
ELSEVIER
Volume 39
Issue 8
August 2012
ISSN 0305-0548
computers & operations research
Special Issue:
Advances of Operations Research in Service Industry
Guest Editors:
Li D. Xu, Shoubo Xu, Zude Zhou, Pan Wang and John Warfield
Editor: Stefan Nickel
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Computers & Operations Research 39 (2012) 1800–1811

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

A corporate credit rating model using multi-class support vector machines with an ordinal pairwise partitioning approach ☆

Kyoung-jae Kim [a], Hyunchul Ahn [b,*]

[a] Department of Management Information Systems, Dongguk University-Seoul, 3-26 Pil-Dong, Chung-Gu, Seoul 100-715, South Korea
[b] School of Management Information Systems, Kookmin University, 861-1, Jeongneung-Dong, Seongbuk-Gu, Seoul 136-702, South Korea

ARTICLE INFO

Available online 8 July 2011

Keywords:
Corporate credit rating
Support vector machines
Multi-class classification
Ordinal pairwise partitioning

ABSTRACT

Predicting corporate credit-rating using statistical and artificial intelligence (AI) techniques has received considerable research attention in the literature. In recent years, multi-class support vector machines (MSVMs) have become a very appealing machine-learning approach due to their good performance. Until now, researchers have proposed a variety of techniques for adapting support vector machines (SVMs) to multi-class classification, since SVMs were originally devised for binary classification. However, most of them have only focused on classifying samples into nominal categories; thus, the unique characteristic of credit-rating – ordinality – seldom has been considered in the proposed approaches. This study proposes a new type of MSVM classifier (named OMSVM) that is designed to extend the binary SVMs by applying an ordinal pairwise partitioning (OPP) strategy. Our model can efficiently and effectively handle multiple ordinal classes. To validate OMSVM, we applied it to a real-world case of bond rating. We compared the results of our model with those of conventional MSVM approaches and other AI techniques including MDA, MLOGIT, CBR, and ANNs. The results showed that our proposed model improves the performance of classification in comparison to other typical multi-class classification techniques and uses fewer computational resources.

1. Introduction

Corporate credit rating is a very important factor in the market of corporate debt. Information concerning corporate operations is often disseminated to market participants through the changes in credit ratings that are published by professional rating agencies, such as Standard & Poor's (S&P) and Moody's Investor Service. Since these agencies generally require large fees for their services and the periodically provided ratings sometimes do not reflect the default risk of the company at the time, it may be advantageous for bond-market participants to be able to classify credit ratings before the agencies publish the ratings. As a result, it is very important for the companies, especially financial companies, to develop a proper model of credit rating [1,68].

From a technical perspective, the credit rating constitutes a typical, multi-class, classification problem because the rating agencies generally have ten or more categories of ratings. For example, S&P's ratings range from AAA for the highest-quality bonds to D for the lowest-quality bonds. Professional rating agencies emphasize the importance of analysts' subjective judgments in determining credit ratings. However, in practice, a mathematical model that uses the financial variables of companies plays an important role in determining credit ratings, since it is convenient to apply and entails less time and cost. These financial variables include the ratios that represent a company's leverage status, liquidity status, and profitability status [1–4,68,69].

Several statistical and artificial intelligence (AI) techniques have been applied as tools for financial decision making such as stock market forecasting or credit ratings prediction [1,3,5]. Among them, the artificial neural networks have been widely used in the area of finance because of their broad applicability to many business problems and their preeminent ability to learn [6,7]. However, besides the risk of over-fitting, artificial neural networks also have many defects, including difficulty in determining the values of control parameters and the number of processing elements in the layer. Support vector machines (SVMs) have recently become popular as a solution to problems that are associated with prediction because of their robustness and high accuracy [8–12,70]. An SVM's solution may be globally optimal because SVMs seek to minimize structural risk. Conversely, the solutions found by artificial neural network models tend to fall into local optimum because they seek to minimize empirical risk. In addition, no parameters need to be tuned in SVMs, barring the

☆ This work was supported by the Korea Research Foundation Grant funded by the Korean Government (KRF-2009-332-B00104). This work was supported by the 2011 research fund of Kookmin University in Korea.
* Corresponding author. Tel.: +82 2 910 4577; fax: +82 2 910 4519.
E-mail addresses: kjkim@dongguk.edu (K.-j. Kim), hcahn@kookmin.ac.kr (H. Ahn).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.06.023

upper bound for non-separable cases in linear SVMs. However, SVMs were originally devised for binary classification; therefore, they are not naturally geared for multi-class classifications, which apply to credit ratings [13]. Thus, researchers have tried to extend the original SVM to multi-class classification.

