Bond rating using support vector machine

Lijuan Cao[a,1], Lim Kian Guan[b] and Zhang Jingqing[a]
[a]*Financial Studies of Fudan University, HanDan Road, ShangHai, P.R. China, 200433*
E-mail: {ljcao,zhangjq}@fudan.edu.cn
[b]*Department of Business, Singapore Management University, 469 Bukit Timah Road, Singapore 259756*
E-mail: kglim@smu.edu.sg

Received 9 August 2005
Revised 8 October 2005
Accepted 2 November 2005

Abstract. This paper deals with the application of support vector machine (SVM) for bond rating. The three commonly used methods for solving multi-class classification problems in SVM, "one-against-all", "one-against-one", and directed acyclic graph SVM (DAGSVM) are used. The performance of SVM is compared with several benchmarks. One real U.S. bond data is collected using the Fixed Investment Securities database (FISD) and the Compustat database. The experiment shows that SVM significantly outperforms the benchmarks. Among the three SVM based methods, there is the best performance in DAGSVM. Furthermore, an analysis of features shows that the generalization performance of SVM can be further improved by performing feature selection.

Keywords: Support vector machine (SVM), multi-class classification, feature selection

1. Introduction

Recently, support vector machine (SVM) has been successfully applied to solve various real problems such as text classification [1], face detection [2] and breast cancer diagnosis [3], due to the good generalization performance of SVM. SVM is originally proposed by Vapnik and his co-workers in 1995 [4]. Unlike most of the traditional methods which implement the Empirical Risk Minimization Principal, SVM implements the Structural Risk Minimization Principal which seeks to minimize an upper bound of the generalization error rather than to minimize the training error, eventually resulting in better generalization performance than the traditional methods. SVM can always converge to the global optimal solution by solving a linearly constrained quadratic programming. This is another advantage in comparison with traditional neural networks which may get stuck into a local minima by solving a non-linear optimization problem. Furthermore, SVM is practical in terms of training time for small size problems with the training data points less than ten thousands.

This paper applies SVM to solve one important financial problem – bond rating. The bond rating problem is to assign each of the issued bonds into one of the pre-defined ratings such as AA, A, BBB,

[1]The research work is funded by National Natural Science Research Fund No. 70501008 and sponsored by Shanghai Pujiang program. E-mail: ljcao@fudan.edu.cn.

etc., which is actually a multi-class classification problem. In this paper, the three commonly used methods for solving multi-class classification problems in SVM, "one-against-all", "one-against-one", and directed acyclic graph SVM (DAGSVM) [5] are used. Their performance is also compared with the backprogation (BP) neural network, logistic regression (LR) and ordered probit regression (OPR) which are mostly used for bond rating. By examining one real US bond data, out experiment shows that SVM performs much better than all of the benchmarks. Among the three SVM based methods, there is the best performance in DAGSVM.

Moreover, sensitivity analysis is also used in SVM for analyzing the importance of features. Sensitivity analysis ranks the importance of features by calculating the partial derivative of the output to the inputs. The details of calculating the sensitivity analysis in SVM is described in our previous paper [6]. By ranking the importance of features based on sensitivity analysis, feature selection is also performed by recursively removing irrelevant features at each step and re-training SVM using the remaining features. The experiment shows that the generalization performance of SVM can be improved by performing feature selection.

This paper is organized as follows. Section 2 presents the background on bond rating. In Section 3, the theory of SVM for classification is briefly introduced. And "one-against-all", "one-against-one" and DAGSVM are described. Section 4 presents the experimental results including data collection, the comparison experiment with benchmarks and the analysis of features. Section 5 concludes the work.

