International Journal of Management Science
Vol 17, No 1, May 2011

Tuning the Architecture of Support Vector Machine: The Case of Bankruptcy Prediction*

Jae H. Min**
Sogang Business School, Sogang University

Chulwoo Jeong
Graduate School of Management of Technology, Sogang University

Myung Suk Kim
Sogang Business School, Sogang University

(Received: September 29, 2010 / Accepted: December 6, 2010)

ABSTRACT

Tuning the architecture of SVM (support vector machine) is to build an SVM model of better performance. Two different tuning methods of the grid search and the GA (genetic algorithm) have been addressed in the literature, each of which has its own methodological pros and cons. This paper suggests a combined method for tuning the architecture of SVM models, which employs the GAM (generalized additive models), the grid search, and the GA in sequence. The GAM is used for selecting input variables, and the grid search and the GA are employed for finding optimal parameter values of the SVM models. Applying the method to a bankruptcy prediction problem, we show that SVM model tuned by the proposed method outperforms other SVM models.

Keywords: Support Vector Machine, Generalized Additive Model, Grid Search Method, Genetic Algorithm

1. Introduction

Among the recent studies on applications of classification and prediction meth-

* The first author and the third author acknowledge that this work was supported by the Sogang University Research Grant of 2009 (200910047. 01) and the Sogang University Research Grant of 2010 (201010024. 01) respectively.

** Corresponding author, E- mail: jaemin@sogang.ac.kr

ods, it has been reported that the SVM (support vector machine) is one of the most powerful tools for bankruptcy prediction problems with respect to prediction performance. Several studies claim that the SVM even outperforms the ANN (artificial neural networks), another popular method for bankruptcy prediction [5, 12, 14, 20, 26].

This study suggests a new fine-tuning method for the architecture of SVM models in order to improve their prediction accuracy even more than ever. For tuning the architecture of SVM, we focus on two factors critical to the performance of SVM models, which are the input variable selection and the parameter value optimization.

Specifically, the suggested tuning method employs the GAM (generalized additive models), the grid search, and the GA in sequence. The three different methods are used for the following respective purposes: first, the GAM is used as a tool for the input variable selection; second, the grid search method initializes a part of population of the GA; and third, the GA optimizes the parameter values of the SVM model.

The suggested method is evaluated by applying it to a bankruptcy prediction problem, which analyzes the data of failed and solvent small- and medium-sized Korean firms from 2001 to 2004. In our empirical analysis, the SVM model tuned by the suggested method significantly outperforms the ones tuned by other existing methods.

This paper is organized as follows. Section 2 gives a concise description of SVM, and reports the existing tuning methods for the architecture of SVM models. Section 3 describes the suggested tuning steps. In addition, the GAM approach, the grid search, and the GA composing the tuning method are also introduced. Section 4 reports the empirical results that are applied to some bankruptcy data, where the superiority of the newly tuned SVM model is compared with existing SVM models. Summary and concluding remarks follow in Section 5.

2. Support Vector Machines

Support vector machine (SVM) has been attractive to academics as well as practitioners due to its methodological merits of simplicity of estimation and high prediction power. Since SVM was first introduced by Boser *et al*. [1], it has been applied to

numerous areas including computer science, bioinformatics, and financial problems [3, 6, 10, 13, 14, 21, 24].

2.1 Basic Algorithm of SVM

The basic algorithm of SVM is described as follows [22, 23]. In the linear separation problem, the function of SVM is to seek out a hyperplane in order to separate a set of positively and negatively labeled train data. The hyperplane is defined by $\mathbf{w}^T\mathbf{x}+b=0$, where the parameter $\mathbf{w}\in\mathbb{R}^m$ is a vector orthogonal to the hyperplane, and $b\in\mathbb{R}$ is the bias. The decision function is the hyperplane classifier

$$H(\mathbf{x})=sign\left(\mathbf{w}^T\mathbf{x}+b\right). \tag{1}$$

The hyperplane is designed such that $y_i\left(\mathbf{w}^T\mathbf{x}_i+b\right)\geq 1$, where $\mathbf{x}_i=(\mathbf{x}_i^{(1)}, \mathbf{x}_i^{(2)}, \cdots, \mathbf{x}_i^{(n)})^T\in\mathbb{R}^n$ is a train data point, and $y_i=\{-1, +1\}$ denotes the class of the vector $\mathbf{x}_i$. The margin is defined by the distance of the two parallel hyperplanes $\mathbf{w}^T\mathbf{x}+b=-1$ and $\mathbf{w}^T\mathbf{x}+b=+1$. Thus, the margin is calculated as $2/\|\mathbf{w}\|$. The margin is related to the concept of generalization of the classifier [22].

The SVM classifier is optimized by solving the following quadratic programming problem.

$$\begin{aligned}&\min\ \ \frac{1}{2}\mathbf{w}^T\mathbf{w}\\&\text{subject to}\\&y_i\left(\mathbf{w}^T\mathbf{x}_i-b\right)\geq 1,\qquad (i=1, 2, \cdots, N)\end{aligned} \tag{2}$$

Equation (2) can be changed into the following dual model:

$$\begin{aligned}&\max\ \ -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha}+\mathbf{1}^T\boldsymbol{\alpha}\\&\text{subject to}\\&\begin{cases}\alpha_i\geq 0,\qquad (i=1, 2, \cdots, N)\\ \mathbf{y}^T\boldsymbol{\alpha}=0\end{cases}\end{aligned} \tag{3}$$

where the variable $\boldsymbol{\alpha} \in \mathbb{R}^n$, the sample labels $\mathbf{y} \in \mathbb{R}^n$, the matrix $\mathbf{Q} \in \mathbb{R}^{n \times n}$, and $Q_{ij} = y_i y_j \mathbf{x}_i^T \mathbf{x}_j$.

The support vectors are defined as the subset of the training vectors with the non-zero dual multiplier α_i. By the complementary slackness condition, $\alpha_i[y_i(\mathbf{w}^T \mathbf{x}_i + b) - 1]$ equals zero for all $i = 1, 2, \cdots, N$ in the optimum. Thus, the support vectors lie on the margin boundary.

