the decision rules defined by an optimal hyperplane separating the binary decision classes is given as the following equation in terms of the support vectors

$$Y = \text{sign} \left( \sum_{i=1}^N y_i \alpha_i (\mathbf{x} \cdot \mathbf{x}_i) + b \right) \quad (1)$$

where  $Y$  is the outcome,  $y_i$  is the class value of the training example  $\mathbf{x}_i$ , and  $\cdot$  represents the inner product. The vector  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  corresponds to an input and the vectors  $\mathbf{x}_i$ ,  $i = 1, \dots, N$ , are the support vectors. In Eq. (1),  $b$  and  $\alpha_i$  are parameters that determine the hyperplane.

For the non-linearly separable case, a high-dimensional version of Eq. (1) is given as follows:

$$Y = \text{sign} \left( \sum_{i=1}^N y_i \alpha_i K(\mathbf{x}, \mathbf{x}_i) + b \right) \quad (2)$$

The function  $K(\mathbf{x}, \mathbf{x}_i)$  is defined as the kernel function for generating the inner products to construct machines with different types of non-linear decision surfaces in the input space. For constructing the decision rules, three common types of SVM are given as follows:

- A polynomial machine with kernel function

$$K(\mathbf{x}, \mathbf{x}_i) = (\mathbf{x} \cdot \mathbf{x}_i + 1)^d$$

where  $d$  is the degree of the polynomial kernel

- A radial basis function machine with kernel function

$$K(\mathbf{x}, \mathbf{x}_i) = \exp(-1/\delta^2 (\mathbf{x} - \mathbf{x}_i)^2)$$

where  $\delta^2$  is the bandwidth of the radial basis function kernel

- A two-layer NN machine with kernel function

$$K(\mathbf{x}, \mathbf{x}_i) = S[\nu(\mathbf{x} \cdot \mathbf{x}_i)] = 1/[1 + \exp\{\nu(\mathbf{x} \cdot \mathbf{x}_i) - c\}]$$

where  $\nu$  and  $c$  are parameters of a sigmoid function  $S[\nu(\mathbf{x} \cdot \mathbf{x}_i)]$  satisfying the inequality  $c \geq \nu$ .

The SVM classification exercise is implemented in solving a linearly constrained quadratic programming (QP) for finding the support vectors and determining the parameters  $b$  and  $\alpha_i$ . For the separable case, there is a lower bound 0 on the coefficient  $\alpha_i$  in Eq. (1). For the non-separable

case, SVM can be generalized by placing an upper bound  $C$  on the coefficients  $\alpha_i$  in addition to the lower bound (Witten & Frank, 2000).

In brief, the learning process to construct decision functions of SVM is completely represented by the structure of two layers, which seems to be similar with BPN. However, learning algorithm is different in that SVM is trained with optimization theory that minimizes misclassification based on statistical learning theory. The first layer selects the basis  $K(\mathbf{x}, \mathbf{x}_i)$ ,  $i = 1, \dots, N$  and the number of support vectors from given set of bases defined by the kernel. The second layer constructs the optimal hyperplane in the corresponding feature space (Vapnik, 1998). The scheme of SVM is shown in Fig. 1.

Compared with the limitations of the BPN, the major advantages of SVM are as follows: first, SVM has only two free parameters, namely the upper bound and kernel parameter. On the other hand, because a large number of controlling parameters in BPN such as the number of hidden layers, the number of hidden nodes, the learning rate, the momentum term, epochs, transfer functions and weights initialization methods are selected empirically, it is a difficult task to obtain an optimal combination of parameters that produces the best prediction performance.

Second, SVM guarantees the existence of unique, optimal and global solution since the training of SVM is equivalent to solving a linearly constrained QP. On the other hand, because the gradient descent algorithm optimizes the weights of BPN in a way that the sum of square error is minimized along the steepest slope of the error surface, the result from training may be massively multimodal, leading to non-unique solutions, and be in the danger of getting stuck in a local minima.

