Bankruptcy Prediction Using SVM Models with a New Approach to Combine Features Selection and Parameters Optimization

Ligang Zhou
Faculty of Management and Administration, Macau University of Science and Technology, Taipa, Macau
Email: ligzhou@must.edu.mo

Kin Keung Lai
Department of Management Sciences, City University of Hong Kong, Kowloon, Hong Kong
College of Management, North China Electric Power University, Beijing, China
Email: mskklai@cityu.edu.hk

Jerome Yen
School of Business, Tung Wah College, Kowloon, Hong Kong
Email: risksolution@gmail.com

Abstract: Due to the economic significance of bankruptcy prediction of companies for financial institutions, investors and governments, many quantitative methods have been used to develop effective prediction models. Support vector machines (SVM), a powerful classification method, has been used for this task, however, the performance of SVM is sensitive to model form, parameters setting and features selection. In this study, a new approach based on direct search and features ranking technology is proposed to optimize features selection and parameters setting for 1-norm and least square SVM models for bankruptcy prediction. This approach is also compared to the SVM models with parameters optimization and features selection by the popular Genetic Algorithm (GA) technique. The experimental results on a data set with 2010 instances show that the proposed models are good alternatives for bankruptcy prediction.

Keywords: Bankruptcy prediction; Support vector machines; Direct search; Genetic Algorithm

1. Introduction

The economic significance of corporate bankruptcy prediction for credit granting institutions, investors and government cannot be overstated. Therefore, many studies have been conducted for developing bankruptcy prediction models with objective of

improving prediction accuracy. These studies can be categorized into two broad streams. For large amounts of financial information disclosed by companies and equally a lot of information about external macroeconomic and financial markets, one stream focuses on selection or derivation of information that is effective to predict corporate bankruptcy. Beaver (1966) identified 30 ratios that he considered important factors for forecasting corporate bankruptcy. These ratios were tested by a univariate discriminant analysis model on 79 pairs of bankruptcy/non-bankruptcy firms; the empirical results showed that "working capital funds flow/total assets" and "net income/total assets" are the two most efficient ratios that could correctly classify 90% and 88% of the firms, respectively. Altman (1968) selected 5 ratios, employed a multivariate discriminant analysis model, and tested the model on 33 pairs of bankruptcy/non-bankruptcy firms. The model could correctly identify 90% of the firms one year prior to failure. The 5 selected ratios were: Working capital/total assets, Retained earnings/Total assets, EBIT/Total assets, Market value equity/Book value of total debt, and Sales/Total assets. Nam et al. (2008) presented a duration model with time-varying covariates and a baseline hazard function incorporating macroeconomic dependencies. Their empirical results showed that forecasting performance can be improved by allowing temporal and macroeconomic dependencies, such as changes in interest rate (CIR) and the volatility of foreign exchange rate (VFE). Almost all the above mentioned studies employ accounting ratios. Therefore, these models are called accounting-ratio-based models. Hillegeist et al. (2004), Vassalou and Xing (2004), Agarwal and Taffler (2008) discussed employment of market information for corporate failure prediction, and such models are called market-based models. Their preference for such market-based models is mainly based on the following reasons:

(a) In efficient markets, stock prices reflect all the information in financial statements and also contain information not available in financial statements;
(b) Market variables are real-time and are not likely influenced by accounting policies;
(c) The output of such models is not time or sample dependent.

In summary, the information available for bankruptcy prediction includes accounting ratios, macroeconomic information and market information. Ravi Kumar and Ravi (2007)

reviewed 128 papers and listed more than 500 different variables that have been used for bankruptcy prediction.

Recently, research has more focus on the choice of methodology. Wilson and Sharda (1994) used neural networks for bankruptcy prediction and their study indicates that neural networks perform significantly better than discriminant analysis. Jo and Han (1996) suggested a new structured model based on case-based forecasting, neural networks, and discriminant analysis for bankruptcy prediction. Their experimental results showed that the integration approach produces higher prediction accuracy than individual models. Premachandra et al. (2011) used an additive super-efficiency data envelopment analysis (DEA) model for predicting corporate failure. They found that DEA model is relatively weak in predicting corporate failures compared to healthy firms prediction, but this weakness can be improved by the assessment index. Chen et al. (2011) proposed a model based on an adaptive fuzzy k-nearest neighbor method and used continuous particle swarm optimization (PSO) approach to select the neighborhood size k and fuzzy strength parameter m. McKee and Lensberg (2002) investigated a hybrid approach based on genetic algorithms and rough sets to bankruptcy prediction. Their findings indicate that the hybrid model is efficient and effective to provide both high prediction accuracy and theoretical insights which can reveal relationships between variables. Divsalar et al. (2011) constructed a hybrid model based on genetic programming with orthogonal least squares and simulated annealing algorithms. Verikas et al. (2010) presented a comprehensive review of hybrid and ensemble-based soft computing techniques applied to bankruptcy prediction.

Support vector machine is a powerful classification method that has been employed in many business applications, including bankruptcy prediction. Shin et al. (2005) investigated the effectiveness of SVM approach in detecting the underlying pattern for corporate failure prediction and made a comparison with Back-propagation neural network (BPN). They found that SVM shows the highest level of accuracy and better generation performance than BPN; especially, the training set size is small. Härdle et al. (2009) explored the suitability of smooth support vector machines (SSVM) in predicting default risk of companies and investigate how important factors such as selection of appropriate accounting ratios, length of training period and structure of the training

sample influence the precision of prediction. Although SVM has god performance on classification accuracy, one main disadvantage of SVM method is the difficulty in interpreting the results. If the decision makers pay more attention to the knowledge behind the models, the regression models, decision tee and some rule-based method (Wang and Dillon 2006) will be the more proper choice.

The performance of SVM is not only dependent on the input predictors, but also on selection of parameters in the models. Most of previous studies focus on employment of genetic algorithms (GA) for parameters selection. Min et al. (2006), Ahn et al. (2006), Wu et al. (2007) introduce GA to perform features selection and parameters optimization for SVM model for bankruptcy prediction. Shie et al. (2011) employed partial swam optimization (PSO) to select features and optimize parameters for SVM models. Gomes et al. (2012) proposed the combination of meta-learning and search algorithms (PSO and Tabu search) to deal with the problem of SVM parameters selection. However, the number of parameters in SVM models is always one or two, and for this kind of optimization problem, with fewer variables, direct search is also a good alternative (Zhou et al. 2008) . In addition, GA is a stochastic method, which cannot get the same model from the same training samples set when running twice. It is difficult for the decision makers to accept it. In this study, a deterministic approach based on direct search and features ranking technology is proposed to combine features selection and parameters optimization for SVM models for bankruptcy prediction and is compared to the SVM models whose parameters and feature selection is conducted by GA.

The rest of this paper is organized as follows. Section 2 briefly introduces SVM models, including 1-norm SVM and least squares SVM (LSSVM). Direct search and genetic algorithm based features selection and parameters optimization methods are described in Section 3. Section 4 reports results of the empirical study and a short discussion, and conclusion is presented in Section 5.