However, equation (3) is not feasible if the classes cannot linearly separate. For the non-separable case, slack variables ξ_i's and kernel funtions are introduced. The SVM model for the non-separable case is defined as follows.

$$\begin{aligned} &\min \quad \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N}\xi_i \\ &\text{subject to} \\ &\begin{cases} y_i\left(\mathbf{w}^T\phi(\mathbf{x}_i) + b\right) \geq 1 - \xi_i, & \left(i = 1, 2, \cdots, N\right) \\ \xi_i \geq 0, & \left(i = 1, 2, \cdots, N\right) \end{cases} \end{aligned} \tag{4}$$

where ξ_i's are the slack variables needed to allow misclassifications in the set of inequalities, and the scalar $C \in \mathbb{R}^+$ is a regularization parameter determining the trade-off between the minimization of the fitting errors and the minimization of the model complexity.

The primal model of equation (4) can be converted into the following dual model:

$$\begin{aligned} &\max \quad -\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} + \mathbf{1}^T\boldsymbol{\alpha} \\ &\text{subject to} \\ &\begin{cases} \mathbf{0} \leq \boldsymbol{\alpha} \leq C\mathbf{1} \\ \mathbf{y}^T\boldsymbol{\alpha} = 0 \end{cases} \end{aligned} \tag{5}$$

where $\boldsymbol{\alpha}$ is the vector of Lagrange multipliers α_i, $\mathbf{Q}$ is a $N \times N$ positive semi-definite matrix, $Q_{ij} \equiv y_i y_j K\left(\mathbf{x}_i, \mathbf{x}_j\right)$, and $K\left(\mathbf{x}_i, \mathbf{x}_j\right) = \phi\left(\mathbf{x}_i\right)^T \phi\left(\mathbf{x}_j\right)$ is the kernel function.

For the general SVM model, the complementary slackness condition has the form

$$\alpha_i \left[y_i \left(\mathbf{w}^T \phi(\mathbf{x}_i) + b \right) - 1 + \xi_i \right] = 0 \tag{6}$$

for all $i = 1, 2, \cdots, N$ in the optimum.

There are several alternatives for the kernel function, which can be found in the previous SVM literature [2, 11]. The following kernel functions are suggested:

- $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^T \mathbf{x}_j$: linear kernel
- $K(\mathbf{x}_i, \mathbf{x}_j) = \left(\gamma \mathbf{x}_i^T \mathbf{x}_j + r \right)^d$: polynomial kernel
- $K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left\{ -\gamma \left\| \mathbf{x}_i - \mathbf{x}_j \right\|^2 \right\}$: Gaussian RBF kernel
- $K(\mathbf{x}_i, \mathbf{x}_j) = \tanh\left\{ \gamma \mathbf{x}_i^T \mathbf{x}_j + r \right\}$: sigmoid kernel

where $d, r \in \mathbb{N}$ and $\gamma \in \mathbb{R}^+$.

Among the alternatives, the nonlinear kernel functions can be effectively used for nonlinear separation of the train data.

Then, the final SVM classifier is constructed as

$$H(\mathbf{x}) = sign\left(\sum_i^N \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}_j) + b \right) \tag{7}$$

where the bias b can be obtained by averaging

$$b = y_i - \sum_{i=1}^N y_i \alpha_i K_{ij}, \qquad \forall j \tag{8}$$

by the condition of equation (6).

2.2 Existing Tuning Methods for the Architecture of SVM Models

Regarding the architecture of the SVM models, several methods have been re-

ported in the literature. The critical factors to the performance of SVM models are the input variable selection and the parameter value optimization.

First, for the input variable selection of SVM models, particularly in the area of bankruptcy prediction problems, there have been several approaches in the literature such as statistical approach [20], literature review based approach [25], hybrid approach of combining several methods [14], and expert judgment based approach [19]. Among these approaches, however, the statistical approach representing independent sample *t*-test, stepwise LR (logistic regression), or stepwise MDA (multivariate discriminant analysis) cannot guarantee the performance of a nonlinear SVM model to be maximized. Likewise, the literature-review-based and the expert-judgment-based methods are known that they are inappropriate for building an SVM model of better performance. Therefore, a nonlinear approach would be needed as an alternative for selecting the input variables in building the nonlinear SVM models.

Second, the literature emphasizes that the values of parameters in SVM have to be carefully chosen in advance for better performance [4, 14, 21]. These parameters include parameter γ, the bandwidth of the Gaussian RBF kernel, and parameter C, the regularizing parameter. Min and Lee [14] proposed the grid search method using 5-fold cross validation to find out the optimal parameter values of the Gaussian RBF kernel. And several studies such as Howley and Madden [9], Pai and Hong [15] and Wu *et al*. [25] suggested the GA for optimizing the parameter values. The two approaches, the grid search and the GA, have their own methodological pros and cons respectively. The grid search method can search for the best one among given sets of discrete values in a short time; however, the values found by the method are more likely to be suboptimal. In contrast, the GA has the advantage of searching a wide space to find out the optimal values; however, it may consume too much time to optimize the parameter values.

3. The Hybrid Tuning Method

We suggest a hybrid tuning method for the architecture of SVM as follows. The main procedure contains three steps, and the method sequentially employs the GAM, the grid search, and the GA, each of which is explained in the following subsections.

Step 1: The GAM approach is used for selecting significant input variables. The GAM is able to capture both linear and nonlinear relationships between variables.

Step 2: The grid search method determines the initial values of searching or estimating target parameter values. Ultimate parameter values are sequentially estimated via the GA using the initial values from the grid search

Step 3: GA searches out the optimal parameter values by using the initial values estimated in Step 2. This sequential method is called a modified GA approach.

3.1 Generalized Additive Models

The generalized additive model (GAM) was proposed by Hastie and Tibshirani [7]. The model assumes that the dependent variables are represented as nonlinear additive functions of the independent variables. The GAM is an extension of the GLM. Once the linear additive term $\sum_{j=1}^{J} \beta_j x_{ij}$ in the GLM is replaced with a more general additive term, $\sum_{j=1}^{J} f_j(x_{ij})$, the GAM is established. Here x_{ij} is the value of the j th variable belonging to the i th observation, and $f_i(x_{ij})$ is an arbitrary unspecified function of the j th variable allowing nonlinearity. The GLM forces linearity on the data, whereas the GAM allows for the nonlinearity on the data. The GAM helps discover the underlying detailed data patterns. The logistic generalized additive model with binary response data takes the form

$$\ln \frac{\pi_i}{1-\pi_i} = \alpha + \sum_{j=1}^{J} f_j(x_{ij}) \tag{9}$$

where π_i is the success probability, α is a constant, and the term on the far left is a logit link function.