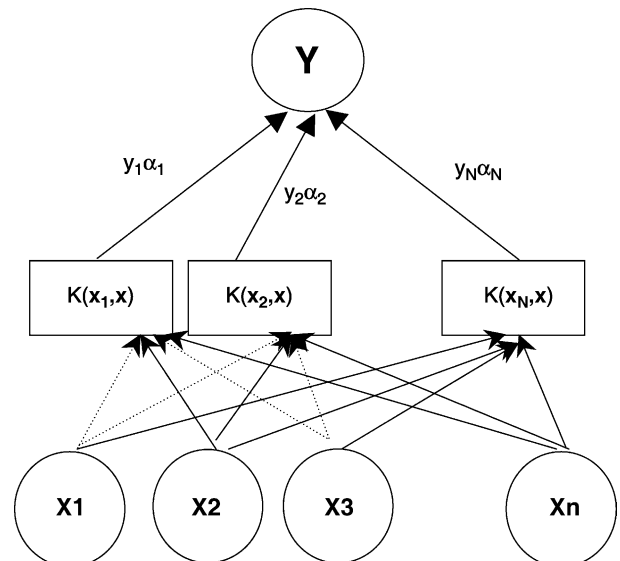


Fig. 1. The scheme of SVM (adapted from Vapnik, 1995).



Third, SVM implement the SRM principle that is known to have a good generalization performance. SRM is the approach to trading off empirical error with the capacity of the set called VC dimension, which seeks to minimize an upper bound of the generalization error rather than minimize the training error. In order to apply SRM, the hierarchy of hypothesis spaces must be defined before the data is observed. But in SVM, the data is first used to decide which hierarchy to use and then subsequently to find a best hypothesis from each. Therefore the argument that good generalization performance of SVM is attributable to SRM is flawed, since the result of SVM is obtained from a data-dependent SRM (Borges, 1998).

Although the other reason why SVM has good generalization performance is suggested (Vapnik, 1998), there exists no explicitly established theory that shows good generalization performance is guaranteed for SVM. However, it seems plausible that performance of SVM is more general than that of BPN by reason that the two measures in terms of the margin and number of support vectors give information about the relation between the input and target function according to different criteria, either of which is sufficient to indicate good generalization. On the other hand, BPN trained based on minimizing a squared error criterion at the network output tends to produce a classifier with the only large margin measure. In addition, flexibility caused by choosing training data is likely to occur with weights of BPN model, but the maximum hyperplane of SVM is relatively stable and gives little flexibility in the decision boundary (Witten & Frank, 2000).

Finally, SVM is constructed with the small training data set size, since it learns by capturing geometric picture corresponding to the kernel function. Moreover, no matter how large the training set size is, SVM has infinite VC dimension. That is why SVM is capable of extracting the optimal solution with the small training set size. On the other hand, for the case of BPN containing a single hidden layer and used as a binary classifier, it is provided that the number of training examples, with an error of 10%, should be approximately 10 times the number of weights in the network. With 10 input and hidden nodes, the learning algorithm will need more than 1000 training set size that is sufficient for a good generalization (Haykin, 1994). However, in most practical applications, there can be a huge numerical gap between the actual size of the training set needed and that is available.

Due to utilizing the feature space images by the kernel function, SVM is applicable in such circumstances that have proved difficult or impossible for BPN where data in the plane is randomly scattered and where the density of the data's distribution is not even well defined (Friedman, 2002). BPN tends to recognize patterns where the training data is dense, at the expense of sparse areas.