2. Background on bond rating

A bond is a debt security issued by a company or a government which has the duty to periodically pay an amount of interest to the investor and redeem the bond at face value at maturity [7]. If the company is unable to meet one of the payments (called missed pay or delayed pay), it is said to be in default. According to the inherent default risk of the issuing companies, bonds are categorized into different ratings. So the ratings are actually the measures of default risk. Usually, a high value of rating denotes low default risk. On the contrary, a low value of rating denotes high default risk. The rating of bonds is generally done by independent rating agencies such as Standard&Poor, Mood's Investor Service and Fitch. In the case of Standard&Poor, the possible ratings of bonds are AAA, AA, A, BBB, BB, B, CCC, CC, C, and D, where AAA denotes the best rating corresponding to the least default risk and D denotes the worst rating corresponding to a full default.

For the rating agency, the determining of bonds' rating is a complex process, which includes analyzing issuing company's financial documents such as past annual balance sheets, past annual income statements, latest quarterly reports, as well as individual bonds' characteristics. Although the exact method for the rating agency to assign the rating is not publicly disclosed, it is believed that the financial variables of the company play an important role on the bond rating. They could be the leverage ratios, the liquidity ratios and the profitability ratios. In general, a company with high profitability ratios and low liquidity ratios will have much capacity to repay the loan, thus getting high ratings for its issued bonds.

As the rating agency requires a large amount of fee for the service, not every company would like to do so. Also for the companies with assigned ratings, the ratings are reviewed periodically. The given rating cannot reflect the default risk of the company in time. For these reasons, it is very meaningful to develop a mathematical model for bond rating to make it less dependent on the rating agency.

In literature a variety of methods are proposed for bond rating. These methods can generally be grouped into two categories: traditional statistical methods and artificial intelligence methods. The traditional statistical methods include linear regression [8], multiple discriminant analysis [9], LR [10] and OPR [11]. The artificial intelligence methods include BP neural network [12–14], self-organization feature map [15] and radial basis function [16]. Many studies show that the artificial intelligence methods can perform better than the traditional statistical methods in bond rating as the statistical methods requires some assumptions which may not be appropriate for bond rating. For example, in [17], Maher and Sen find that the BP neural network performs significantly better than LR. In [18], Kim finds that the BP neural network is a better tool than linear regression, multiple discriminant analysis and LR. Chaveesuk et al. [19] also find that the BP neural network is a better choice than other neural network models such as radial basis function. In this paper, the BP neural network, LR and OPR are used as the benchmarks.

3. Support vector machine (SVM) classifier

SVM is originally designed for binary classification. When used for solving multi-class classification problems, two categories of approaches are generally used. One is to construct several binary SVM classifiers and then to combine them to solve the multi-class classification problems, such as "one-against-all", "one-against-one" and DAGSVM. The other approach attempts to construct a multi-class classifier by considering all the classes of data at one time. As shown in [20], the later method is more difficult to implement than the binary based approaches. In this section, the theory of SVM for binary classification is firstly introduced. The idea of "one-against-all", "one-against-one" and DAGSVM are then described.

SVM performs classification by mapping the original input vector into a high dimensional feature space and then constructing an optimal hyperplane in the high dimensional feature space. The optimal hyperplane will separate the training data with a maximal margin – the minimum distance from the separating hyperplane to the closest training data points. As maximizing the margin gives the minimal value on the bound of the generalization error, SVM could achieve good generalization performance by using the optimal hyperplane.

Given a set of data points $\{(X_i, y_i)\}_{i=1}^{l}, X_i \in R^n, y_i = \pm 1$ (X_i is the n-dimensional input vector, y_i is the actual class label, l is the total number of training data patterns), SVM approximates the separating hyperplane by

$$f(x) = \operatorname{sign}(W \cdot \phi(X) + b) \tag{1}$$

where $\phi(X)$ is the high dimensional feature space which is nonlinearly mapped from the input space X.