In the GAM in the additive term selection, two aspects are considered: which variables should be included in the model, and how smooth a variable should be if it is left in the model. In other words, we need to decide which x_j should remain in the model and what the smoothing amount of the term $f_j(\cdot)$ should be. Those decisions can be made using the estimated degrees of freedom (df) of each variable.

According to Hastie and Tibshirani [8], the variables with $df = 0$ should be deleted. For the linear relationship of a variable, the corresponding df equals one. On the other hand, nonlinear terms have a corresponding function where $df > 1$ [18]. A larger df causes a rougher fit. After the model fit is finished, the significance of each additive term is tested with the χ^2 test or F test. Based on these results, each additive term is either eliminated or allowed to remain in the model.

We employed a modified backward selection approach, which is similar to the backward selection method of ordinary linear models. The procedure uses the following steps:

Step 1: Place all of the input variables in a generalized additive model.

Step 2: Check the df of the additive terms to determine whether the df is near 0 or not. As previously mentioned, if the df on an additive term is near 0, the corresponding variable is deleted from the set of input variables.

Step 3: Place all of the remaining input variables in a generalized additive model.

Step 4: Check the df of the additive terms to determine whether the df is near 0 or not. If there are any additive terms whose df are estimated near 0, remove the corresponding variables and return to Step 3. Otherwise, check the significance of the terms by p-values from χ^2 tests or F tests, and eliminate a single variable of the term with the highest non-significant p-value from the model.

Step 5: Place all of the remaining input variables in an additive model.

Step 6: Repeat Steps 4 and 5 until all the remaining terms are significant.

Through backward selection, we can find an additive model in which all the terms of the remaining input variables are significant under the significance level of 0.05. The variables in this last additive model are applied to the SVM model prediction. To implement the GAM, we utilized the *"mgcv"* package in *R* software, which provides the significance of the additive terms in its summary of the model fit.

3.2 Grid Search Method

A grid search method is applied to suggest the appropriate initial values for searching target parameter values. In this article, the grid search is implemented as a

preliminary method before the GA is employed. The grid search procedure helps the GA detect the global optimal set of target parameter values in a moderate searching time.

If either the grid search or the GA is exclusively used, the optimization faces some limitations. The grid search method and the GA both offer methodological pros and cons. One advantage is that the grid search selects a better set of values among given sets of discrete values in a relatively short time; however, the set of values estimated by the grid search is more likely to be the suboptimal set. On the contrary, the GA performs a global search to find optimal values from numerous options. The GA's weak point is its relatively long searching time for the global optimum. If the search range is too wide, it takes much more time to discover the optimal set of values. Therefore, by employing the two methods in sequence, we expect they complement each other.

Candidates of initial values for target parameters are usually evaluated via the ν-fold cross-validation approach if a classification problem is involved. In this approach, the overall training data set is divided into ν subsets of equal size. Each of the ν subsets is tested using the classifiers that are trained from the remaining $(\nu-1)$ subsets. The overall misclassification rate, which is the mean of the misclassification rates of all the ν subsets, is computed for performance evaluation. In this study, the grid search determines the two parameters γ and C using 5-fold cross validation. Several candidates for the pairs of γ and C are evaluated through cross validation. Among the given pairs of the two parameters, the q pairs with the lowest overall misclassification rates are selected as the set of initial values for the following GA search.

3.3 Genetic Algorithm

A genetic algorithm (GA) is an algorithm representing the evolving mechanism in nature. It stochastically searches for wide and complex spaces to detect the optimal solution. In general, the procedure of a standard GA can be described using the following itemized steps:

Step 1: Generate a set of n solutions (called chromosomes in GA terms) in an initial

population.

Step 2: Evaluate those solutions in the population, and sort them in order of how much they contribute to fitness function.

Step 3: Select the best p solutions among the population and remove the remaining solutions.

Step 4: Generate new $(n-p)$ solutions using genetic operators such as selection and mutation. The newly generated solutions replace the removed solutions of the population.

Step 5: Repeat Steps 2 to 4 until a predetermined stopping condition is satisfied.

Step 6: Choose the best contributed solution to the fitness function among the population as the optimal solution.

In this article, we suggest a modified version of the standard GA. The population in the ordinary GA has only n randomly selected solutions, whereas the modified GA has an initial population that consists of two parts: the randomly selected $(n-q)$ solutions; and the remaining q solutions, where q solutions are obtained in advance through the grid search procedure.

The evaluation criterion of the GA in this study is designed as

$$\min \frac{\sum_{l=1}^{L} E_l(\gamma, C)}{L} \tag{10}$$

where γ and C are the parameter values of an SVM model. L indicates the number of sub data sets, γ indicates the bandwidth of the Gaussian RBF kernel, and C indicates the parameter for regularizing the model's complexity. $E_l(\gamma, C)$ is the misclassification rate of the model built with parameter γ and C for sub data set l.

Each of the sub data sets has the same number of data; these data are randomly selected from the training data set without replacement. The remaining data are used for training an SVM model, and the data in the sub data set are used for validating the model's performance. The researchers select the number of sub data sets, which is set to be 5 in our empirical analysis. As the number of sub data sets increases, the optimized parameter values of SVM models are expected to gain more generality.

4. Empirical Analysis

4.1 Data and Variable Selection

In this subsection, our data set is described, and the variable selection method via the GAM approach is explained in detail. Furthermore, this method is validated by comparing its performance with those of the corresponding SVM models with several different sets of input variables via other methods.

The final data set for the empirical analysis consists of the financial ratios of 2,542 externally audited small- and medium-sized manufacturing firms in Korea. Among them, 1,271 firms filed for bankruptcy and the other 1,271 firms did not file for bankruptcy during the period from 2001 to 2004. The bankruptcy in this paper is defined as a legally declared inability of a firm to pay its creditors. Following the several literatures on bankruptcy prediction [14, 16, 17, 27], a balanced sample of bankrupt and non-bankrupt firms is used.