#### 4. Research data and experiments

The research data we employ is provided by Korea Credit Guarantee Fund in Korea, and consists of externally non-audited 2320 medium-size manufacturing firms, which filed for bankruptcy (1160 cases) and non-bankruptcy (1160 cases) from 1996 to 1999. We select 1160 non-bankrupt firms randomly from among all solvent firms, so the choice covers the whole spectrum from healthy to borderline firms in order to avoid any selection bias. The data set is arbitrarily split into two subsets; about 80% of the data is used for a training set and 20% for a validation set. The training data for SVM is totally used to construct the model and for BPN is divided into 60% training set and 20% test set. The validation data is used to test the results with the data that is not utilized to develop the model.

Using this first data set, seven different datasets are constructed that differs in the number of cases included in the training and test subsamples reduced to scale. The validation set of all datasets consisting of 464 cases is identical, which ensures that the obtained results from validation data are not influenced by the fluctuation of data arrangement.

We apply two stages of the input variable selection process. At the first stage, we select 52 variables among more than 250 financial ratios by independent-samples *t*-test between each financial ratio as an input variable and bankrupt or non-bankrupt as an output variable. In the second stage, we select 10 variables using a MDA stepwise method to reduce dimensionality. We select input variables satisfying the univariate test first, and then select significant variables by the stepwise method for refinement. The selected variables for this research are shown in Table 1.

In this study, the radial basis function is used as the kernel function of SVM. Since SVM does not have a general guidance for determining the upper bound  $C$  and the kernel parameter  $\delta^2$ , this study varies the parameters to select optimal values for the best prediction performance. The MATLAB SVM toolbox version 0.55 beta executes these processes (Cawley, 2000).

For verifying the applicability of SVM, we also design BPN as the benchmark with the following controlling

Table 1  
Definition of variables

Variable	Definition
$x_1$	Total asset growth
$x_2$	Contribution margin
$x_3$	Operating income to total asset
$x_4$	Fixed asset to sales
$x_5$	Owner's equity to total asset
$x_6$	Net asset to total asset
$x_7$	Net loan dependence rate
$x_8$	Operating asset constitute ratio
$x_9$	Working capital turnover period
$x_{10}$	Net operating asset turnover period

parameters. The structure of BPN is standard three-layer with the same number of input nodes in the hidden layer and the hidden and output nodes use the sigmoid transfer function. For stopping the training of BPN, test set that is not a subset of the training set is used, but the optimum network for the data in the test set is still difficult to guarantee generalization performance. The NN algorithms software NeuroShell 2 version 4.0 executes these processes.

## 5. Results and analysis

To investigate the effectiveness of the SVM approach trained by small data set size in the context of the corporate

bankruptcy classification problem, we conduct the experiment with respect to various kernel parameters and the upper bound  $C$ , and compare the prediction performance of SVM with various parameters as the training set size gets smaller. Based on the results proposed by Tay and Cao (2001), we set an appropriate range of parameters as follows: a range for kernel parameter is between 1 and 100 and a range for  $C$  is between 1 and 100. Test results for this study are summarized in Table 2. Each cell of Table 2 contains the accuracy of the classification techniques.

The results in Table 2 show that the overall prediction performance of SVM on the validation set is consistently good as the number of training set size decreases. Moreover, the accuracy and the generalization using a small size of

Table 2  
Classification accuracies (%) of various parameters in SVM on various data set sizes