For the linear function Eq. (1), the value of the margin in the optimal hyperplane is equal to $\frac{2}{\|W\|}$. To achieve the maximal margin, SVM estimates the coefficient W and b by minimizing

$$\text{minimize:} C \sum_{i=1}^{l} \xi_i + \frac{1}{2} \|W\|^2 \tag{2}$$

subject to:

$$y_i(W \cdot \phi(X_i) + b) \geqslant 1 - \xi_i,\ i = 1,\ \ldots,\ l$$

In the risk function Eq. (2), the first term $C \sum_{i=1}^{l} \xi_i$ is the sum of the training errors. The second term $\frac{1}{2} \|W\|^2$, on the other hand, attempts to maximize the margin. The parameter C is used to control the trade-off between the training error and the margin, which is a user-defined parameter.

By introducing Lagrange multipliers and exploiting the optimality constraints, the decision function Eq. (1) has the following explicit form:

$$f(X, a_i) = \mathrm{sign}\left(\sum_{i=1}^{l} a_i y_i K(X_i, X) + b\right) \tag{3}$$

In Eq. (3), a_i is the so-called Lagrange multiplier. It is a positive value and obtained by maximizing the dual function of Eq. (2), which has the following form:

$$R(a_i) = \sum_{i=1}^{l} a_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l} a_i a_j y_i y_j K(X_i, X_j) \tag{4}$$

with the following constraints:

$$\sum_{i=1}^{l} a_i y_i = 0 \tag{5}$$

$$0 \leqslant a_i \leqslant C$$

$K(X_i, X_j)$ is called as the kernel function. The mostly commonly used kernel function is the Gaussian kernel $K(X_i, X_j) = e^{\frac{-(X_i - X_j)^2}{\sigma^2}}$, where σ^2 is the kernel parameter and also determined by users.

3.1. *One-against-all*

In multi-class classification, the "one-against-all" method develops several binary SVM classifiers by using the training data points in one class with the actual label of $+1$ and all the remaining training data points with the actual label of -1 to separate one class to the others. In such a way, a total of k SVM classifiers need to be developed for k $(k > 2)$ classes. The new testing data point is then classified as the class whose classifier has the largest output value according to Eq. (3).

3.2. *One-against-one*

Different from "one-against-all", the "one-against-one" method develops the binary SVM classifiers by using the training data points in one class with the actual label of $+1$ and the training data points in another class with the actual label of -1. That is, one classifier is developed for each pair of classes. For a total of k classes, $\frac{k(k-1)}{2}$ SVM classifiers are developed. In the testing, the voting strategy is used. That is, for each of the developed SVM classifiers, if the output value is $+1$, the corresponding positive class gets one vote. Otherwise, if the output value is -1, the corresponding negative class gets one vote. Finally, the new testing data point is classified as the class with the largest votes. If there is the case where two classes obtain the same number of votes, the testing data point is arbitrary assigned to the class with the smaller index, the same as used in [20].

3.3. *Directed acyclic graph SVM (DAGSVM)*

In DAGSVM, SVM classifiers are developed using the same approach as used in "one-against-one" method. That is, one binary SVM classifier is developed for each pair of classes. But the directed acyclic graph architecture is used for the testing, instead of the voting strategy as used in "one-against-one". The architecture consists of $\frac{k(k-1)}{2}$ nodes and k leaves. Each node represents a different binary SVM classifier. Each leaf represents one predicted class. Starting from the rooted node, the testing data point is evaluated according to the nodes, finally reaching the leaf which indicates the predicted class. A detailed description of DAGSVM can be referred to the paper [5].

4. Experimental results

4.1. *Data collection*

The data set used is collected using two large databases: the Fixed Investment Securities database (FISD) and the Compustat database. The whole process in data collection involves the following three steps: the choice of rating agency, the selection of company universe and the collection of financial ratios. The selection of company universe as well as their issued bonds' ratings is performed using FISD. And the Compustat database is then used to collect the financial ratios for the selected companies.