The data selection process began as we gathered 27 financial ratios of 2,814 bankrupt and non-bankrupt firms to conduct an empirical analysis. For bankrupt firms, we gathered the financial ratios as of one year prior to bankruptcy. Second, we used the means and standard deviations of the financial ratios of 2,814 companies to standardize them as Z-values. Third, the observations with Z-values that were beyond the range of [-3, 3] were considered outliers and were deleted from the data set. Fourth, in order to equalize the numbers of bankrupt and non-bankrupt firms, we excluded 45 randomly selected non-bankrupt firms from the data set. The definitions of the initially considered 27 financial ratios are illustrated in Table 1. In the table's right column, the corresponding financial categories are notated: "a" corresponds to productivity, "b" to profitability, "c" to stability, "d" to activity, and "e" to liquidity. These categories have been popularly utilized in the accounting area to encapsulate the meanings of various financial ratio variables.

For the SVM model application, only some significant variables out of the original 27 variables are selected. To detect the significant variables, the proposed backward selection method via the GAM was applied. The procedure sequentially dropped a single term with the largest insignificance from the F test for the GAM fitting and re-fitting until all the remaining terms were significant. Specifically, in the first conduct of the GAM, the values of df of the additive terms X1, X4, X5, X6, X8, X12, X13, X14, X19, X22, X24, and X26 were very close to zero. Hence, these variables were

eliminated from the candidates of input variables. In the second trial, the df value of the additive term X27 was so close to zero that this variable was excluded from the candidates. In the third attempt, the df values of all the remaining additive terms in the model were not near to zero. But the term X17 indicated the largest p-value for the F test among the remaining terms, so X17 was removed from the candidates. In this way, the backward selection steps via the GAM were sequentially conducted until all the additive terms of the GAM model were significant at the level of 0.05.

Table 1. Input Variables for the Analysis

Variable	Definition	Category
X1	Gross Value Added to Sales	a
X2	Gross Value Added to Total Assets	a
X3	Growth Rate of Total Assets	a
X4	Ordinary Income to Sales	b
X5	Net Income to Sales	b
X6	Operating Income to Sales	b
X7	Costs of Sales to Sales	b
X8	Net Interest Expenses to Sales	b
X9	Ordinary Income to Total Assets	b
X10	Rate of Earnings on Total Capital	b
X11	Net Working Capital to Total Assets	c
X12	Current Liabilities to Total Assets	c
X13	Stockholders' Equity to Total Assets	c
X14	Total Borrowings and Bonds Payable to Total Assets	c
X15	Total Assets Turnover	d
X16	Ordinary Income to Total Assets	d
X17	Net Working Capital to Sales	d
X18	Stockholders' Equity to Sales	d
X19	Ordinary Income to Total Assets	d
X20	Depreciation Expense	d
X21	Operating Assets Turnover	d
X22	Interest Expenses to Total Expenses	e
X23	Net Interest Expenses	e
X24	Break-Even Point Ratio	b
X25	Employment Costs	e
X26	Net Income to Total Assets	b
X27	Earnings Before Interest and Tax to Sales	b

Table 2 shows the output of the last step of backward selection. According to Table 2, nine variables were finally selected through the backward procedure: X2, X3, X9, X10, X11, X20, X21, X23, and X25. These nine variables moderately reflect five popular financial accounting categories in Table 1. In summary, X2 and X3 are related to pro-

ductivity, X9 and X10 are related to profitability, X11 is related to stability, X20 and X21 are related to activity, and X23 and X25 are related to liquidity.

Table 2. Significance of the Additive Terms at the Last Step

Additive Term	*df*	*F*	*p*-value
s(X2)	4.261	40.152	< 2.00E-16
s(X3)	5.078	9.972	1.42E-09
s(X9)	2.333	3.075	0.03854
s(X10)	4.542	3.394	0.00625
s(X11)	1.987	5.166	0.00588
s(X20)	6.488	5.452	6.48E-06
s(X21)	1.985	6.409	0.00172
s(X23)	4.635	11.54	2.12E-10
s(X25)	7.092	54.839	< 2.00E-16

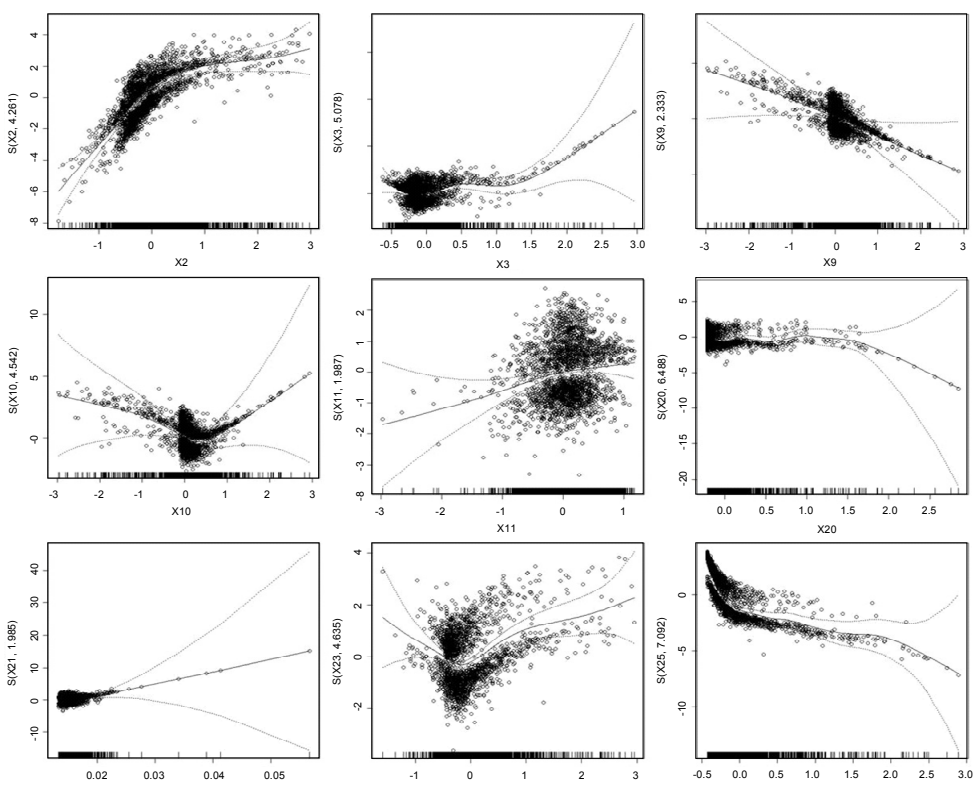

Figure 1. Plots of Partial Residuals of the Additive Terms

Figure 1 shows the degree of linearity or nonlinearity of the relationships between the input variables and the corresponding additive terms. As the degree of freedom of the additive function is close to one, the relationship between the corresponding input variable and the value of the additive function comes near to linearity. As the degree of freedom of the additive function deviates more from one, on the contrary, the relationship is more prone to nonlinearity.