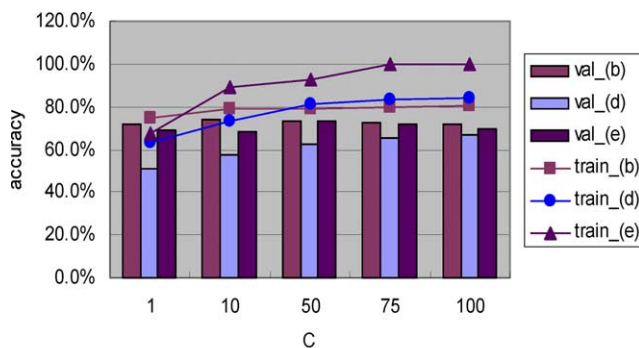
C	$\delta^2=1$		$\delta^2=25$		$\delta^2=50$		$\delta^2=75$		$\delta^2=100$	
	Training	Val	Training	Val	Training	Val	Training	Val	Training	Val
<i>(a) 1st set (1856)</i>										
1	81.0	69.8	72.4	75.0	71.2	73.3	70.4	72.0	70.2	72.0
10	88.1	67.0	74.6	76.7	74.3	76.1	73.5	75.9	72.8	75.0
50	93.6	62.7	75.5	75.2	74.5	76.1	74.3	76.3	74.6	75.6
75	95.2	61.0	75.5	73.7	75.0	75.9	74.6	76.3	74.3	76.3
100	95.7	60.3	75.4	73.9	75.2	76.5	74.5	76.3	74.2	76.3
<i>(b) 2nd set (SVM training set of (a)*25: 464)</i>										
1	87.3	65.9	74.4	71.6	72.8	68.8	71.3	67.9	70.3	66.2
10	93.8	64.4	78.9	73.9	77.4	75.0	75.2	74.6	74.6	74.4
50	97.8	61.0	78.7	73.3	79.5	73.7	78.2	74.8	77.8	75.0
75	98.9	60.1	79.7	72.6	79.7	73.1	78.9	73.5	77.6	75.6
100	99.1	59.9	80.6	71.8	79.7	72.6	79.3	73.7	78.7	73.7
<i>(c) 3rd set (SVM training set of (b)*50: 232)</i>										
1	81.2	61.2	61.0	59.7	62.0	55.4	57.3	50.9	57.3	52.4
10	92.0	56.5	69.5	65.7	64.8	62.7	62.4	60.6	61.5	58.8
50	98.1	56.5	72.3	65.7	70.9	66.8	70.4	66.2	70.0	63.8
75	99.5	57.1	73.7	65.5	70.9	65.9	71.4	66.8	71.4	65.5
100	100.0	57.1	73.2	65.5	71.8	66.2	70.9	67.5	71.8	67.0
<i>(d) 4th set (SVM training set of (c)*50: 116)</i>										
1	85.8	55.0	63.2	51.1	57.5	51.9	56.6	51.3	54.7	50.6
10	97.2	57.8	73.6	57.8	67.9	55.6	64.2	54.1	63.2	53.2
50	99.1	56.3	81.1	62.7	74.5	58.6	69.8	57.5	70.8	57.5
75	100.0	55.6	83.0	65.7	78.3	60.3	73.6	57.8	69.8	57.5
100	100.0	54.5	84.0	67.0	79.2	60.8	75.7	59.5	70.8	57.5
<i>(e) 5th set (SVM training set of (d)*50: 58)</i>										
1	93.1	61.6	72.4	66.2	70.7	68.3	67.2	67.0	67.2	66.8
10	100.0	60.6	82.9	72.0	75.9	70.5	70.7	69.2	70.7	67.7
50	100.0	60.1	87.9	75.6	84.5	74.4	79.3	72.6	74.1	70.5
75	100.0	59.7	93.1	73.3	84.5	73.9	84.5	73.5	77.6	70.9
100	100.0	59.9	91.4	72.6	87.9	75.2	84.5	73.3	81.0	73.5
<i>(f) 6th set (SVM training set of (e)*50: 28)</i>										
1	100.0	60.6	67.9	69.2	60.7	57.5	60.7	57.5	64.3	64.9
10	100.0	58.8	89.3	68.3	71.4	64.9	71.4	64.2	71.4	64.9
50	100.0	58.8	92.9	73.3	89.3	70.0	89.3	69.8	89.3	69.4
75	100.0	58.8	100.0	71.6	89.3	71.1	89.3	71.1	89.3	70.0
100	100.0	58.8	100.0	70.0	92.9	73.3	89.3	70.9	89.3	70.3
<i>(g) 7th set (SVM training set of (f)*50: 14)</i>										
1	100.0	54.3	78.6	66.4	71.4	62.1	71.4	61.2	78.6	64.2
10	100.0	54.3	85.7	61.9	85.7	61.0	85.7	62.3	85.7	62.9
50	100.0	54.3	100.0	61.6	92.9	62.9	85.7	62.9	85.7	62.7
75	100.0	54.3	100.0	62.1	92.9	62.9	92.9	62.1	85.7	63.1
100	100.0	54.3	100.0	62.1	92.9	64.0	92.9	63.8	92.9	62.9