For each bond, FISD provides the ratings of Standard&Poor, Moody and Fitch from the period of April, 1995 to February, 2002. Only the Standard&Poor's ratings are used. The company universe will be of the same country code and the same sector code. This is because companies from different countries or different sectors may display very different characteristics for the same financial variables, which will increase the difficulty of prediction. The company universe used consists of the US companies from the manufacturing sector with at least one Standard&Poor's rating given in the evaluated time period according to FISD database.

After obtaining the company universe and their issued bonds' ratings, the next step is to use the Compustat date to collect the financial variables which will be used as the inputs of the models. It should be noted that the previous selected companies using FISD that are not found in Compustat are removed. Following the recommendation by Tan [15], six categories of financial ratios are used as illustrated in Table 1. They are the interest coverage ratios, the leverage ratios, the profitability ratios, the liquidity ratios, the size ratios and the market ratios. The interest coverage ratios measure the extent to which the earnings of a company cover debt or interest. The leverage ratios measure the financial leverage created when a company borrows money. The profitability ratios measure the profits of a company relative to its assets or sales. The liquidity ratios measure the ability of the firm to meet future short-term financial obligations. The size ratios measure the size of the company. The market ratios are used to assess the value of a company that investors assign to. For assuring the accuracy of collection, the Compustat database is only used for collecting the financial variables. The financial ratios are then computed according to the corresponding formulae as illustrated in Table 1. Furthermore, the financial variables are collected in a quarterly way. Thus, in the development of the models the financial ratios one quarter before the ratings will be used as the inputs. This is in consistent with the fact that the rating agency usually gives the ratings based on the financial ratios one quarter ahead.

According to the above procedure, a total of 239 data points are collected. The number of data points in each rating is given in Table 2. It can be observed that the distribution of the data points is biased to the ratings of A, BBB, BB and B. Due to few data points, the classification of the ratings of AAA,

Table 1
The used financial ratios

Categories	Financial ratios	Formulae
Interest coverage	EBIT interest coverage	$\frac{\text{earnings before interest and taxes}}{\text{interest expense}}$
	EBITDA interest coverage	$\frac{\text{earnings before interest, taxes, depreciation and amortization}}{\text{interest expense}}$
Leverage	EBIT/total debt	$\frac{\text{earnings before interest and taxes}}{\text{total debt}}$
	Debt ratio	$\frac{\text{long term debt}}{\text{long term debt}+\text{equity }+\text{minority interest}}$
	Debt-equit-1	$\frac{\text{long term debt}}{\text{equity}}$
	Debt-equity-2	$\frac{\text{long term debt}}{\text{long term debt }+\text{equity}}$
	Net gearing	$\frac{\text{total liability} - \text{cash}}{\text{equity}}$
Profitability	Return on equity	$\frac{\text{net income}}{\text{average equity}}$
	Return on total assets	$\frac{\text{earnings before interest and taxes}}{\text{total assets}}$
	Operating income/sales	$\frac{\text{operating income before depreciation}}{\text{sales}}$
	Net profit margin	$\frac{\text{net income}}{\text{sales}}$
Liquidity	Current ratio	$\frac{\text{current assets}}{\text{current liability}}$
	Acid test ratio	$\frac{\text{current assets} - \text{inventory} - \text{accounts payable} - \text{taxes payable}}{\text{current liability}}$
Size	Total assets	total assets
	Market value	(price per share) $\times$ (number of shares outstanding)
Market	Market beta	snapshot taken on the last trading day
	Earnings per share	$\frac{\text{earnings applicable to common stock}}{\text{total number of shares outstanding}}$

Table 2
The number of data points in each rating

Ratings	Number of data points
AAA	4
AA	11
A	43
BBB	62
BB	42
B	59
CCC	14
CC	2
C	1
D	1

CC, C and D is not considered. That is, only the data points in the ratings of AA, A, BBB, BB, B and CCC are used, with a total of 6 classes. Finally, the z-score method is used to normalize each variable by subtracting the average and dividing the standard deviation of the variable from each instance.