2. Support Vector Machines Models

2.1 1-norm SVM

Given a training set $\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots (\boldsymbol{x}_k, y_k), \cdots (\boldsymbol{x}_N, y_N)\}$, where $\boldsymbol{x}_k \in R^m$ and $y_k \in \{-1, +1\}$, and each element $(\boldsymbol{x}_k, y_k)$ is corresponding to a point in a high-dimensional space. Suppose all the points can be separated by a hyperplane $\boldsymbol{w}^T \boldsymbol{x} + b = 0$; then it is natural to construct a linear classifier as follows (Vapnik 1998).

$$y(\mathbf{x}) = \mathrm{s}ign(\boldsymbol{w}^T \boldsymbol{x} + b) \tag{1}$$

If all the data of the two classes are separable, one can say that

$$\begin{cases} \boldsymbol{w}^T \boldsymbol{x}_k + b \geq +1, & if\ y_k = +1 \\ \boldsymbol{w}^T \boldsymbol{x}_k + b \leq -1, & if\ y_k = -1 \end{cases} \tag{2}$$

The above set of inequalities can be transformed into the following compact form:

$$y_k(\boldsymbol{w}^T \boldsymbol{x}_k + b) \geq 1,\ k = 1, \ldots, N. \tag{3}$$

The traditional SVM finds an optimal separating hyperplane to maximize the margin, subject to the condition that all training data points need to be correctly classified, and it gives the following optimization problem:

$$\begin{aligned} &\underset{\boldsymbol{w},b}{Min}\ Z = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} \\ &subject\ to: y_k(\boldsymbol{w}^T \boldsymbol{x}_k + b) \geq 1,\ k = 1, \ldots, N. \end{aligned} \tag{4}$$

Usually, for most real-life problems, it is difficult to find a hyperplane that can separate all the data points correctly. Therefore, an extension of linear SVM to a non-separable case is made, to introduce an additional slack variable $\xi_k \geq 0$, which indicates the misclassification errors. The problem of finding an optimal separating hyperplane combining two objectives: maximizing the margin and minimizing the classification errors is formulated as follows.

$$\begin{aligned} &\underset{\boldsymbol{w},b,\xi}{Min}\ Z = \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} + C\sum_{k=1}^{N} \xi_k \\ &subject\ to: y_k(\boldsymbol{w}^T \boldsymbol{x}_k + b) \geq 1 - \xi_k,\ k = 1, \ldots, N \\ &\qquad\qquad \xi_k \geq 0,\ k = 1, \ldots, N \end{aligned} \tag{5}$$

where C is a penalty parameter on the training error, which is a positive real constant.

The above linear SVM model has been extended to a nonlinear model, which was viewed as an important progress in SVM theory, made by Vapnik. The main idea is to map the training samples in input space to a high dimensional feature space by a nonlinear mapping function $\boldsymbol{\varphi}(\boldsymbol{x})$, and then construct the linear separating hyperplane in this high dimensional feature space. The problem of finding the optimal hyperplane is as in (6) below:

$$\begin{aligned} &\underset{\boldsymbol{w},b,\boldsymbol{\xi}}{Min}\ Z = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{k=1}^{N}\xi_k \\ &subject\ to: y_k[\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k)+b] \geq 1-\xi_k,\ k=1,\ldots,N \\ &\qquad\qquad \xi_k \geq 0,\ k=1,\ldots,N \end{aligned} \tag{6}$$

To solve the problem of (6), the following Lagrangian function can be constructed:

$$\begin{aligned} &\min_{\boldsymbol{w},b,\boldsymbol{\xi}} L(\boldsymbol{w},b,\boldsymbol{\xi};\boldsymbol{\alpha},\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{k=1}^{N}\xi_k \\ &-\sum_{k=1}^{N}\alpha_k\{y_k[\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k)+b]-1+\xi_k\} - \sum_{k=1}^{N}\lambda_k\xi_k \end{aligned} \tag{7}$$

where $\alpha_k, \lambda_k, k=1,\ldots,N$ are non-negative Lagrange multipliers.

With the Karush-Kuhn-Tucker (KKT) conditions derived from (7), the problem becomes:

$$\max L(\alpha_k) = -\frac{1}{2}\sum_{k,l=1}^{N}\alpha_k\alpha_l y_k y_l K(\boldsymbol{x}_k,\boldsymbol{x}_l) + \sum_{k=1}^{N}\alpha_k$$

Subject to

$$\sum_{k=1}^{N}\alpha_k y_k = 0$$

$$0 \leq \alpha_k \leq C, k=1,...,N.$$

where α_k are the Lagrange multipliers, function $K(\boldsymbol{x}_k,\boldsymbol{x}_l) = \boldsymbol{\varphi}(\boldsymbol{x}_k)^T\boldsymbol{\varphi}(\boldsymbol{x}_k)$ is the kernel function. Some typical kernel functions are:

Linear: $K(\mathrm{x},\mathrm{x}_k) = \mathrm{x}^T\mathrm{x}_k$,

Polynomial: $K(\mathrm{x},\mathrm{x}_k) = (\mathrm{x}^T\mathrm{x}_k+1)^d$,

Radial-basis function network (RBF): $K(\mathrm{x},\mathrm{x}_k) = \exp\left(-\|\mathrm{x}-\mathrm{x}_k\|^2/(2\sigma^2)\right)$.

Based on the solution of the above quadratic programming problem, the nonlinear SVM classifier can be defined as:

$$y(\boldsymbol{x}) = sign\left[\sum_{k=1}^{N}\alpha_k y_k K(\boldsymbol{x}, \boldsymbol{x}_k) + b\right]$$

2.2 Least Squares SVM (LSSVM)

The least square support vector machine was proposed by Suykens et al. (2002). The optimization problem, i.e. finding the optimal hyperplane that can separate the two different classes with maximum margin in the feature spaces for LSSVM, is as follows.

$$\begin{aligned} &\underset{\boldsymbol{w},b,\xi}{Min}\, Z = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\sum_{k=1}^{N}\xi_k^2 \\ &subject\ to: y_k\left[\boldsymbol{w}^T\varphi(\boldsymbol{x}_k) + b\right] = 1 - \xi_k,\ k = 1, \ldots, N. \end{aligned} \tag{2.18}$$

To solve it, the following Lagrangian function can be constructed:

$$\min L(\boldsymbol{w}, b, \xi; \alpha) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{C}{2}\sum_{k=1}^{N}\xi_k^2$$
$$-\sum_{k=1}^{N}\alpha_k\left\{y_k\left[\boldsymbol{w}^T\boldsymbol{\varphi}(\boldsymbol{x}_k) + b\right] - 1 + \xi_k\right\}$$

where α_k are the Lagrange multipliers. One can obtain the following KKT system:

$$\begin{bmatrix} 0 & y^T \\ y & \boldsymbol{\Omega} + V_C \end{bmatrix}\begin{Bmatrix} b \\ \boldsymbol{\alpha} \end{Bmatrix} = \begin{Bmatrix} 0 \\ \boldsymbol{E} \end{Bmatrix}$$

where $\boldsymbol{\Omega} = \boldsymbol{U}\boldsymbol{U}^T$ with element

$$\begin{aligned} \Omega_{kl} &= y_k y_l \varphi(\mathrm{x}_k)^T \varphi(\mathrm{x}_l) \\ &= y_k y_l K(\mathrm{x}_k, \mathrm{x}_l), k, l = 1, \ldots, N. \end{aligned}$$

and $K(\cdot, \cdot)$ is the kernel function.