Hitherto, a variety of techniques to extend standard SVMs to multi-class SVMs (MSVMs) have been proposed in the literature. These techniques include approaches that construct and combine several binary classifiers as well as approaches that directly consider all the data in a single optimization formulation. However, most published techniques have focused on classifying samples into nominal categories [8,14–21]. Even those prior studies that applied MSVMs to credit ratings also used standard MSVM models that were not designed to reflect the ordinal nature of this domain [1,3,22,23]. Furthermore, most of these studies tested at most a few types of MSVM.

In this study, we propose a novel computational approach for MSVMs, which takes into account the ordinal characteristics for efficiently and effectively handling multiple ordinal classes; we term the approach, ordinal multi-class support vector machine (or OMSVM, in short). Similar to traditional MSVMs, our model basically combines several binary SVM classifiers. However, it is different from the traditional approaches since it extends the binary SVMs using the ordinal pairwise partitioning (OPP) approach [24]. Using the latter approach, our model uses fewer classifiers, but nevertheless may more accurately predict classes because it exploits additional hidden information, namely, the order of classes. To validate the effectiveness of our model, we applied the model to a real-world case of bond rating in Korea. We compared the results of the model to those of traditional MSVM approaches. We also compared the results of the model to those of traditional techniques for credit ratings, such as multiple discriminant analysis (MDA), multinomial logistic regression (MLOGIT), case-based reasoning (CBR), and artificial neural networks (ANNs) [25–33]. In addition, to examine the effect of OPP in depth, we applied OPP to both MSVMs and ANNs, and we compared the prediction results that were generated by these two techniques.

The rest of this paper is organized as follows. The next section reviews the literature on SVMs and MSVMs, in addition to studies on credit ratings that employed data mining. In Section 3, our approach for ordinal multi-class classification is proposed. Section 4 describes the data and experiments for validating our model. In Section 5, the empirical results are summarized and discussed. The final section presents the conclusions and future research direction of this study.

2. Literature review

In this section, we introduce the basic concept of conventional SVM, and we summarize the studies that have attempted to extend the conventional SVM to multi-class classification. Then, we briefly review the studies on credit ratings that have used the techniques of data mining. We will also discuss the major studies in the literature that have adopted MSVMs to classify credit ratings.

2.1. Conventional (binary) SVM

The conventional SVM achieves classification by mapping the input vectors on to a high-dimensional feature space and by then constructing a linear model that implements nonlinear class boundaries in the original space. SVM employs an algorithm that finds a special kind of linear model, namely, the optimal hyperplane. The optimal hyperplane refers to the maximum-margin hyperplane, which yields the maximum separation between decision classes. Thus, the optimal hyperplane separates the training examples with the maximum distance from the separating hyperplane to the closest training data samples. Those training examples that are closest to the maximum-margin hyperplane are called support vectors. All other training examples, other than the support vectors, are useless for constructing the optimal hyperplane. As a result, it is possible for SVMs to effectively perform binary classification with a small size of training samples [1,13,34,35,70].

For the linearly separable case, a hyperplane, which separates the binary decision classes in the case of n attributes, can be represented as the following equation:

$$y = w_0 + \sum_{i=1}^{n} w_i x_i \tag{1}$$

where y is the outcome, x_i are the attribute values (i=1, …, n), and $\{w_i{:}i{=}0, \ldots, n\}$ are the $n+1$ weights to be learned by the learning algorithm. In Eq. (1), the weights $\{w_i{:}i{=}0, \ldots, n\}$ are the parameters that determine the hyperplane. As shown in Eq. (2), SVMs approximate the optimal hyperplane (i.e., the maximum-margin hyperplane) using the support vectors:

$$y = b + \sum \alpha_i y_i \mathbf{x}(i) \cdot \mathbf{x} \tag{2}$$

In Eq. (2), y_i is the class-value of the training example $\mathbf{x}(i)$ and $\bullet$ represents the dot product. The vector $\mathbf{x}$ represents a test example, and the vectors $\mathbf{x}(i)$ are the support vectors. In this equation, b and $\{\alpha_i\}$ are parameters that determine the optimal hyperplane. In theory, the problem of finding the support vectors and parameters b and $\{\alpha_i\}$ can be transformed into a linearly constrained quadratic programming (QP) problem [13,34]. For more details, please refer to Fig. 1.