Seven additional sets of input variables via other variable selection methods were obtained so that they could be compared with the performance of the model whose input variables are selected by the GAM. These variables are summarized in Table 3. The input variables in each group were selected by employing the following methods: stepwise logistic regression (LR), stepwise MDA, and the GAM approach. Groups 1, 2, and 3 are composed of input variables by linear-model-based methods. Group 4 has the input variables selected by the GAM. Groups 4 to 7 have the input variables by both linear-model-based methods and GAM. And finally, group 8 includes all the input variables in the data.

A total of eight groups of selected variables were evaluated by comparing the performances of the corresponding SVM models.

Table 3. Groups of the Input Variables by the Selection Methods

Group	Description	Variables
1	Selected by stepwise LR	X2, X3, X7, X9, X12, X20, X21, X22, X25
2	Selected by stepwise MDA	X2, X3, X9, X12, X15, X20, X21, X22, X23, X25
3	Selected by Stepwise LR or MDA	X2, X3, X7, X9, X12, X15, X20, X21, X22, X23, X25
4	Significant variables from GAM	X2, X3, X9, X10, X11, X20, X21, X23, X25
5	Variables in Group 1, 4	X2, X3, X7, X9, X10, X11, X12, X20, X21, X22, X23, X25
6	Variables in Group 2, 4	X2, X3, X9, X10, X11, X12, X15, X20, X21, X22, X23, X25
7	Variables in Group 1, 2, 4	X2, X3, X7, X9, X10, X11, X12, X15, X20, X21, X22, X23, X25
8	All the variables from the data	X1-X27

4.2 Tuning the SVM Models by the Sequential Method

The next step is to estimate the approximate values of two parameters of the SVM model, γ and C, by the grid search. The grid search was conducted on expo-

nentially growing sequences $[2^{-5},\ 2^{-4},\ 2^{-3},\ 2^{-2},\ 2^{-1},\ 2^{0}]$ for γ, and $[2^{0},\ 2^{1},\ 2^{2},\ 2^{3},\ 2^{4},\ 2^{5}]$ for C using 5-fold cross validation.

Figure 2 shows the contours of misclassification rates of 8 SVM models from the grid search. In Figure 2, SVM(#) denotes the SVM model with the input variables of Group # in Table 3. The darkness of the color indicates the level of misclassification rate. The misclassification rates generated by the corresponding grids of the two parameters, γ and C, differ in the ranges from about 0.20 to 0.34.

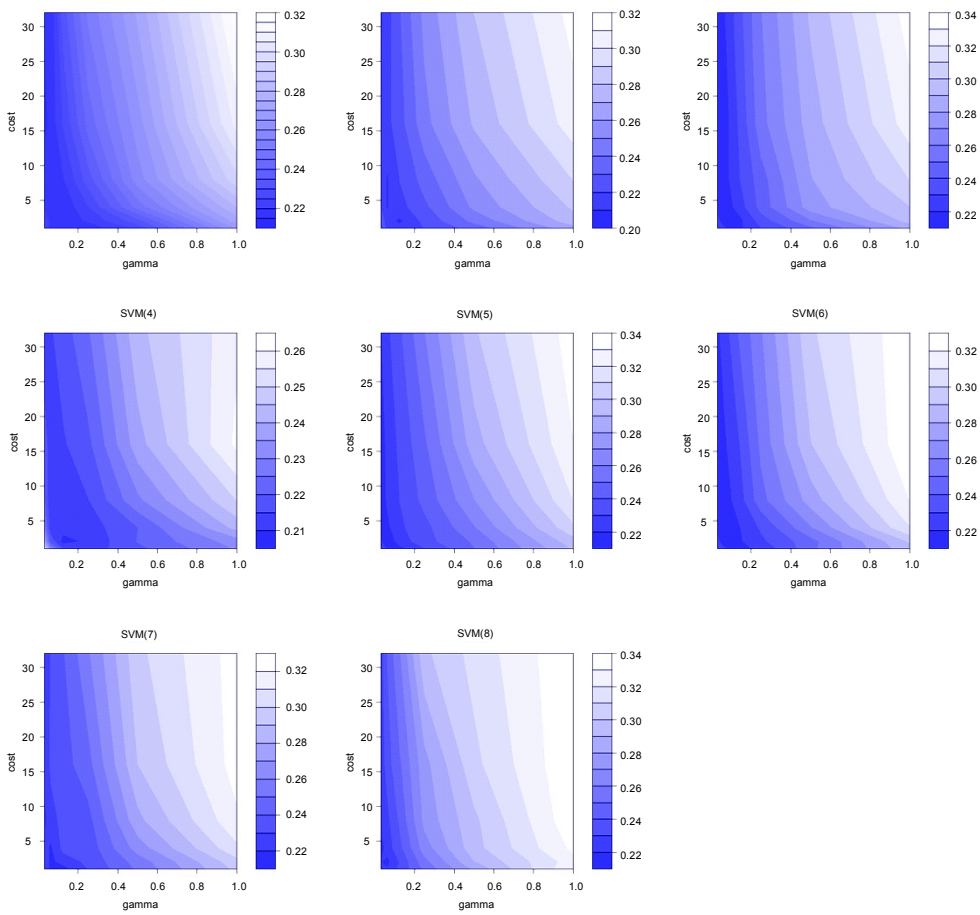

Figure 2. Grid search on 2^{-5} to 2^{-0} for γ and 2^{0} to 2^{5} for C

Table 4 shows the best among the given sets of γ and C for each SVM model, which is estimated by the grid search. They generate the lowest misclassification rate

for each SVM model. We can see that there are differences in the estimated best values of γ and C by the SVMs built with different input variables.

Table 4. Estimated Values of the Parameters

Model	γ	C
SVM(1)	2^{-5}	2^{4}
SVM(2)	2^{-3}	2^{1}
SVM(3)	2^{-5}	2^{4}
SVM(4)	2^{-3}	2^{1}
SVM(5)	2^{-4}	2^{1}
SVM(6)	2^{-4}	2^{2}
SVM(7)	2^{-3}	2^{0}
SVM(8)	2^{-5}	2^{2}

To compare the performance of the SVM models built by the different sets of parameters in Table 4, a holdout validation was conducted. We first split the data into train and test data sets in a proportion of 8 : 2. The train data set was used to build the SVM models, and the test data set was used to evaluate the performance of the SVM models. This process was repeated 100 times. Figure 3 and Table 5 depicts the results of the holdout validation.