The classifier of LSSVM in the dual space is:

$$y(\boldsymbol{x}) = sign\left[\sum_{k=1}^{N}\alpha_k y_k K(\mathrm{x}, \mathrm{x}_k) + b\right]$$

3. Parameters and Features Optimization Methods

Suppose the number of input features of the training dataset is m, then the features selection vector is $\beta^T = \{\beta_1, \beta_2, \ldots, \beta_m\}$, $\beta_k \in \{0, 1\}$. Let the number of parameters in kernel function $K_\theta(\cdot,\cdot)$ be n_p, where $\theta = \{\theta_1, \ldots, \theta_{n_p}\}$. So, the parameters that need to be determined are β, θ, C, and the number of parameters is $m + n_P + 1$.

3.1 Genetic Algorithm Based Features Selection and Parameters Optimization for SVM (GA-SVM)

Several key issues about the GA for optimizing parameters in SVM models are discussed as follows.

(1) The structure of chromosomes

In our GA-SVM model, the binary genetic algorithm is adopted. In the binary GA, each variable that needs to be optimized is denoted by a series of binary bits with value of either 1 or 0. Each variable corresponds to a gene and the set of all variables forms the chromosome. The binary GA works with bits. However, the fitness function in GA is always defined on the variables; hence, there must be a way to convert the chromosome into a continuous variable, and vice versa. The length of a gene is determined by the precision required of its corresponding variable. A gene can be denoted by $\left[b_1\, b_2 \ldots b_{N_g} \right]$, where $b_i \in \{0,1\}$ and N_g is the length of the gene.

In GA-SVM model, the length of genes of all parameters in the SVM model, such as error penalty constant C and σ in RBF kernel function, are set at the same value, N_{gp}, with precision requirement of $1/2^{N_{gd}}$ ($N_{gd} < N_{gp}$). Thus the length of bits that express the integer part of the parameters is $N_{gp} - N_{gd}$. The chromosome is the array of the gene. The structure of the chromosome in GA-SVM is:

$$\beta_1\beta_2\cdots\beta_m \underbrace{b_1^1 b_2^1 \ldots b_{N_{gp}}^1}_{gene_{p1}} \cdots \underbrace{b_1^{n_F} b_2^{n_F} \ldots b_{N_{gp}}^{n'_p}}_{gene_{pn'_p}}$$

where n'_p is the number of total parameters in the support vector machines model, m is the number of input features, b_i^k is the i^{th} bit of gene of the k^{th} parameter.

The parameters are decoded from genes as follows.

$$\theta_i = b_1^i \cdot 2^{(N_{gp}-N_{gd}-1)} + b_2^i \cdot 2^{(N_{gp}-N_{gd}-2)} + \cdots + b_{(N_{gp}-N_{gd})}^i \cdot 2^0 + b_{(N_{gp}-N_{gd}+1)}^i \cdot 2^{-1} + \cdots + b_{N_{gp}}^i \cdot 2^{-N_{gd}}$$

For example, let $N_{gd} = 5$, and the gene of the parameter is [1011001010]; then the value of this parameter is 21.3125. The parameters in the model have the same mechanism to code and decode the gene.

(2) Fitness function

Fitness function is a function of a chromosome whose value is always used to measure the performance of a group of parameters that the chromosome corresponds to. Fitness function can be chosen in terms of needs of business objectives. The fitness function is the classification error of the SVM model, with corresponding parameters in a chromosome, if the business objective is to maximize the classification accuracy.

(3) Stop criteria

The stop criteria are used to determine when the evolution process will be stopped. The common stop criteria include ((MATHWORKS 2006): (1) when the number of generations that the GA produces exceeds the predetermined number MAXGEN; (2) when the interval of computational time during which there is no improvement in the best fitness value, in seconds, exceeds a specified value TNOIMPROVE; and (3) when weighted average change in the fitness function value, over STALLGEN generations, is less than the specified function tolerance FTOL. There are some other stop criteria which can be defined on some indicators from the fitness function. The stop criteria control the computational time and efficiency of the genetic algorithm. Most of the parameters in the criteria are chosen by users' experience. Fortunately, performance of the GA is not

sensitive to small or even large deviations of these parameters, over a wide range of values.

With the above setting, the GA-SVM algorithm can be described as follows.

Suppose the samples set is $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \ldots, (\boldsymbol{x}_N, y_N)\}$, and the size of the samples set is $N = \sum_{t=t_1}^{t_L} N_t$, where N_t denote the number of samples from year t. To test the performance of the samples set in year t_i $(1 < i \leq L)$, denoted by S_i, samples from year t_1 to t_i-1 will consist of training samples S_r. Parameters optimization and features selection are based on the average performance of k-fold validation on the training samples. The years that will be tested are from t_{i_0} to t_L; the algorithm for building a hybrid classifier based on GA and SVM is:

Step 0. $i=i_0$;

Step 1. While $i \leq L$

Initialize the populations of chromosomes which encode the selection vector of input features, and the parameters for the SVM model;

Step 2. For each chromosome in the population**:**

2.1. Decode the features selection vector and parameters for SVM model from the chromosome, and get β, θ, C; evenly divide training sample indices set S_r into k-fold subset S_r^j, $j = 1,\ldots,k$, let $j = 1$;

2.2. While $j \leq k$,

Select all samples S_r^j to construct the validation sample set $S'_{validation}$, with other samples in training samples S_r forming the training sample set $S'_{training} = S_r - S'_{validation}$, using $S'_{training}$ to train the SVM model with parameters β, θ, C, and using $S'_{validation}$ to validate the performance of the SVM model; denote the performance by f'_j;

$j = j + 1$.

2.3. Average value of the performance of f'_j, $j = 1,..., k$ is taken as the fitness of the chromosome.

Step 3. Evolution processing: Select the elite chromosomes having higher fitness values and pass them to the next generation directly, perform mutation and crossover operation on the population, and generate a new population.

Step 4: Go back to Step 2, until the stop criteria are met.

Step 5. Optimization parameters are $\beta^r = \beta, \theta^r = \theta, C^r = C$, from training set S_r, performance of SVM with these optimization parameters on S_i is f_i, $i = i + 1$, go to Step 1.

Step 6. Overall performance value is $\frac{1}{L - i_0 + 1} \sum_{n=i_0}^{L} f_n$.

3.2 Direct Search Based Features Selection and Parameters Optimization for SVM (DS-SVM)

Direct search methods are simple and easy to implement and can be applied almost immediately to many nonlinear optimization problems, especially problems with lower dimension search space. Zhou et al. (2008), Zhou and Lai (2012) introduced direct search to parameters optimization and features selection in SVM models.

Suppose the samples set is $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2) \ldots, (\boldsymbol{x}_N, y_N)\}$, and the size of the samples set is $N = \sum_{t=t_1}^{t_L} N_t$, where N_t denote the number of samples from year t. The features set is $F = \{F_1, F_2, \ldots, F_m\}$, it is ranked by the features ranking method, and the features set in order is $F' = \{F_1', F_2', \ldots F_m'\}$, where $F_k' \in F, k = 1, 2, ..., m$. The top $N_{F'}$ features are selected for model training. Thus, the parameters that need to be optimized are $N_{F'}$, θ and C. So the total number of parameters is $1 + n_P + 1$. The employed features ranking methods in this study include:

(1) *t-test based features ranking (t-test)*

Suppose $v_j^{+1} = \text{var}(x_{ij} \mid y_i = +1)$, $v_j^{-1} = \text{var}(x_{ij} \mid y_i = -1)$, where var(·) is the variance of a group of values, $m_j^{+1} = mean(x_{ij} \mid y_i = +1)$, $m_j^{-1} = mean(x_{ij} \mid y_i = -1)$, where mean(·) is the mean of a group of values, and features weighting strategy, based on t-test on the training dataset, is defined as follows (Guyon et al. 2006):

$$z_j = \frac{\left| m_j^{+1} - m_j^{-1} \right|}{\sqrt{v_j^{+1} / n^+ + v_j^{-1} / n^-}}$$

(2) *ROC based features ranking (ROC)*

The idea of ROC based features ranking is to rank the features in terms of area under the convex hull of the ROC curve. The ROC curve can be easily constructed by sweeping the threshold and computing percentages of wrong and correct classifications over the available training feature vectors (Theodoridis and Koutroumbas 2003).