As shown in Fig. 1, for the linearly separable case, we assume that all data is at least distance 1 from the hyperplane $\mathbf{w}^T\mathbf{x}_i + b = 0$. Then, given a training set of instance-label pairs $(\mathbf{x}_i, y_i)$, $i = 1, \ldots, m$ where $\mathbf{x}_i \in R^n$ and $y_i \in \{+1, -1\}$, the data points will be correctly classified by

$$\mathbf{w}^T\mathbf{x}_i + b \geq +1 \text{ for } y_i = +1 \tag{3}$$

$$\mathbf{w}^T\mathbf{x}_i + b \leq -1 \text{ for } y_i = -1 \tag{4}$$

These equations can be transformed into one set of inequalities of

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq +1 \tag{5}$$

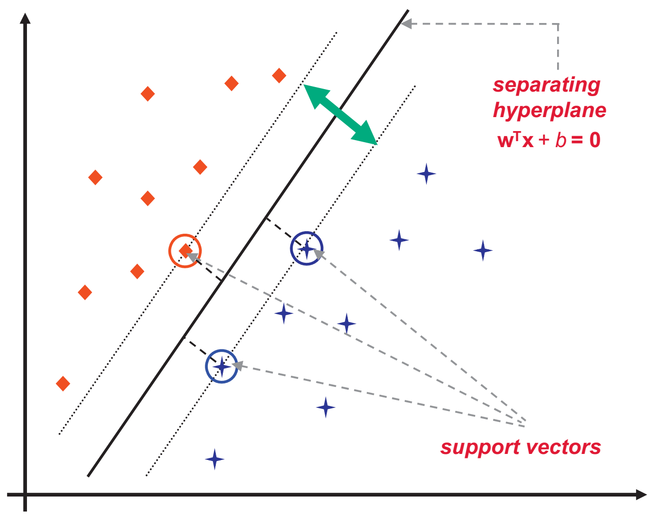

Fig. 1. Graphical interpretation of conventional SVM: a linearly separable case.

Table 1
MSVM techniques proposed in the literature.

Approach	Proposed techniques	No. of classifiers to be constructed[a]	Proposed literature
Constructing several binary classifiers	One-Against-All	k	Kreβel [42]
	One-Against-One	${}_kC$ $(=k(k-1)/2)$	Friedman [44]
	DAGSVM	${}_kC$	Platt et al. [15]
	Error-Correcting Output Codes	$\log_2 k \sim 2^{k-1}-1$	Klautau et al. [46]
Considering all the data at once	Weston and Watkins	1	Weston and Watkins [47]
	Crammer and Singer	1	Crammer and Singer [14]

[a] Assumed the multi-class classification problem that consists of k classes.

Here, SVM finds an optimal separating hyperplane with the maximum margin by solving the following quadratic optimization problem:

$$\mathrm{Min}_{\mathbf{w},b}\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w} \text{ subject to } y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i+b)-1 \geq 0 \tag{6}$$

However, the above concepts should be extended to a non-separable case because, in reality, there are few cases whose data points are linearly divided without any exception. To do this, we adopt non-negative slack variables, so Eq. (6) is transformed into Eq. (7). By solving Eq. (7), we can find the hyperplane that provides the minimum number of training errors [36]

$$\mathrm{Min}_{\mathbf{w},b,\xi}\Phi(\mathbf{w}) = \frac{1}{2}\mathbf{w}^{\mathbf{T}}\mathbf{w}+C\sum_{i=1}^{m}\xi_i \text{ subject to } y_i(\mathbf{w}^{\mathbf{T}}\mathbf{x}_i+b)+\xi_i-1 \geq 0,\ \xi_i \geq 0 \tag{7}$$

For the nonlinearly separable case, SVMs are able to undertake the classification by constructing a linear model that implements the nonlinear class boundaries by transforming the inputs into the high-dimensional feature space. In this case, Eq. (2) can be modified into a high-dimensional version, as presented in

$$y = b+\sum \alpha_i y_i K(\mathbf{x}(i),\mathbf{x}) \tag{8}$$

The function $K(\mathbf{x}(i),\mathbf{x})$ that transforms the input vector into a high-dimensional feature space is called the kernel function. To solve the optimization problem using Eq. (8), the kernel functions should be positive-definite functions that satisfy Mercer's theorem.[1] Common examples of such kernel functions include the following: the linear kernel, $K(x,y)=x^Ty$; the polynomial kernel, $K(x,y)=(x^Ty+1)^d$ (where d is the degree of the polynomial kernel); the Gaussian radial basis function (RBF), $K(x,y)=\exp(-1/\sigma^2(x-y)^2)$, where σ^2 is the bandwidth of the Gaussian RBF kernel. For the separable case, 0 is a lower bound on the coefficients $\{\alpha_i\}$ in Eq. (8). For the non-separable case, SVM can be generalized by placing an upper bound C on the coefficients $\{\alpha_i\}$ in addition to the lower bound [35]. The detailed process of formulating and solving SVMs has been discussed in prior studies, including [34,36,38,39]. For more details on the SVM algorithm, readers may refer to these references.