Table 3
The predicted and actual ratings in "one-against-all"

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	6	1	0	0	1	0	75.00
A	1	38	0	0	1	0	95.00
BBB	1	5	46	3	1	0	82.14
BB	0	0	5	30	5	0	75.00
B	1	0	0	4	45	6	80.36
CCC	0	0	1	1	1	5	62.50
Total				—			81.73

Table 4
The predicted and actual ratings in "one-against-one"

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	7	1	0	0	0	0	87.50
A	3	36	0	0	1	0	90.00
BBB	3	5	45	3	0	0	80.36
BB	1	0	4	32	3	0	80.00
B	2	0	0	5	46	3	82.14
CCC	0	0	0	1	1	6	75.00
Total				—			82.69

4.2. Comparison result

In addition to 17 financial ratios, the previous rating of bonds is also used as the input of the models, with a total of 18 features. The addition of the previous rating is because it can significantly improve the generalization performance of the models as demonstrated by our experiment. For this small data set, the 8-fold cross validation method is used to evaluate the performance of the models. That is, the whole data set in each rating is firstly randomly and equally partitioned into 8 parts. 7 parts are used as the training set, and the remaining 1 part is used as the validation set. In the partition of the data set, the remainderof the partition is not used in the experiment. This process is repeated until each of the partitioned parts is used as the validation set. The best average result on the cross validation sets is used to evaluate the models.

For training SVM, Keerthi et al.'s modified sequential minimal optimization algorithm [21] is implemented and the program is developed using VC^{++} language. The Gaussian function is used as the kernel function of SVM. To choose the best free parameters of σ^2 and C in SVM, a range of values are investigated by multiplying 2 at each step: $\sigma^2 = [2^{-10},\ 2^{-9},\ \dots\ ,\ 2^{4}]$ and $C = [2^{-2},\ 2^{-1},\ \dots\ ,\ 2^{12}]$. For each binary SVM classifier, the values of σ^2 and C are used as the same. As the number of training data points in the positive class and in the negative class may be unbalanced, the ratio of C in the positive class (C_+) to that of the negative class (C_-) is always used as the inverse of the ratio of the number of training data points in the two classes ($\frac{C_+}{C_-} = \frac{l_-}{l_+}$, $C_+ = C$). As demonstrated by chew et al. [22], this strategy can improve the generalization performance of SVM.

A standard three-layer BP neural network, LR and ordered probit regression (OPR) are used as benchmarks. The LR and OPR program are both developed based on SAS. The BP neural network program is developed based on Matlab 5.1. neural network toolbox. In the BP neural network, the hidden nodes use the sigmoid transfer function and the output node uses the linear transfer function. There are 18 nodes in the input layer which is equal to the number of features. The output node is equal

Table 5
The predicted and actual ratings in DAGSVM

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	7	1	0	0	0	0	87.50
A	3	36	0	1	0	0	90.00
BBB	3	4	45	3	1	0	80.35
BB	1	0	4	33	1	1	77.50
B	2	0	0	4	49	1	87.50
CCC	0	0	1	0	1	6	75.00
Total			—				84.61

Table 6
The predicted and actual ratings in the BP neural network

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	7	1	0	0	0	0	87.50
A	4	35	1	0	0	0	87.50
BBB	3	6	43	3	1	0	76.78
BB	0	0	3	33	4	0	82.50
B	4	0	2	6	44	0	78.57
CCC	0	0	0	2	1	5	62.50
Total			—				80.28

to 6, the total number of ratings to be classified. The number of hidden nodes is determined by trial and errors. As shown in our experiment, the use of 8 hidden nodes in combination with the learning rate as 0.01, the momentum term as 0.9, the number of epochs as 1000 produces the best result on the 8-fold cross validation sets.