There are $n_P + 2$ parameters that need to be optimized, but only parameters included in SVM models, i.e. θ and C, are optimized by direct search. Since the number of selected top features has great effect on the convergence of the search process, it is optimized by grid search with fixed steps. Let $n = n_P + 1$, a point $\boldsymbol{p}$ in this space can be denoted by $(d_1, d_2, ..., d_n)$. Pattern $\boldsymbol{v}$, which is a collection of vectors, used to determine which points should be searched in terms of the current point. $\boldsymbol{v} = [v_1, v_2, ..., v_{2n}]$, v_1=[1,0,…,0], v_2=[0,1,0,…,0], …, v_n=[0,0,…,1], v_{n+1}=[-1,0,…,0], v_{n+2}=[0,-1,0,…,0], v_{2n}=[0,…,0,1], $v_i \in R^n, i = 1, 2, ..., 2n$. The point set around current point $\boldsymbol{p}$ to be searched is defined by the coordinate of the current point and the pattern mesh, which is determined by pattern vectors $\boldsymbol{v}$ and a scalar e called the mesh size. The mesh can be denoted by the points set M={m_1, m_2, …, m_{2n}}. The evaluation function of each point, denoted by g(.), is defined as the negative value of average classification accuracy, from k-fold cross validation on training samples set S_r. If there is at least one point in the mesh whose objective function value is less than that of the current point, we call the poll successful. The status of the poll is denoted by *Flag*. The algorithm of direct search based features selection and parameters optimization for SVM models (DS-SVM) is as follows:

Step 0. i=i_0;

Step 1. While $i \leq L$

The testing samples set S_i consists of samples in year t_i ($1 < i \leq L$), samples from year t_1 to t_i-1 will consist of the training samples set S_r. Rank all the input features in S_r

with features ranking technology and get the ranked features set $F' = \{F_1', F_2', \ldots F_m'\}$.

$N_{F'} = 1$, H={};

Step 2. while $N_{F'} \leq |F|$

Step 3. Set the initial current point $\boldsymbol{p}$, $\boldsymbol{p}^* = \boldsymbol{p}$, $h^* = g(\boldsymbol{p})$, $e = 1$;

Step 4. Form the mesh $M=\{m_1, m_2, \ldots, m_{2n}\}$ of current point $\boldsymbol{p}$, Flag = FALSE, $k=1$,

Step 5. while (k ≤2n), do

$h = g(m_k, N_{F'})$;

if ($h < h^*$), then

$h^* = h$; $\boldsymbol{p}^* = m_k$; $\boldsymbol{p} = m_k$; Flag=TRUE; break;

end if

$k = k + 1$

end while

Step 6. If *Flag*==TRUE, $e = 2 \times e$; else e =1/2×e; go to Step 4 until the stop criteria is met.

Step 7. Optimization parameters vector are $\boldsymbol{p}^*$, from training set S_r with top $N_{F'}$ selected features. H={H, $g(\boldsymbol{p}^*, N_{F'})$}; $N_{F'} = N_{F'} + \Delta N_{F'}$, go to Step 2.

Step 8. Find the minimum element in H and get the optimal $N_{F'}^* = N_{F'}$.

Step 9. The performance of SVM with these optimization parameters $N_{F'}^*$ and $\boldsymbol{p}^*$ on S_i is f_i, $i = i + 1$, go to Step 1.

Step 10. Overall performance value is $\frac{1}{L - i_0 + 1} \sum_{n=i_0}^{L} f_n$.

The stop criteria is met when any of the following criteria occurs (MATHWORKS, 2006): (1) The mesh size is less than the predetermined mesh tolerance denoted by Mesh_tol; (2) The number of iterations performed by the algorithm reaches the value of predetermined maximum iteration Max_iter; (3) The total number of objective function evaluations performed by the algorithm reaches the value of Max function evaluations Max_fun; (4) The distance between the point found at one successful poll and the point found at the next successful poll is less than predetermined tolerance X_tol. (5) The

change in the objective function from one successful poll to the next successful poll is less than function tolerance Fun_tol.

The process to calculate evaluation function g(.) is defined as follows.

Step 1. Select the top $N_{F'}$ features in S_r samples set, and then evenly divide training sample indices set S_r into k-fold subset S_r^j, $j = 1,\ldots,k$, let $j = 1$;

Step 2. While $j \leq k$,

select all samples S_r^j to construct the validation sample set $S'_{validation}$, with other samples in training samples S_r forming the training sample set $S'_{training} = S_r - S'_{validation}$, using $S'_{training}$ to train the SVM model with parameters from vector $\boldsymbol{p}$ and uses $S'_{validation}$ to validate the performance of the SVM model, and denotes the performance by f'_j;

$j = j + 1$.

Step 3. The negative average value of performance of f'_j, $j = 1,..., k$ is taken as value of $g(\boldsymbol{p})$.

4. Empirical Study

4.1 Dataset

The companies used in this study are selected from Compustat North America, Wharton research data service. Companies or firms become inactive or disappear due to several reasons, such as merger, bankruptcy, liquidation, etc. Only firms with reason of deletion as bankruptcy or liquidation are considered to be insolvent companies in this study. There are total 1,219 insolvent firms and 6,509 solvent firms in the dataset for this study, covering the period 1980-2006.

Insolvent firms in the sample are those for which all valid values are available for calculating all the ratios listed in Table 1, for the year preceding failure. In bankruptcy prediction studies, it is usual to select insolvent firms from various years, in order to have a larger sample size since that can help the model capture the pattern of insolvency effectively. So, there were 1005 firms that had failed in year *t*, which is between 1981 and 2006; full information about variables R1-R27 for the previous year *t*-1 was available

in respect of these firms. The number of insolvent firms in each year in the samples set is shown in Figure 1. Then solvent firms are randomly selected, from among all solvent firms with full and valid figures for ratios listed in Table 1. Samples distribution in different years is the same for both insolvent and solvent firms, as shown in Figure 1. Thus there are 1005 insolvent firms and an equal number of solvent firms, making a total of 2010 observations in the final data set. The ratios R1-R27 are selected on the basis of comprehensive review from (Ravi Kumar and Ravi 2007), and two criteria: (1) They are used by more than 3 of the 128 reviewed literatures about bankruptcy prediction, and (2) The information required to calculate each ratio should be available in the data set. Some of the ratios are highly correlated, and all ratios are selected automatically in the GA-SVM and DS-SVM models, based on performance of validation samples. For the single benchmarking models, the highly correlated ratios (with correlation coefficient greater than 0.9) are removed. Ratios highly correlated to other ratios include R3, R5, R6, R7, R9 and R11, all of which are removed.