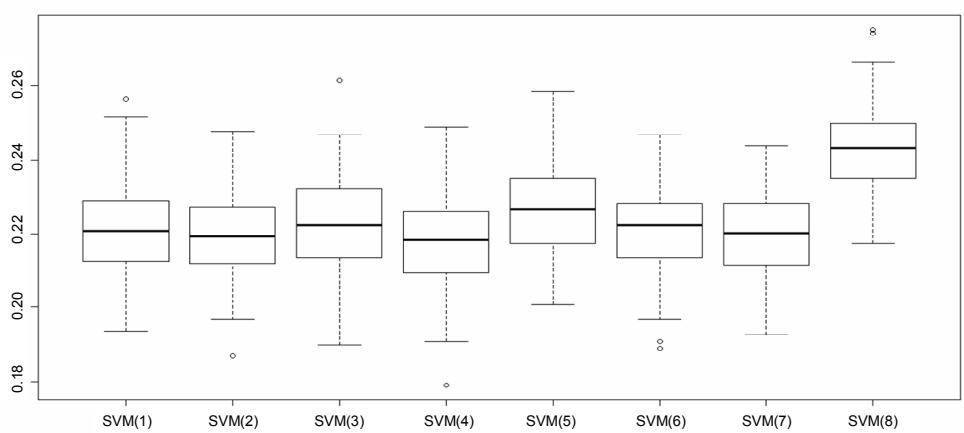

Figure 3. Box plots of Misclassification Rates of the SVM Models

The box plots in Figure 3 show the respective distributions of the misclassification rates of 8 SVM models. The box plots of SVM(5) and SVM(8) show relatively

high level of misclassification rates in comparison with the others, which implies that SVM(5) and SVM(8) are inferior to the other SVM models in terms of classification power. There seems to be little difference in the misclassification rates among the other models.

Table 5. Summary Statistic of Misclassification Rates of 8 SVM Models

Model	N	Minimum	Maximum	Mean	Std. Deviation
SVM(1)	100	0.1937	0.2566	0.2212	0.0124
SVM(2)	100	0.1868	0.2478	0.2199	0.0124
SVM(3)	100	0.1898	0.2616	0.2232	0.0125
SVM(4)	100	0.1790	0.2488	0.2181	0.0122
SVM(5)	100	0.2006	0.2586	0.2273	0.0131
SVM(6)	100	0.1888	0.2468	0.2210	0.0118
SVM(7)	100	0.1927	0.2439	0.2200	0.0119
SVM(8)	100	0.2173	0.2753	0.2431	0.0118

Table 5 shows that SVM(4) has the lowest mean of misclassification rates; however, before we confirm that the model outperforms the other ones, we need an additional analysis to check if there exist statistically significant differences among the misclassification rates of the 8 SVM models. Table 6 shows the results of the paired t-tests between the misclassification rates of SVM(4) and each of the other ones.

Table 6. Results of Paired t−Tests

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
SVM(1)-SVM(4)	0.0031	0.0095	0.0009	0.0012	0.0050	3.2812	99	0.0014
SVM(2)-SVM(4)	0.0018	0.0104	0.0010	-0.0003	0.0038	1.6976	99	0.0927
SVM(3)-SVM(4)	0.0051	0.0115	0.0012	0.0028	0.0074	4.4486	99	0.0000
SVM(5)-SVM(4)	0.0092	0.0117	0.0012	0.0069	0.0115	7.8865	99	0.0000
SVM(6)-SVM(4)	0.0029	0.0095	0.0010	0.0010	0.0048	3.0360	99	0.0031
SVM(7)-SVM(4)	0.0019	0.0102	0.0010	-0.0002	0.0039	1.8294	99	0.0704
SVM(8)-SVM(4)	0.0250	0.0118	0.0012	0.0227	0.0274	21.2323	99	0.0000

From the results in Table 6, SVM(4), SVM model with the input variables selected by the GAM approach, significantly outperforms most of the alternatives under the significance level of 0.05 while it weakly performs better than SVM(2) and SVM(7).

This implies that the GAM is a promising approach for the input variable selection that can replace the other linear methods.

The next step of the proposed method is to make the GA search out the spaces of two parameters γ and C to find the optimal parameter values. The values of the parameters in Table 4 are just the best ones among the discrete values given by the authors, not the global optimal values. Hence, it is necessary to search a wide space of real numbers in order to get the optimal values. The GA is employed for the purpose of searching the global optimum.

The parameter values estimated by the grid search can be used for a part of the initial population chromosomes in the GA step. By doing this, we can save the searching time by the GA, and increase the probability of finding out the optimal solutions. In this study, 5 pairs of parameter values showing the best performance among the 16 alternative pairs of parameters in Table 7 are used for the initial chromosome values in the population. The performance level is measured using the mean of the misclassification rates.

Table 7. Performance of the Set of Parameter Values of SVM by the Grid Search

Set	γ	C	Mean of Misclassification Rate
1	0.1250	2	0.2070
2	0.2500	4	0.2085
3	0.1250	4	0.2113
4	0.0625	16	0.2117
5	0.2500	8	0.2121
6	0.0625	8	0.2121
7	0.0625	4	0.2121
8	0.2500	2	0.2129
9	0.0313	16	0.2133
10	0.1250	8	0.2148
11	0.1250	16	0.2164
12	0.0625	2	0.2180
13	0.0313	8	0.2188
14	0.2500	16	0.2243
15	0.0313	4	0.2251
16	0.0313	2	0.2381

With the modified GA in use, the chromosome structure of the model is de-

scribed as follows. Each chromosome is composed of two kinds of genes representing γ and C. The genes for parameters γ and C are set to be positive real numbers in the ranges of [0, 1] and [1, 16] respectively. To search for the optimal values, we set the population size to be 50, among which 5 chromosomes are given from the grid search and 45 chromosomes are randomly generated. While running the GA algorithm, the rates of crossover and mutation are set to be 0.2 for both γ and C. The search algorithm is designed to repeat 100 times.

The process of optimizing two parameters C and γ over 100 generations by the GA is shown in Figure 4. We can see the initial values of the parameters are converging to a certain set of values over 100 generations.