data set (5th, 6th, and 7th set) are even better than those using a large size of data set (3rd and 4th set). Especially the performance of the 5th and 6th set is similarly excellent compared to that of the 1st set.

The experimental result also shows that the prediction performance of SVM is sensitive to the various kernel parameter  $\delta^2$  and the upper bound  $C$ . In Table 2, the results of SVM show the best prediction performances when  $\delta^2$  is 25 and  $C$  is 75 on the most data set of various data set sizes. Fig. 2 gives the results of SVM on the 2nd, 4th, 6th data set with various  $C$  where  $\delta^2$  is fixed at 25.

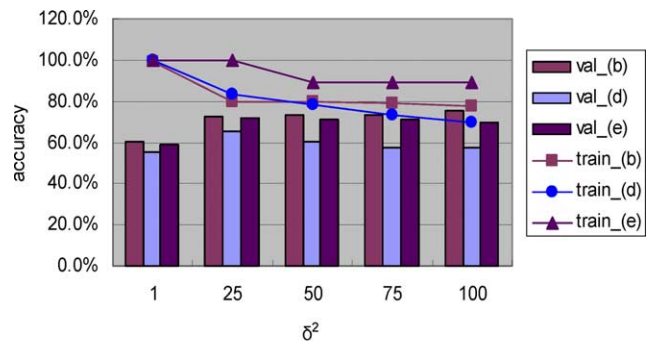
The accuracy on the training set increases monotonically as  $C$  increases; on the contrary, the accuracy on the validation set shows a tendency to increase slightly. This indicates that a large value for  $C$  has an inclination to over-fit the training data and an appropriate value for  $C$  plays a leading role on preventing SVM from deterioration in the generalization performance. According to Tay and Cao (2001), a small value for  $C$  would under-fit the training data because the weight placed on the training data is too small thus resulting in small values of prediction accuracy on both the training and validation sets while a large value for  $C$  would over-fit the training data. In this study, the prediction performance on the training set increases as  $C$  increases while the prediction performance on the validation set maintains an almost constant value as  $C$  increases. These results partly support the conclusions of Tay and Cao (2001).

Fig. 3 gives the results of SVM on the 2nd, 4th, 6th data set with various  $\delta^2$  where  $C$  is fixed at 75. The accuracy on the training set of the most data set decreases as  $\delta^2$  increases; on the other hand, the accuracy on the validation set shows a tendency to increase with increasing  $\delta^2$ . In addition, however, the prediction performance of the validation set is more stable and insensitive than that of training set. This indicates that a small value for  $\delta^2$  has an inclination to over-



val\_(b): the validation set of the 2<sup>nd</sup> data set (b)  
 val\_(d): the validation set of the 4<sup>th</sup> data set (d)  
 val\_(e): the validation set of the 6<sup>th</sup> data set (e)  
 train\_(b): the train set of the 2<sup>nd</sup> data set (b)  
 train\_(d): the train set of the 4<sup>th</sup> data set (d)  
 train\_(e): the train set of the 6<sup>th</sup> data set (e)

Fig. 2. Results of SVM with various  $C$  where  $\delta^2$  is fixed at 25.