Table 1. Selected ratios for bankruptcy prediction

No.	Description	No.	Description
R1	Net income/total assets (ROA)	R15	Cash flow/total assets
R2	Current ratio [a]	R16	Net income/stockholders' equity
R3	Retained earnings/total assets	R17	Current liabilities/total assets
R4	Working capital/total assets	R18	Net income/sales
R5	EBIT/total assets	R19	Total liabilities/total assets
R6	Sales/total assets	R20	Size
R7	Quick ratio [b]	R21	Net income/net worth
R8	Cash flow/total debt	R22	Quick assets/sales
R9	Current assets/total assets	R23	Sales/cash
R10	Quick assets/current liabilities	R24	Working capital/sales
R11	Total debt/total assets	R25	Dividend
R12	Cash/total assets	R26	Fixed assets/(stockholder's equity + long-term liabilities)
R13	Quick assets/total assets	R27	Stock holders' equity/total assets
R14	Total assets		

[a] Current ratio = Current assets/current liabilities

[b] Quick ratio = (cash and cash equivalent + marketable securities + accounts receivable)/current liabilities

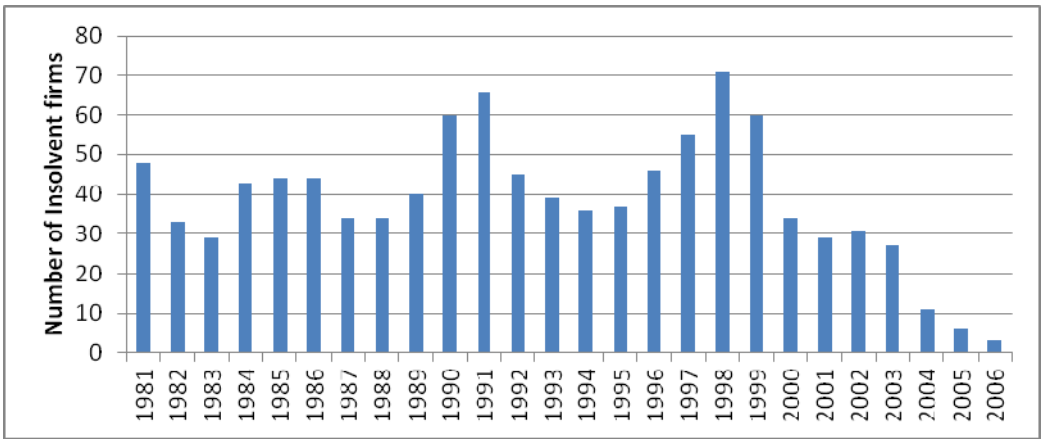


Figure 1. Selected insolvent firms distribution in years 1981-2006

4.2 Experimental results

Three common performance measures are selected, as follows.

$$\text{Sensitivity (Sen)} = \frac{SS}{SI + SS}$$

$$\text{Specificity (Spe)} = \frac{II}{II + IS}$$

$$\text{Percentage correctly classified (PCC)} = \frac{SS + II}{SS + SI + II + IS}$$

where *SS*: Solvent classified as solvent, *SI*: solvent classified as insolvent, *II*: insolvent classified as insolvent, *IS*: insolvent classified solvent.

Some parameters in stop criteria for GA-SVM are set as: Population Size = 30, MAXGEN = 50, STALLGEN = 50; other parameters use the default values in Genetic Algorithm and Direct Search Toolbox (GADS) of MATLAB 2008. Three main parameters for controlling the stop criteria in DS-SVM are set as follows: Mesh_Tol = 0.05, Fun_tol = 0.01 and $N_{F'} = 5$. Other parameters in DS-SVM are the default values,

as defined in GADS. The value of k of k-fold in evaluation function $g()$ is set to 5. These two different types of methods, with features selection and parameters optimized, are compared with other classic methods, such as linear discriminant analysis (LDA), logistic regression (LOG), backpropagation neural networks models (BP) with three layers, Satlins transfer function in hidden layer, and linear transfer function in output layer (BPSat), BP model with Tan-Sigmoid transfer function in hidden layer and linear transfer function in output layer (BPsig), and decision tree C4.5 (DTC4.5). GA-SVM and DS-SVM are also compared to SVM models with parameters optimization but without features selection. The SVM model is implemented with the toolbox LibSVM proposed by Chang and Lin (2001), and the LSSVM is implemented with the LS-SVM toolbox proposed by Suykens et al. (2002). The libSVM model with RBF kernel function is denoted by libSVMrbf, and LSSVM models with linear kernel function and RBF kernel function are denoted by LSSVMlin and LSSVMrbf. Since the lower computational efficiency of libSVM with linear kernel function results in libSVMrbf always outperforming it, libSVM with linear kernel function is not tested in this study. The GA-SVM with libSVM model, and LSSVM model of RBF kernel function, are denoted by GASVM-librbf and GASVM-lsrbf, respectively. GA-SVM with LSSVM model of linear kernel function is denoted by GASVM-lslin. Likewise, the three corresponding DS-SVM models with t-test based features selection method are denoted by DSSVM-tlibrbf, DSSVM-tlsrbf and DSSVM-tlslin, and those with ROC based features selection method are denoted by DSSVM-rlibrbf, DSSVM-rlsrbf and DSSVM-rlslin.

The test samples are selected from fiscal year t, between 1996 and 2006. To analyze the time effect of training samples, the experiments are conducted in three different scenarios, with training samples from different periods. Training samples in the first scenario consist of all samples with observed results in and before year t-1, and after year 1980. So it is obvious that the training samples size is increasing as t increases because of accumulation of historical samples. Those in the second scenario consist of samples from year t-1 to year t-10, and those in the third scenario consist of samples from year t-1 to t-5. The average performance of all test samples from year 1996 to 2006, for the three scenarios is shown in Tables 2, 3 and 4. All PCCs greater than 0.75 are in bold.

Table 2. Average performance of test samples from observed year *t* in 1996-2006 while taking all samples before the observed year to be training sample set (Scenario 1)

Models	Sen	Spe	PCC	Models	Sen	Spe	PCC
LDA	0.7802	0.6542	0.7172	GASVM-librbf	0.7453	0.7802	**0.7627**
QDA	0.9357	0.3190	0.6273	GASVM-lslin	0.7694	0.7105	0.7399
LOG	0.7828	0.6971	0.7399	GASVM-lsrbf	0.7855	0.7507	**0.7681**
BPSat	0.7507	0.7614	**0.7560**	DSSVM-tlibrbf	0.8097	0.6300	0.7198
BPSig	0.7426	0.7292	0.7359	DSSVM-tlslin	0.7158	0.7534	0.7346
*k*NN-10	0.6434	0.5764	0.6099	DSSVM-tlsrbf	0.8177	0.6434	0.7306
DTC4.5	0.9276	0.0858	0.5067	DSSVM-rlibrbf	0.8525	0.6434	0.7480
libSVMrbf	0.6783	0.5174	0.5979	DSSVM-rlslin	0.7614	0.7426	**0.7520**
LSSVMlin	0.7802	0.6568	0.7185	DSSVM-rlsrbf	0.7802	0.7614	**0.7708**
LSSVMrbf	0.7694	0.7668	**0.7681**				

Table 3. Average performance of test samples from observed years 1996-2006 while taking samples from *t*-1 to *t*-10 to be training samples set (Scenario 2)