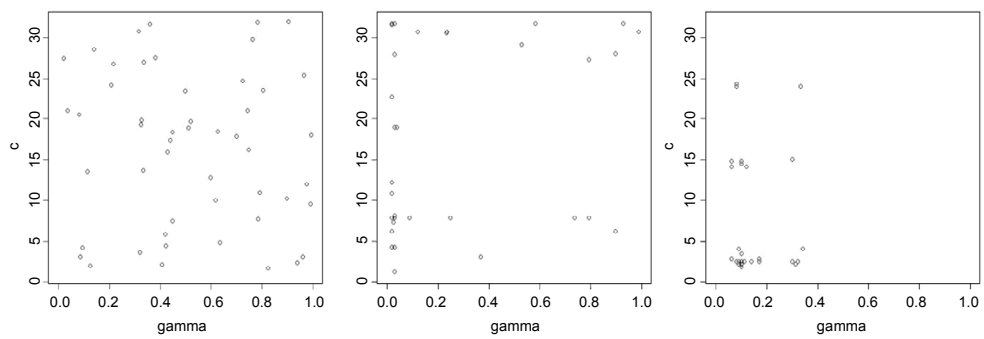

Figure 4. Process of Optimizing by the GA

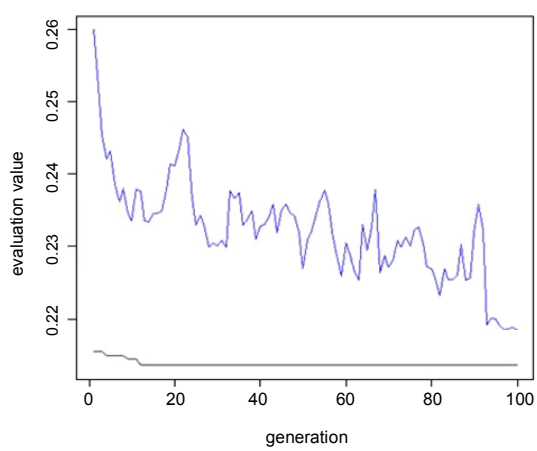

Figure 5. Best and Mean Evaluation Values over 100 Generations in the case of SVM (GA1)

Figure 5 illustrates the trend of the best and the mean of evaluation values, the mean of misclassification rates, in the population over 100 generations in case of SVM (GA1). SVM (GA1) is the SVM model with the grid search. The upper line indicates the mean of evaluation values in the population, and the lower line indicates the best of evaluation values in the population. We note that the best evaluation value remains constant from about the 12th generation through 100th generation, which implies that the convergence has been made in an early stage.

On the other hand, Figure 6 shows the case of SVM (GA2). SVM (GA2) is the SVM model without the grid search. Comparing it with Figure 5, we can tell that the convergence of the best evaluation values is made in a very late stage, around at the 120th generation. This result indicates that conducting the GA step after the grid search is more efficient to find out the optimal solutions than conducting the GA step without the preliminary grid search.

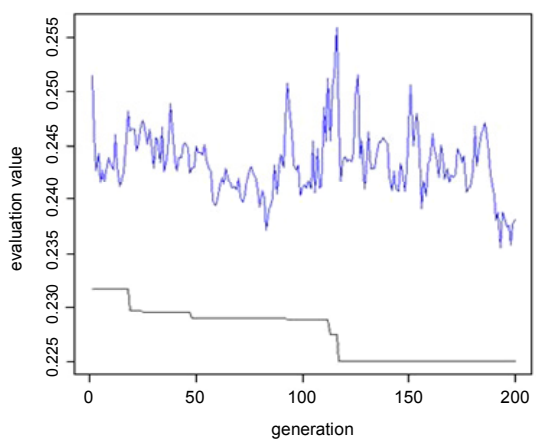

Figure 6. Best and Mean Evaluation Values over 200 Generations in the case of SVM (GA2)

Table 8 shows the optimal estimates of parameters γ and C for the three SVM models. Here, γ and C of SVM (grid) are estimated by the grid search (same as the values for SVM(4) in Table 4), those of SVM (GA1) are optimized by using both the grid search and the GA (the suggested modified GA), and those of SVM (GA2) are optimized only by the GA without the preliminary grid search (the standard GA). In order to optimize the parameter values of SVM (GA2), we set the searching space to be real numbers between 2^{-5} and 2^{-0} for γ, and between 2^{0} and 2^{5} for C. In addi-

tion, we set the number of generations to be 200, which are twice as many generations as for the SVM (GA1).

Table 8. Estimated Values of the Parameters

SVM Model	γ	C
SVM (grid)	2^{-3}	2^{1}
SVM (GA1)	0.1008534	2.52947
SVM (GA2)	0.0411414	22.11387

After optimizing the values of the two parameters of SVM, we performed a holdout validation once again to compare the performances of the SVM models built by three different sets of values in Table 8. A proportion of 8 to 2 was applied to the ratio of the train data set and the test data set. Table 9 shows the results of the holdout validation.

Table 9. Summary Statistic of Misclassification Rates of SVM Models

SVM Model	N	Minimum	Maximum	Mean	Std. Deviation
SVM (grid)	100	0.1849	0.2458	0.2168	0.0122
SVM (GA1)	100	0.1878	0.2439	0.2158	0.0121
SVM (GA2)	100	0.1898	0.2458	0.2172	0.0106

According to Table 9, it is clear that SVM (GA1) is superior to both SVM (grid) and SVM (GA2) in the sense of a smaller mean of misclassification rates, which implies that the SVM model using both the grid search and the GA shows more accurate prediction power on the average than the SVM model using only one method of the two. Note that SVM (GA1) still outperforms SVM (GA2) even though the algorithm iterations in SVM (GA2) were increased; this increase simply caused an increase in the searching time. This result confirms that SVM (GA1) performs better than SVM (GA2) in a moderately finite searching time. To statistically assure this claim, however, it may be necessary to conduct additional paired t-tests between the mean misclassification rates of SVM (GA1) and each of the other two SVM models. Table 10 shows the results of the paired t-tests.

Table 10 clearly shows that there exist statistically significant differences between the mean of misclassification rates of SVM (GA1) and those of the other two models,

SVM (grid) and SVM (GA2), under the significance level of 0.05. The sequential searching method via the combination of the grid search and the GA significantly outperforms the exclusive method either by the grid search or the conventional GA in this empirical analysis.