val\_(b): the validation set of the 2<sup>nd</sup> data set (b)  
 val\_(d): the validation set of the 4<sup>th</sup> data set (d)  
 val\_(e): the validation set of the 6<sup>th</sup> data set (e)  
 train\_(b): the train set of the 2<sup>nd</sup> data set (b)  
 train\_(d): the train set of the 4<sup>th</sup> data set (d)  
 train\_(e): the train set of the 6<sup>th</sup> data set (e)

Fig. 3. Results of SVM with various  $\delta^2$  where  $C$  is fixed at 75.

fit the training data and an appropriate value for  $\delta^2$  also plays an important role on the generalization performance of SVM. According to Tay and Cao (2001), a small value of  $\delta^2$  would over-fit the training data while a large value of  $\delta^2$

Table 3  
 Comparison of classification accuracies (%) between the best SVM and BPN on various data set sizes

Data set	SVM		BPN	
	Number of set	Accuracy	Number of set	Accuracy
<i>(a) 1st set</i>				
	Train (1856)	74.6	Train (1392)	71.7
	Val (464)	76.7	Val (464)	72.2
	Total (2320)		Test (464)	74.1
<i>(b) 2nd set (SVM training set of (a)*25%)</i>				
	Train (464)	77.6	Train (370)	75.7
	Val (464)	75.6	Val (464)	71.6
	Total (928)		Test (94)	72.3
<i>(c) 3rd set (SVM training set of (b)*50%)</i>				
	Train (232)	70.9	Train (184)	57.1
	Val (464)	67.5	Val (464)	56.0
	Total (696)		Test (48)	56.3
<i>(d) 4th set (SVM training set of (c)*50%)</i>				
	Train (116)	84.0	Train (92)	55.4
	Val (464)	67.0	Val (464)	54.5
	Total (580)		Test (24)	54.2
<i>(e) 5th set (SVM training set of (d)*50%)</i>				
	Train (58)	87.9	Train (46)	47.8
	Val (464)	75.6	Val (464)	50.9
	Total (522)		Test (12)	50.0
<i>(f) 6th set (SVM training set of (e)*50%)</i>				
	Train (28)	92.9	N/A	
	Val (464)	73.3		
	Total (492)			
<i>(g) 7th set (SVM training set of (f)*50%)</i>				
	Train (14)	78.6	N/A	
	Val (464)	66.4		
	Total (478)			

N/A, not applicable.

would under-fit the training data. These results also support the conclusions of Tay and Cao (2001).

In addition, the results of the best SVM model that present the best prediction performance for the validation set obtained from various data set sizes are compared with those of BPN and are summarized in Table 3. Each cell of Table 3 contains the accuracy of the classification techniques.

In Table 3, SVM has higher prediction accuracy than BPN as the training set size gets smaller. While the results of BPN are comparable with SVM in large size of data set (1st and 2nd set), in case that a training set size is less than 200, it is hard (3rd, 4th, and 5th set) or even impossible (6th and 7th set) to learn the model. Since BPN is not a well-controlled learning machine, training of BPN with a small size of data set requires more experience and care than that with a large size of data set for a good performance.

From the results of experiments, we can conclude that SVM is the better approach to learn a small size of data patterns as opposed to ordinary BPN. It can be said that SVM is a promising classifier to obtain a superior prediction performance from small data sets for the given task.

## 6. Conclusions

In this study, we show that the proposed classifier of SVM approach outperforms BPN to the problem of corporate bankruptcy prediction. Our experimentation results demonstrate that SVM has the highest level of accuracies and better generalization performance than BPN as the training set size is getting smaller sets. We also examine the effect of various values of parameters in SVM such as the upper bound  $C$  and the bandwidth of the kernel function. In addition, we investigate and summarize the several superior points of the SVM algorithm compared with BPN.

Our study has the following limitations that need further research. First, in SVM, the choice of the kernel function and the determination of optimal values of the parameters have a critical impact on the performance of the resulting system. We also need to investigate to develop a structured method of selecting an optimal value of parameters in SVM for the best prediction performance as well as the effect of other factor that is fixed in the above experiment such as the kernel function. The second issue for future research relates to the generalization of SVM on the basis of the appropriate level of the training set size and gives a guideline to measure the generalization performance.