Models	Sen	Spe	PCC	Models	Sen	Spe	PCC
LDA	0.7775	0.6139	0.6957	GASVM-librbf	0.7587	0.7399	0.7493
QDA	0.9303	0.3405	0.6354	GASVM-lslin	0.7855	0.7292	**0.7574**
LOG	0.6542	0.7641	0.7091	GASVM-lsrbf	0.7936	0.7587	**0.7761**
BPSat	0.7426	0.7587	**0.7507**	DSSVM-tlibrbf	0.8177	0.7480	**0.7828**
BPSig	0.6729	0.7399	0.7064	DSSVM-tlslin	0.7507	0.7185	0.7346
*k*NN-10	0.6783	0.5871	0.6327	DSSVM-tlsrbf	0.7855	0.6381	0.7118
DTC4.5	0.9357	0.0563	0.4960	DSSVM-rlibrbf	0.8338	0.6595	0.7466
libSVMrbf	0.6649	0.4906	0.5777	DSSVM-rlslin	0.8043	0.7668	**0.7855**
LSSVMlin	0.7748	0.6220	0.6984	DSSVM-rlsrbf	0.7855	0.7587	**0.7721**
LSSVMrbf	0.7989	0.7480	**0.7735**				

Table 4. Average performance of test samples from observed year 1996-2006 while taking samples from *t*-1 to *t*-5 to be training samples set (Scenario 3)

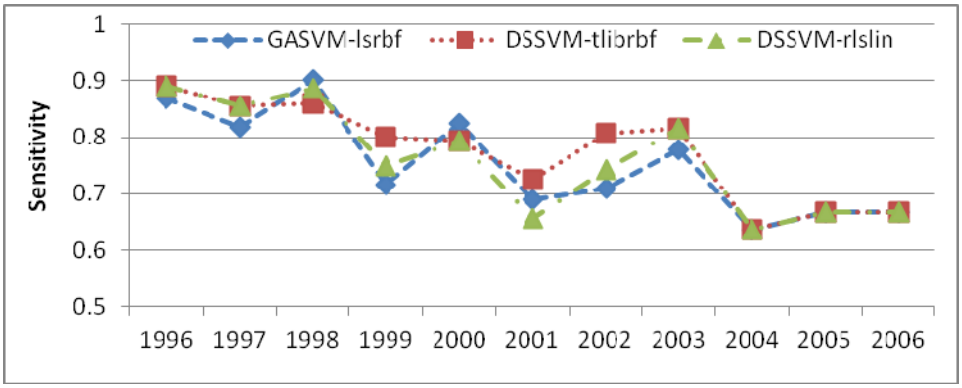
Models	Sen	Spe	PCC	Models	Sen	Spe	PCC

Models	Sen	Spe	PCC	Models	Sen	Spe	PCC
LDA	0.7212	0.6220	0.6716	GASVM-librbf	0.7721	0.7641	**0.7681**
QDA	0.9062	0.4397	0.6729	GASVM-lslin	0.7587	0.7105	0.7346
LOG	0.7212	0.7346	0.7279	GASVM-lsrbf	0.7909	0.7346	**0.7627**
BPSat	0.7560	0.7399	0.7480	DSSVM-tlibrbf	0.7668	0.7587	**0.7627**
BPSig	0.7158	0.7212	0.7185	DSSVM-tlslin	0.7721	0.7319	**0.7520**
*k*NN-10	0.5603	0.6273	0.5938	DSSVM-tlsrbf	0.7534	0.6783	0.7158
DTC4.5	0.9625	0.0295	0.4960	DSSVM-rlibrbf	0.8150	0.6461	0.7306
libSVMrbf	0.6783	0.4155	0.5469	DSSVM-rlslin	0.7775	0.7399	**0.7587**
LSSVMlin	0.7158	0.6273	0.6716	DSSVM-rlsrbf	0.7909	0.7480	**0.7694**
LSSVMrbf	0.7694	0.7480	**0.7587**				

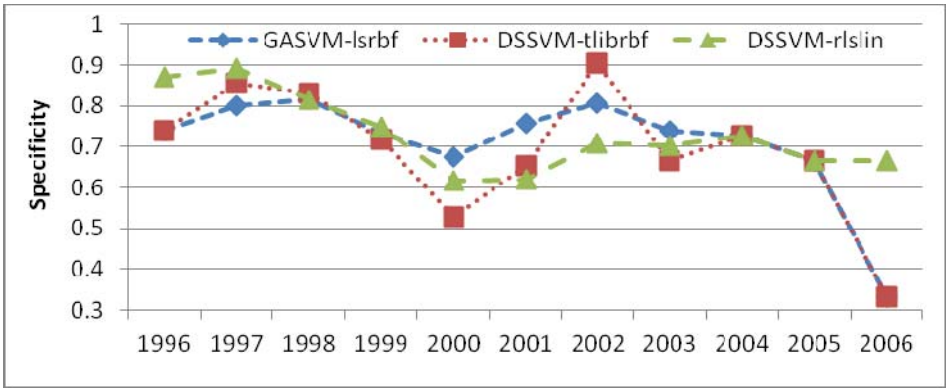
From Tables 2 to 4, it can be observed that: (1) All neural networks models and SVM based models can strike a good balance between sensitivity and specificity, but QDA and DTC4.5 have good sensitivity performance but very bad specificity, which means that QDA and DTC4.5 classify almost all samples as solvent firms. (2) PCC performance of the BPsat, LDA model decreases significantly as the size of training samples decreases. However, most SVM-based models can still keep higher performance as the size of training samples decreases. It demonstrates that SVM models can keep good performance with small training samples, which has been proved in many other applications also. (3) Theoretically, if the stop criteria of GA guided features selection and parameters optimization in SVM model are set properly, GA-SVM can get the same features and parameters as are selected or optimized by the direct search method. However, the ROC features ranking based LSSVM models, with parameters optimized by direct search, i.e. DSSVM-rlslin, DSSVM-rlsrbf, slightly outperform the GASVM-lslin and GASVM-lsrbf in most cases. (4) In the three scenarios, the LSSVMrbf model is the only one among the 10 single models that can consistently provide PCC performance greater than 75%. Among the 9 hybrid models with feature selection, DSSVM-rlslin, DSSVM-rlsrbf, and GASVM-lsrbf are able to offer PPC performance greater than 75%. Although all these three hybrid models are based on LSSVM, it is difficult to draw a conclusion that LSSVM based hybrid models have better performance than LibSVM based models. However, from the perspective of computational time, LSSVM models are

better since they consume less computational time. (5) The three top models (DSSVM-rlslin, DSSVM-rlsrbf, and GASVM-lsrbf) achieve their best performance in the scenario where samples from *t*-1 to *t*-10 are taken to be training samples set (*t* is the observed year of test samples). Thus it shows that more training samples does not mean improvement of performance of models, and at the same time, less training samples may not be helpful to improve the performance of models. It may depend on the pattern of bankruptcy in test and training samples. If the training samples are enough to cover all the patterns in test samples, and the model is powerful enough to capture the patterns, then the performance of the test samples could be good. If the size of training samples is so big that the model is trained overfittingly, or prefers some special patterns, the performance of the test samples may be bad. In this experiment, samples from 10 years before the concerned year are a good choice.