The predicted ratings and the actual ratings on the "one-against-all", "one-against-one", DAGSVM, the BP neural network, LR and OPR are respectively illustrated in Tables 3–8. The classification accuracy in each rating is also illustrated in the tables. The classification accuracy (CA) is calculated as

$$\text{CA} = \frac{\text{the number of cases correctly classified}}{\text{the total number of cases in the class}} \times 100\% \tag{6}$$

It can be observed that there is different classification accuracy in each rating, with the best accuracy in the rating of A and the worst performance in the rating of CCC. The classification accuracy on the whole data set is summarized in the tables. Obviously, all of the three SVM based methods ("one-against-all", "one-against-one" and DAGSVM) outperform the BP neural network, RL, and OPR. Among all the methods, there is the best performance in DAGSVM. The tables also show that the BP neural network performs better than RL and OPR. And there is similar performance between RL and OPR.

4.3. Analysis of features

In the above experiment, the used features are arbitrarily selected. Without a priori, it cannot be known which features are more important to the classification relative to the others. In this section, sensitivity analysis is used to analyze how important each feature is. The performance of SVM is also investigated by recursively removing irrelevant features based on the sensitivity analysis.

The idea of sensitivity analysis is to first train SVM using all available features. The optimal values of a_i and the separating hyperplane Eq. (3) are obtained by solving the quadratic programming Eq. (4).

Table 7
The predicted and actual ratings in LR

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	7	1	0	0	0	0	87.50
A	2	36	2	0	0	0	90.00
BBB	3	6	43	4	0	0	76.78
BB	0	0	4	29	3	4	72.50
B	1	0	0	5	41	9	73.21
CCC	0	0	0	0	2	6	75.00
Total			—				77.88

Table 8
The predicted and actual ratings in OPR

Actual ratings	Predicted ratings						Accuracy (%)
	AA	A	BBB	BB	B	CCC	
AA	5	3	0	0	0	0	62.50
A	2	36	2	0	0	0	90.00
BBB	0	7	45	4	0	0	80.35
BB	0	0	4	29	5	2	72.50
B	1	0	0	5	41	9	73.21
CCC	0	0	0	0	3	5	62.50
Total			—				77.40

The importance of features is then estimated by calculating the first partial derivative of the output with respect to the inputs. The higher the sensitivity value, the more sensitive the output to the input, the more important the corresponding feature. In SVM, the sensitivity value s_i is estimated by

$$s_k = \frac{\sum_{i=1}^{l} \left| \frac{\partial f(X_i)}{\partial X_{ik}} \right|}{l}, k = 1, \ \dots \ , \ n \tag{7}$$

where $\frac{\partial f(X_i)}{\partial X_{ik}} = \sum_{j=1}^{l} a_j y_j \frac{\partial K(X_j, X_i)}{\partial X_{ik}}$ [6]. For the Gaussian kernel $K(X_j, X_i) = e^{\frac{-\sum_{l=1}^{n}(X_{jl}-X_{il})^2}{\sigma^2}}$, $\frac{\partial K(X_j,X_i)}{\partial X_{ik}} = -\frac{2}{\sigma^2}(X_{jk} - X_{ik}) e^{\frac{-\sum_{l=1}^{n}(X_{jl}-X_{il})^2}{\sigma^2}}$.

In multi-class classification, the sensitivity values on individual SVM classifiers are averaged to obtain the sensitivity value of each feature according to

$$\overline{s_k} = \frac{\sum_{i=1}^{m} s_k^i}{m}, k = 1, \ \dots \ , \ n \tag{8}$$

where $m$is the total number of SVM classifiers. Given the values of $\overline{s_k}$, $k = 1, \ \dots \ , \ n$, the importance of features is ranked, with the most important feature corresponding to the largest value of $\overline{s_k}$ and the least important feature corresponding to the smallest value of $\overline{s_k}$.