## References

Altman, E. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. *The Journal of Finance*, 23, 589–609.

Altman, E. (1983). *Corporate financial distress—a complete guide to predicting, avoiding and dealing with bankruptcy*. New York: Wiley.

Atiya, A. (2001). Bankruptcy prediction for credit risk using neural networks: a survey and new results. *IEEE Transactions on Neural Networks*, 12(4).

Barniv, R., Agarwal, A., & Leach, R. (1997). Predicting the outcome following bankruptcy filing: a three-state classification using neural networks. *Intelligent Systems in Accounting, Finance and Management*, 6, 177–194.

Beaver, W. (1966). Financial ratios as prediction of failure. Empirical research in accounting: selected studies. *Journal of Accounting Research*, 4, 71–111.

Bell, T. (1997). Neural nets or the logit model? A comparison of each model's ability to predict commercial bank failures. *Intelligent Systems in Accounting, Finance and Management*, 6, 249–264.

Ben-David, S., & Lindenbaum, M. (1997). Learning distributions by their density levels: a paradigm for learning without a teacher. *Journal of Computer and System Sciences*, 55, 171–182.

Boritz, J., & Kennedy, D. (1995). Effectiveness of neural networks types for prediction of business failure. *Expert Systems with Applications*, 9, 503–512.

Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2), 955–974.

Burges, C. J. C., & Scholkopf, B. (1997). Improving the accuracy and speed of support vector machines. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in Neural Information Processing Systems* (pp. 475–481). Cambridge, MA: MIT Press.

Cawley, G. C. (2000). *Support vector machine toolbox* University of East Anglia, School of Information Systems available on.

Charalambous, C., Charitous, A., & Kaourou, F. (2000). Comparative analysis of artificial neural network models: application in bankruptcy prediction. *Annals of Operations Research*, 99, 403–425.

Chung, H., & Tam, K. (1992). A comparative analysis of inductive learning algorithm. *Intelligent Systems in Accounting, Finance and Management*, 2, 3–18.

Cortes, C., & Vapnik, V. N. (1995). Support vector networks. *Machine Learning*, 20, 273–297.

Deakin, B. E. (1976). A discriminant analysis of predictors of business failure. *Journal of Accounting Research*, 167–179.

Etheridge, H., & Sriram, R. (1997). A comparison of the relative costs of financial distress models: artificial neural networks, logit and multivariate discriminant analysis. *Intelligent Systems in Accounting, Finance and Management*, 6, 235–248.

Fan, A., Palaniswami, M. (2000). A new approach to corporate loan default prediction from financial statements. *Proceedings of the computational finance/forecasting financial markets conference, London (CD), UK*.

Fletcher, D., & Goss, E. (1993). Forecasting with neural networks: an application using bankruptcy data. *Information and Management*, 24(3), 159–167.

Friedman, C. (2002). *Credit model technical white paper. Standard and Poor's*. New York: McGraw-Hill.

Grice, S. J., & Dugan, T. M. (2001). The limitations of bankruptcy prediction models: some cautions for the researcher. *Review of Quantitative Finance and Accounting*, 17, 151–166.

Häardle, W., Moro, R., & Schäfer, D. (2003). *Predicting corporate bankruptcy with support vector machines* Working Slide, Humboldt University and the German Institute for Economic Research available on.

Haykin, S. (1994). *Neural networks: A comprehensive foundation*. New York: Macmillan.

Jo, H., Han, I., & Lee, H. (1997). Bankruptcy prediction using case-based reasoning, neural networks, and discriminant analysis. *Expert Systems With Applications*, 13(2), 97–108.

Joachims, T. (2002). *Learning to classify text using support vector machines*. London: Kluwer Academic Publishers.