Since both DSSVM-rlslin and DSSVM-tlibrbf produce the best PCC performance when taking samples from 10 years before the year of the bankruptcy event, among the three scenarios, it is important to explore their features selection results and compare them with GASVM-lsrbf, which gets the best PCC performance among the three GA based hybrid models in this scenario. The Sen and Spe performance of the above mentioned three models is shown in Figure 2. Since the number of samples from solvent and insolvent groups in each tested year is the same, the PCC performance is actually the mean of Sen and Spe values. The test samples cover 11 years and the training samples are from rolling time-windows, so the selected features for each year and each model are different. Figure 3 shows the distribution of number of features that are employed from the three models: GASVM-lsrbf, DSSVM-tlibrbf and DSSVM-rlslin. It is interesting to see that DSSVM-tlibrbf constantly selects the top 5 features, based on the performance on the training dataset; however, it does not mean that t-test based features selection models prefer less features since we explore results of DSSVM-tlslin and DSSVM-tlsrbf models and find that both select different numbers of features for each year and the average number of features selected by DSSVM-tlsrbf is greater than 20. In addition, although the number of selected features is 5 for DSSVM-tlibrbf, different features are selected in different years.



(a) Sensitivity performance



(b) Specificity performance

Figure 2. Sensitivity and specificity performance of three selected models

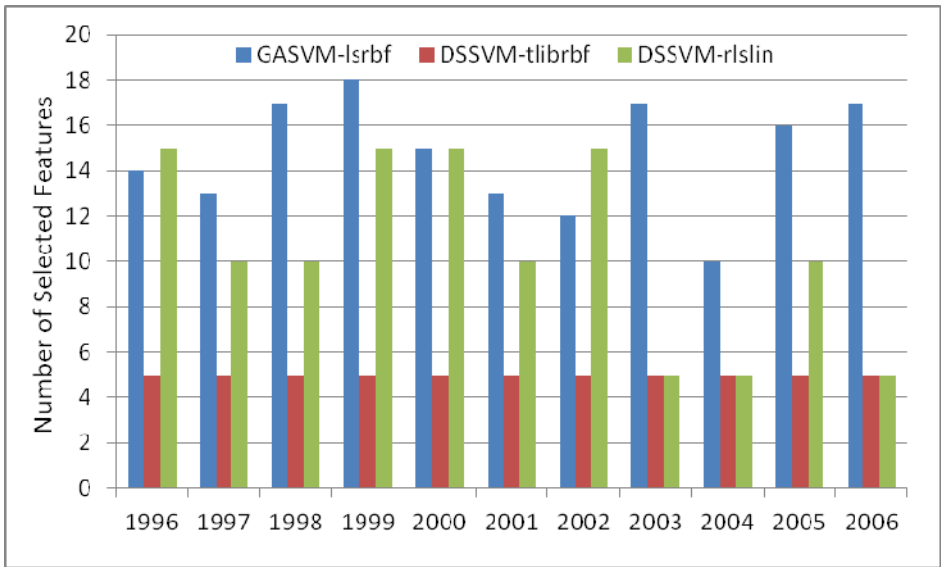


Figure 3. Number of selected features from three selected models

The main drawback of powerful nonlinear models, such as neural networks and SVM, is that they are black-boxes; it is difficult to explain their predictions in terms of the factors. The proposed models in this study also have the same problem, and this problem is still open. From the perspective of finance and economics, the financial market and macro-environment are always changing. Moreover, positions of the concerned firms in their respective industry segments are also changing constantly; it is natural that the important factors that affect a firm may change over time. Although it is still difficult to determine what factors are important in what kind of macro-environment, the features that are selected by the three models, with powerful bankruptcy prediction capability, may be helpful to get some ideas about the black-boxes, to some degree. Figure 4 shows the histogram of the 27 features selected by the three models. We can find that features that are selected most frequently are:

(1) R1: Net income/total assets (ROA) ;

(2) R5: EBIT/total assets;

(3) R11: Total debt/total assets;

(4) R17: Current liabilities/total assets;

(5) R19: Total liabilities/total assets;

(6) R27: Stock holders' equity/total assets;

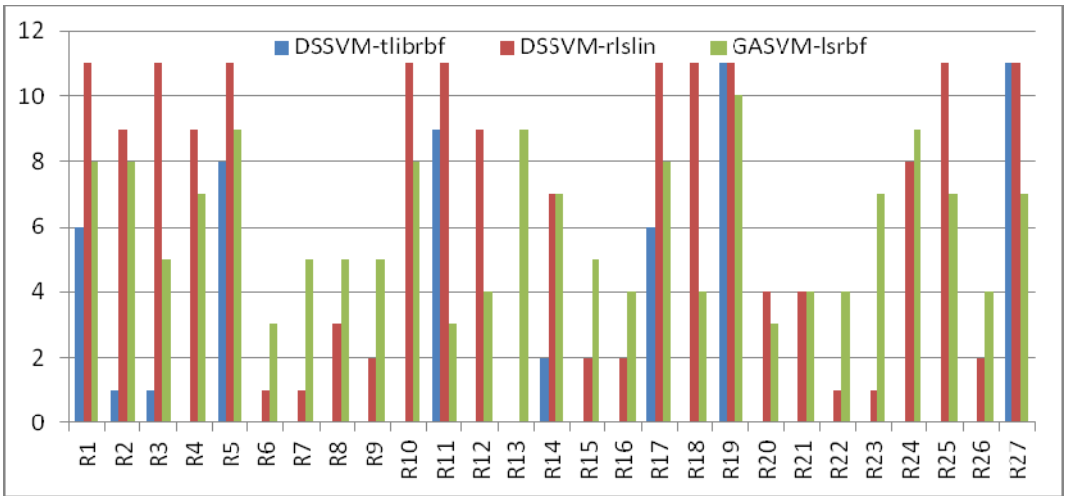


Figure 4. Distribution of features selected by the three most powerful models.

All the experiments are conducted on PC with Intel Core Duo CPU 2.33GHz and 2 GB RAM. Computational time of three GA-based models, and DSSVM-tlibrbf and

DSSVM-rlslin, for all test samples in each year, with training samples, was as in Figure 5. It can be observed that the standard deviation of computational time of GASVM-lsrbf and DSSVM-rlslin is small, meaning that their computational time is not sensitive to the change of training samples. Thus, for large scale problems, GASVM-lsrbf and DSSVM-rlslin may be the better choice.

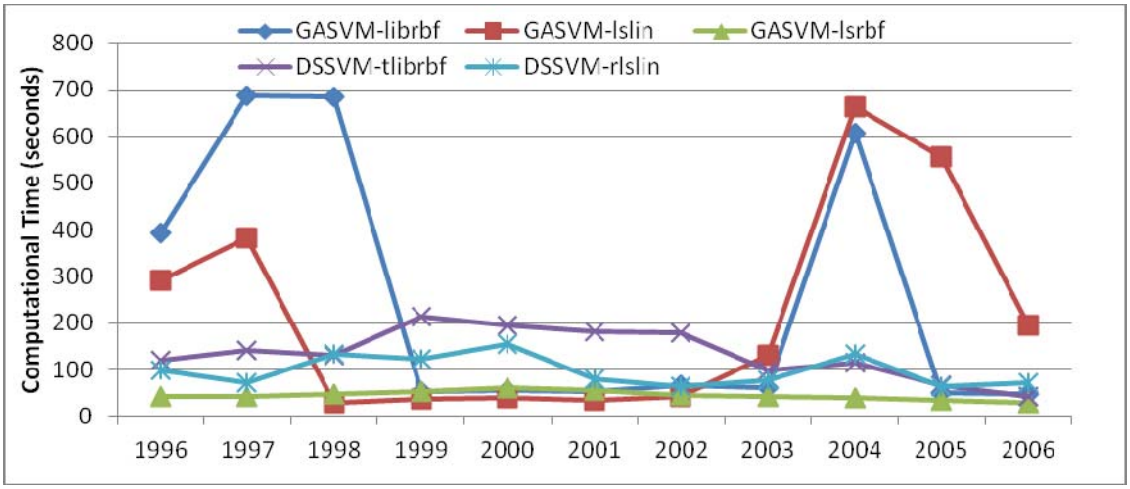


Figure 5. Computational time of 5 selected models for all test samples in each year

5. Conclusion

In this study, a new approach based on features ranking technologies and direct search is proposed to construct hybrid SVM models for bankruptcy prediction. This approach has been compared to the traditional genetic algorithm based feature selection and parameters optimization methods. Sensitivity analysis of training samples size, type of SVM models, and features ranking methodology are conducted on a data set comprising 2010 non-financial companies' financial records, covering from year 1980 to 2006. From the experimental results, DSSVM-rlslin, DSSVM-rlsrbf and GASVM-lsrbf are the three hybrid modes that can consistently keep high performance with average PCC greater than 75% with different training samples selection strategies. Moreover, these hybrid models outperform the traditional single models, such as linear discriminant analysis, logistic regression, neural network, decision tree and *k*-nearest neighbor. However, these hybrid models consume more computational time.