The sensitivity value of each feature in DAGSVM is given in Table 9. The features are ranked (expressed by the indices) from the least important to the most important according to their corresponding sensitivity

Table 9
The sensitivity value of each feature in DASVM

Index	Inputs	Sensitivity	Sorted sensitivity (ascending)	Sorted index (ascending)
1	Previous rating	0.8316	0.0684	4
2	EBIT interest coverage	0.1583	0.0694	6
3	EBITDA interest coverage	0.1148	0.0910	9
4	EBIT/total debt	0.0684	0.0992	17
5	Debt ratio	0.1284	0.1148	3
6	Debt-equit-1	0.0694	0.1192	18
7	Debt-equity-2	0.1268	0.1268	7
8	Net gearing	0.1292	0.1284	5
9	Return on equity	0.0910	0.1292	8
10	Return on total assets	0.1613	0.1356	15
11	Operating income/sales	0.2082	0.1583	2
12	Net profit margin	0.1646	0.1613	10
13	Total assets	0.3068	0.1616	16
14	Market value	0.2638	0.1646	13
15	Market beta	0.1356	0.2082	11
16	Earnings per share	0.1616	0.2638	14
17	Current ratio	0.0992	0.3068	13
18	Acid test ratio	0.1192	0.8316	1

Table 10
The number of classification errors in DAGSVM

Number of features	Classification errors
18	32
16	31
14	31
12	30
10	30
9	30
8	29
7	29
6	28
5	30
4	30
3	31
2	34

value. As illustrated in the table, the most important feature is the previous rating with the sensitivity value of 0.8316. On the contrary, the least important feature is EBIT/total debt with the sensitivity value of 0.0684.

Based on the sensitivity values, irrelevant features with the smallest sensitivity values are removed. SVM is re-trained using surviving features. And the remaining features are further analyzed using sensitivity analysis. By recursively performing this procedure, the performance of SVM with respect to different feature subsets is obtained as listed in Table 10. It can be observed that the classification errors of DAGSVM firstly decrease by removing irrelevant features up to the use of 6 features and then lightly increase by further removing features. The 6 features corresponding to the best performance are the previous rating, total asset, market value, earnings per share, debt-equity-1, return on total assets. This demonstrates that the generalization performance of SVM can be improved by performing feature selection based on sensitivity analysis.

5. Conclusions

This paper applies SVM to deal with the bond rating problem. The "one-against-one", "one-against-all" and DAGSVM are used for dealing with multi-class classification in bond rating. The BP neural network, RL and OPR are used as benchmarks. By examining one real U.S. bond data set, the experiment shows that SVM significantly outperforms the benchmarks. The result shows that the classification accuracy of SVM is 84.61% in contrast to 80.28% in BP neural network, 77.88% in LR and 77.40% in OPR. It demonstrates the effectiveness of SVM in bond rating. Among the three SVM based methods, there is the best performance in DAGSVM as the classification accuracy is respectively 84.61%, 81.73% and 82.69% in DAGSVM, "one-against-all" and "one-against-one".

The sensitivity analysis is also used for analyzing the importance of features. Based on the sensitivity value, feature selection is performed by recursively removing irrelevant features with the smallest sensitivity value and re-training SVM using surviving features. The experiment shows that the generalization performance of SVM can be improved by performing feature selection based on sensitivity analysis.

One limitation of this paper is that only one bond data set is examined. Future work will further test the generalization performance of SVM in bond rating by using more data sets in different countries and different industries. Other features related to bond rating can also be tested in the future work.

References

[1] T. Joachims, *Transductive Inference for Text Classification Using Support Vector Machines*, In ICML-99: Proc. of the 16^{th} International Conference on Machine Learning, 1999, Bled, Slovenia, 200–209.

[2] M. Pontil and A. Verri, Support vector machines for 3D object recognition, *IEEE transactions on pattern analysis and machine intelligence* **20**(6) (June 1998).

[3] M. Brown, W. Grundy, D. Lin, N. Christianini, C. Sugnet, M. Ares, Jr. and D. Haussler, Support Vector Machine Classification of Microarray Gene Expression Data, University of California, Santa Cruz, Technical Report UCSC-CRL-99-09.