Table 10. Results of Paired t-Tests between SVM Models

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Dev.	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
SVM (grid)-SVM (GA1)	0.0010	0.0027	0.0003	0.0004	0.0015	3.6340	99	0.0004
SVM (GA2)-SVM (GA1)	0.0014	0.0058	0.0006	0.0002	0.0025	2.3561	99	0.0204

5. Conclusions

This paper suggests a hybrid tuning method for the architecture of SVM models, which employs the GAM, the grid search, and the GA in sequence. The GAM, a nonlinear model, was applied to the input variable selection for building nonlinear SVM models. It is shown that the GAM is superior to other input variable selection methods in terms of misclassification rates. The grid search is used for estimating the approximate values of two parameters for the SVM model using RBF kernel function. It reduces the searching space so that the GA can find optimal parameter values in a reasonable amount of time. The GA then investigated the space to detect the optimal parameter values of the SVM models.

The suggested method has some advantages in building SVM models. First, by using the GAM, the method can find out which variables have significant linear or nonlinear pattern for classifying data. Second, by using both the grid search and the GA, the method can perform a global search to obtain the optimal parameter values in a relatively short time.

Applying the suggested tuning method to a bankruptcy prediction problem, we empirically showed that the SVM model tuned by the suggested method significantly outperformed the other SVM models. The method can serve as a fine-tuner for the

SVM model to enhance its predicting power.

In spite of the contributions from the newly suggested method, there are a couple of notable limitations we would like to mention. First, our data set for the empirical analysis is somewhat limited. The data set includes risky small- and medium-sized manufacturing firms in a specific country during a specific period. To generalize the merit of the suggested tuning method, this technique needs to be applied to other versatile data sets in various areas. Second, the variable selection results via the GAM can vary. Although the GAM captures the nonlinear relationships among variables, the literature reports that the variable selection results through this method are somewhat unstable. The significance of the selected variables via the GAM can be changed by including new data into the existing data set or by removing some data from the existing set. A robust method leading to consistently stable variable selection by the GAM needs to be developed in the future.

References

[1] Boser, B. E., I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," *Proceedings of the 5th annual ACM workshop on computational learning theory*, New York, NY: ACM Press, 1992.

[2] Campbell, C., "Kernel methods: a survey of current techniques," *Neurocomputing* 48 (2002), 63-84.

[3] Cao, L. J., "Support vector machines experts for time series forecasting," *Neurocomputing* 51 (2003), 321-339.

[4] Duan, K. and S. S. Keerthi, "Evaluation of simple performance measures for tuning SVM hyperparameters," *Neurocomputing* 51 (2003), 41-59.

[5] Fan, A. and M. Palaniswami, "A new approach to corporate loan default prediction from financial statements," *Proceedings of Computational Finance/ Forecasting Financial Markets Conference (CF/FFM-2000)*, London, 2000.

[6] Gestel, T. V., J. A. K. Suykens, D. E. Baestaens, A. Lambrechts, G. Lanckriet, B. Vandaele, B. De Moor, and J. Vandewalle, "Financial time series prediction using least squares support vector machines within the evidence framework," *IEEE Transactions on Neural Networks* 12 (2001), 809-821.

[7] Hastie, T. and R. Tibshirani, "Generalized additive models," *Statistical Science* 1 (1986), 297-318.

[8] Hastie, T. and R. Tibshirani, *Generalized additive models*. London: Chapman and Hall, 1990.

[9] Howley, T. and M. G. Madden, "The genetic kernel support vector machine: Description and evaluation," *Artificial Intelligence Review* 24 (2005), 379-395.

[10] Huang, Z., H. Chen, C. Hsu, W. Chen, and S. Wu, "Credit rating analysis with support vector machines and neural networks: A market comparative study," *Decision Support Systems* 37 (2004), 543-55.

[11] Kecman, V., *Learning and soft computing,* Cambridge, MA: The MIT Press, 2001.

[12] Kim, H. S. and S. Y. Sohn, "Support vector machines for default prediction of SMEs based on technology credit," *European Journal of Operational Research* 201 (2010), 838-846.

[13] Kim, K. J., "Financial time series forecasting using support vector machines," *Neurocomputing* 55 (2003), 307-320.

[14] Min, J. H. and Y. C. Lee, "Bankruptcy prediction using support vector machine with optimal choice of kernel function parameters," *Expert Systems with Applications* 28 (2005), 603-614.

[15] Pai, P.-F. and W.-C. Hong, "Support vector machines with simulated annealing algorithms in electricity load forecasting," *Energy Conversion and Management* 46 (2005), 2669-2688.

[16] Pendharkar, P., "A threshold-varying artificial neural network approach for classification and its application to bankruptcy prediction problem," *Computers & Operations Research 32* (2005), 2561-2582.

[17] Piramuthu, S., H. Ragavan, and M. J. Shaw, "Using feature construction to improve performance of neural networks," *Management Science* 44 (1998), 416-430.

[18] Ruppert, V., M. P. Wand, and R. J. Carroll, *Semiparametric regression*, New York: Cambridge Press, 2003.

[19] Sancho, S., D. Mario, S. M. Jesús, P. Fernando, and B. Carlos, "Feature selection methods involving support vector machines for prediction of insolvency in non-life insurance companies," *International Journal of Intelligent Systems in Accounting, Finance and Management* 12 (2004), 261-281.

[20] Shin, K. S., T. S. Lee, and H. J. Kim, "An application of support vector machines in bankruptcy prediction model," *Expert Systems with Applications* 28 (2005),

127-135.

[21] Tay, F. E. H. and L. Cao, "Application of support vector machines in financial time series forecasting," *OMEGA: The International Journal of Management Science* 29 (2001), 309-317.

[22] Vapnik, V. N., *The Nature of Statistical Learning Theory,* Verlag: Springer, 1995.

[23] Vapnik, V. N., *Statistical Learning Theory,* New York: Wiley, 1998.

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

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

[26] Yoon, J. S. and Y. S. Kwon, "A practical approach to bankruptcy prediction for small businesses: Substituting the unavailable financial data for credit card sales information," *Expert Systems with Applications* 37 (2010), 3624-3629.

[27] Zhang, G., M. Hu, B. Patuwo, and D. Indro, "Artificial neural networks in bankruptcy prediction: General framework and cross validation analysis," *European Journal of Operational Research* 116 (1999), 16-32.