Although the DSSVM-rlsin model achieves the best PCC (78.55%) when using training samples strategies as defined in Table 3, its PCC is only 1.2 percentage points higher than the GASVM-lsrbf model. When compared to genetic algorithm based features selection and parameters optimization SVM models, direct search guided SVM models have less parameters to control the optimization process; it is a deterministic method while genetic algorithm is a stochastic one. So when using GA based SVM models on the same training samples twice, we may get two different models, and the decision on the same test sample may also be different. This stochastic characteristic of this method may be unacceptable for the decision makers or the analysts.

Since the financial market, macro-environment and the business of a company are dynamic, the factors that should be employed for bankruptcy prediction may be different in different conditions. Hybrid SVM models with features selection for bankruptcy prediction select different factors in terms of performance on training samples. The selected factors include only factors derived from financial statements of the firms, without distinguishing between different industries. Are the factors for good bankruptcy prediction sensitive to the industry segment of the company? Can macro-economic information help improve prediction accuracy for companies in specific industry segments? These questions offer directions for future research.

Acknowledgements

We thank two reviewers for their comments and suggestions.

References

Agarwal, V., and Taffler, R., (2008), 'Comparing the performance of market-based and accounting-based bankruptcy prediction models', *Journal of Banking and Finance*, 32, 1541-1551.

Ahn, H., Lee, K., and Kim, K.J., (2006), 'Global optimization of support vector machines using genetic algorithms for bankruptcy prediction', in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 420-429.

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

Beaver, W.H., (1966), 'Financial ratios as predictors of failure', *Journal of Accounting Research*, 4, 71-111.

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

Chen, H.L., Yang, B., Wang, G., Liu, J., Xu, X., Wang, S.J., and Liu, D.Y., (2011), 'A novel bankruptcy prediction model based on an adaptive fuzzy k-nearest neighbor method', *Knowledge-Based Systems*, 24, 1348-1359.

Divsalar, M., Javid, M.R., Gandomi, A.H., Soofi, J.B., and Mahmood, M.V., (2011), 'Hybrid genetic programming-based search algorithms for enterprise bankruptcy prediction', *Applied Artificial Intelligence*, 25, 669-692.

Gomes, T.A.F., Prudĉncio, R.B.C., Soares, C., Rossi, A.L.D., and Carvalho, A., (2012), 'Combining meta-learning and search techniques to select parameters for support vector machines', *Neurocomputing*, 75, 3-13.

Guyon, I., Gunn, S., Nikravesh, M., and Zadeh, L.A., (2006), *Feature Extraction: Foundations and Applications*, Springer.

Härdle, W., Lee, Y.J., Schäfer, D., and Yeh, Y.R., (2009), 'Variable selection and oversampling in the use of smooth support vector machines for predicting the default risk of companies', *Journal of Forecasting*, 28, 512-534.

Hillegeist, S.A., Keating, E.K., Cram, D.P., and Lundstedt, K.G., (2004), 'Assessing the probability of bankruptcy', *Review of Accounting Studies*, 9, 5-34.

Jo, H., and Han, I., (1996), 'Integration of case-based forecasting, neural network, and discriminant analysis for bankruptcy prediction', *Expert Systems with Applications*, 11, 415-422.

MATHWORKS, (2006). 'Genetic algorithm and direct search toolbox: user's guide'. http://www.mathworks.com/

McKee, T.E., and Lensberg, T., (2002), 'Genetic programming and rough sets: A hybrid approach to bankruptcy classification', *European Journal of Operational Research*, 138, 436-451.

Min, S.H., Lee, J., and Han, I., (2006), 'Hybrid genetic algorithms and support vector machines for bankruptcy prediction', *Expert Systems with Applications*, 31, 652-660.

Nam, C.W., Kim, T.S., Park, N.J., and Lee, H.K., (2008), 'Bankruptcy prediction using a discrete-time duration model incorporating temporal and macroeconomic dependencies', *Journal of Forecasting*, 27, 493-506.

Premachandra, I.M., Chen, Y., and Watson, J., (2011), 'DEA as a tool for predicting corporate failure and success: A case of bankruptcy assessment', *Omega*, 39, 620-626.

Ravi Kumar, P., and Ravi, V., (2007), 'Bankruptcy prediction in banks and firms via statistical and intelligent techniques - A review', *European Journal of Operational Research*, 180, 1-28.

Shie, F.S., Chen, M.Y., and Liu, Y.S., (2011), 'Prediction of corporate financial distress: an application of the America banking industry', *Neural Computing and Applications*, DOI: 10.1007/s00521-011-0765-5

Shin, K.S., Lee, T.S., and Kim, H.J., (2005), 'An application of support vector machines in bankruptcy prediction model', *Expert Systems with Applications*, 28, 127-135.

Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., Moor, B.D., and Vandewalle, J., (2002), *Least Squares Support Vector Machines*, Siningapore: World Scientific.

Theodoridis, S., and Koutroumbas, K., (2003), *Pattern Recognition* (2nd), San Diego: Acadmic Press.

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

Vassalou, M., and Xing, Y., (2004), 'Default Risk in Equity Returns', *Journal of Finance*, 59, 831-868.

Verikas, A., Kalsyte, Z., Bacauskiene, M., and Gelzinis, A., (2010), 'Hybrid and ensemble-based soft computing techniques in bankruptcy prediction: A survey', *Soft Computing*, 14, 995-1010.

Wang, D.H., and Dillon, T.S. (2006), 'Extraction of classification rules characterized by ellipsoidal regions using soft-computing techniques', *International Journal of Systems Science,* 37, 969-980.

Wilson, R.L., and Sharda, R., (1994),' Bankruptcy prediction using neural networks', *Decision Support Systems*, 11, 545-557.

Wu, C.H., Tzeng, G.H., Goo, Y.J., and Fang, W.C., (2007), 'A real-valued genetic algorithm to optimize the parameters of support vector machine for predicting bankruptcy', *Expert Systems with Applications*, 32, 397-408.

Zhou, L., and Lai, K.K., (2012), 'Corporate Financial Crisis Prediction Using SVM Models with Direct Search for Features Selection and Parameters Optimization', in *International Joint Conference on Computational Sciences and Optimization*, pp. 760-764.

Zhou, L., Lai, K.K., and Yu, L., (2008), 'Credit scoring using support vector machines with direct search for parameters selection', *Soft Computing*,13, 149-155.