[4] V.N. Vapnik, *The Nature of Statistical Learning Theory*, New York: Springer, 2000.

[5] J.C. Platt, Nello Cristianini, and John Shawe-Taylor, Large Margin DAGs for multiclass classification, *Advances in Neural Information Processing Systems* **12** (2000), 547–553, MIT Press.

[6] L.J. Cao, K.S. Chua, Q.M. Gu and H. P. Lee, *Saliency Analysis of Support Vector Machines for Gene Selection in Tissue Classification*, accepted in Neural Computing & Applications, 2003.

[7] F.J. Fabozzi, *Bond markets, analysis, and strategies*, Upper Saddle River, N.J. : Prentice Hall, c2000.

[8] L. Fisher, Determinants of Risk Premium on Corporate Bonds, *Journal of Political Economy* (1959), 217–237.

[9] G.E. Pinches and K.A. Mingo, A Multivariate Analysis of Industrial Bond Ratings, *Journal of Finance* **28**(1) (1973), 1–18.

[10] H.L. Ederington, Classification Models and Bond Ratings, *Financial Review* **20**(4) (1985), 237–262.

[11] J.A. Gentry, D.T. Whitford and P. Newbold, Predicting Industrial Bond ratings with a probit model and funds flow components, *Financial Review* **23**(3) (1988), 269–286.

[12] J. Moody and J. Utans, Architecture Selection Strategies for Neural Networks Applications to Corporate Bond Ratings, in: *Neural Networks in the Capital Markets*, A. Refenes, ed., John Wiley, Chichester, 1995.

[13] J.C. Singleton and A.J. Surkan, Bond Rating with Neural Networks, in: *Neural Networks in the Capital Markets*, A. Refenes, ed., John Wiley, Chichester, 1995.

[14] Y.S. Kwon, I.G. Han and K.C. Lee, Ordinal Pairwise Paritioning (OPP) Approach to Neural Networks Training in Bond Rating, Intelligent Systems in Accounting, *Finance and Management* **6** (1997), 23–40.

[15] R.P.G.H. Tan, *Credit rating Prediction using Self Organizing Maps*, *Master's thesis*, Erasmus University of Rotterdam, 2000, available at http://www.eur.nl/few/people/jvandenberg/masters.htm (#34).

[16] X. Parra and X. Parra, *Qualitative Input Conditioning to Enhance RBF Neural Networks Generalization in Financial Rating Classification*, Available at http://www.vuse.vanderbilt.edu/˜qr-01/Parra,Agell.htm.

[17] J.J. Maher and T.K. Sen, Predicting Bond Ratings Using Neural Networks: A Comparison with Logistic Regression, Intelligent Systems in Accounting, *Finance and Management* **6** (1997), 59–72.

[18] J.W. Kim, Expert Systems for Bond Rating: A Comparative Analysis of Statistical, Rule-Based and Neural Network Systems, *Expert Systems* **10** (1993), 167–171.

[19] R. Chaveesuk, C. Srivaree-Ratana and A.E. Smith, Alternative Neural Network Approaches to Corporate Bond Rating, *Journal of Engineering Valuation and Cost Analysis* **2**(2) (1999), 117–131.
[20] C.W. Hsu and C.-J. Lin. A comparison of methods for multi-class support vector machines, *IEEE Transactions on Neural Networks* **13**(2) (March 2002).
[21] S.S. Keerthi, S.K. Shevade, C. Bhattaacharyya and K.R. K. Murthy, Improvements to Platt's SMO Algorithm for SVM Classifier design, *Neural Computation* **13** (2001), 637–649.
[22] H.G. Chew, R.E. Bogner and C.C. Lim, *Dual nu-support vector machine with error rate and training size biasing*, ICASSP 2001, page to appear, 2001.