Kim, K. J. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, 55(1/2), 307–319.



- Lee, K. C., Han, I. G., & Kwon, Y. (1996). Hybrid neural network models for bankruptcy predictions. *Decision Support Systems*, 18, 63–72.
- Leshno, M., & Spector, Y. (1996). Neural network prediction analysis: the bankruptcy case. *Neurocomputing*, 10, 125–247.
- Liang, T. P., Chandler, J. S., & Han, I. (1990). Integrating statistical and inductive learning methods for knowledge acquisition. *Expert Systems with Applications*, 1, 391–401.
- Messier, W., & Hansen, J. (1998). Inducing rules for expert system development: an example using default and bankruptcy data. *Management Science*, 34(12), 1403–1415.
- Mukherjee, S., Osuna, E., Girosi, F. (1997). Nonlinear prediction of chaotic time series using support vector. *Proceedings of the IEEE workshop on neural networks for signal processing, Amelia Island, FL* (pp. 511–520).
- Odom, M., & Sharda, R. (1990). A neural networks model for bankruptcy prediction. *Proceedings of the IEEE International Conference on Neural Network*, 2, 163–168.
- Ohlson, J. (1980). Financial ratios and the probabilistic prediction of bankruptcy. *Journal of Accounting Research*, 18(1), 109–131.
- Osuna, E., Freund, R., & Girosi, F. (1997). Training support vector machines: an application to face detection. *Proceedings of Computer Vision and Pattern Recognition*, 130–136.
- Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, 1, 81–106.
- Salchenberger, L., Cinar, E., & Lash, N. (1992). Neural networks: a new tool for predicting thrift failures. *Decision Sciences*, 23, 899–916.
- Shaw, M., & Gentry, J. (1990). Inductive learning for risk classification. *IEEE Expert*, 47–53.
- Shin, K. S., Han, I. (1998). Bankruptcy prediction modeling using multiple neural networks models. *Proceedings of Korea management science institute conference*.
- Stoneking, D. (1999). Improving the manufacturability of electronic designs. *IEEE Spectrum*, 36(6), 70–76.
- Tam, K., & Kiang, M. (1992). Managerial applications of neural networks: the case of bank failure predictions. *Management Science*, 38(7), 926–947.
- Tarassenko, L., Hayton, P., Cerneaz, N., Brady, M. (1995). Novelty detection for the identification of masses in mammograms. *Proceedings fourth IEE international conference on artificial neural networks, Cambridge* (pp. 442–447).
- Tay, F. E. H., & Cao, L. (2001). Application of support vector machines in financial time series forecasting. *Omega*, 29, 309–317.
- Van Gestel, T., Baesens, B., Suykens, J., Espinoza, M., Baestaens, D.E., Vanthienen, J., De Moor, B. (2003). Bankruptcy prediction with least squares support vector machine classifiers. *Proceedings of the IEEE international conference on computational intelligence for financial engineering, Hong Kong* (pp. 1–8).
- Vapnik, V. N. (1995). *The nature of statistical learning theory*. New York: Springer.
- Vapnik, V. N. (1998). *Statistical learning theory*. New York: Wiley.
- Wilson, R., & Sharda, R. (1994). Bankruptcy prediction using neural networks. *Decision Support Systems*, 11(5), 545–557.
- Witten, I. H., & Frank, E. (2000). *Data mining: Practical machine learning tools and techniques with Java implementations*. San Francisco, CA: Morgan Kaufmann.
- Zhang, G., Hu, Y. M., Patuwo, E. B., & Indro, C. D. (1999). Artificial neural networks in bankruptcy prediction: general framework and cross-validation analysis. *European Journal of Operational Research*, 116, 16–32.
- Zmijewski, M. E. (1984). Methodological issues related to the estimated of financial distress prediction models. *Journal of Accounting Research*, 22(1), 59–82.