2.2. Multi-class support vector machines

The conventional SVMs were originally designed for binary classification, which requires only one classifier. Thus, for performing multi-class classifications, the conventional SVMs should be modified. Indeed, the extension of conventional SVMs to multi-class SVMs (MSVMs) is an ongoing topic of research. In the past five years, a variety of techniques have been proposed for implementing MSVMs. In general, there are two approaches for extending SVMs to multi-class problems. The first approach is to decompose the multi-class problems into several binary subproblems. In this approach, MSVMs can be implemented by constructing and combining several binary SVM classifiers. The second approach is to directly consider all the data in a single optimization formulation. In this case, modifying the conventional training algorithm of SVMs is required. There are different kinds of techniques in each approach [40,41]. Table 1 provides an overview of these proposed techniques for MSVMs.

Constructing several binary classifiers—One-Against-All (also known as 'One-versus-All', 'One-Against-Rest', and 'One-versus-Rest'): Conceptually, this is the simplest multi-class method. This method constructs k binary SVM classifiers for a k-class classification: class 1 (positive) versus all other classes (negative); class 2 versus all other classes; …; and class k versus all other classes [42].

The combined One-Against-All decision-function chooses the class for a sample that corresponds to the maximum of k binary classification functions that are specified by the farthest positive hyperplane. In the process, the decision hyperplanes are calculated by the k SVMs shift, which throws into question the optimality of the multi-class classification [43].

Constructing several binary classifiers—One-Against-One (also known as 'All-Against-All', 'One-versus-One', and 'Pairwise Classification'): In this method, the model constructs binary SVM classifiers for all pairs of classes; in total, there are ${}_kC_2=k(k-1)/2$ pairs. This means that for each pair of classes, a binary SVM classifier is constructed by solving the underlying optimization problem to maximize the margin between the two classes in the pair. The decision-function assigns an instance to a class that has the largest number of votes—the so-called *Max Wins* (*also known as* '*vote count*' *and* '*winner-takes-all*') strategy. If ties occur, then a sample will be assigned a label that is based on the classification resulted in by the farthest hyperplane [42,44].

Constructing several binary classifiers—DAGSVM: The third algorithm for constructing several binary classifiers is the directed acyclic graph SVM (DAGSVM). The training phase of this algorithm is similar to that of the One-Against-One method that employs multiple binary classifiers; however, DAGSVM uses a graph-visiting strategy for testing. The testing phase of DAGSVM requires the construction of a rooted binary-decision directed acyclic graph (DDAG) that uses ${}_kC$ classifiers. Each node of this graph represents a binary SVM for a pair of classes, e.g., (p, q). At the topologically lowest level, there are k leaves corresponding to k classification decisions. Every non-leaf node (p, q) has two branches: the left branch corresponds to the decision, "not p", while the right branch corresponds to "not q." (Please refer to 'DAGSVM' in Fig. 3.) The choice of the class order in the DDAG list can be arbitrary, as empirically demonstrated by [15].

An advantage of using a directed acyclic graph (DAG) is that some theoretical generalizations can be established. As of yet, there are no similar theoretical results for the One-Against-All

[1] Mercer's theorem gives the necessary and sufficient conditions of $K(x,y)$: $\chi\times\chi\rightarrow R$. For more detailed information on the relationship between the kernel function and Mercer's theorem, please refer to pp. 32–35 in Chapter 2 of [37].

Classifier: class p (+1) vs. class q (–1)

	Classifier 1 : 1 vs. (2,3,4)	Classifier 2 : 2 vs. (1,3,4)	Classifier 3 : 3 vs. (1,2,4)	Classifier 4 : 4 vs. (1,2,3)	Classifier 5 : (1,2) vs. (3,4)	Classifier 6 : (1,3) vs. (2,4)	Classifier 7 : (1,4) vs. (2,3)
Class 1	+1	–1	–1	–1	+1	+1	+1
Class 2	–1	+1	–1	–1	+1	–1	–1
Class 3	–1	–1	+1	–1	–1	+1	–1
Class 4	–1	–1	–1	+1	–1	–1	+1

Fig. 2. Error-correcting codes for a 4-class classification.

and One-Against-One methods. In addition, the testing time for DAGSVM is less than that for the One-Against-One method [15].

Constructing several binary classifiers—Error-Correcting Output Codes: The Error-Correcting Output Coding (ECOC) approach, adopted from the digital communication theory, fuses the decisions that are generated by individual SVM classifiers [45]. After completing the training, this method constructs a code matrix, where row i represents the code-vector of class i, and column j represents a classifier assignment. Then, to determine the class, ECOC compares the error-correcting codes with each row of the matrix. In general, the row with the minimum hamming distance is selected [19,46].[2] Fig. 2 shows an example of the error-correcting codes for a 4-class classification where a classifier (p versus q) responds with $+1$ when the output class is p, and -1 when the output class is q.

Where there are k classes, the number of binary classifiers that the ECOC approach uses for fusing decisions can be varied from $\log_2 k$ to $2^{k-1}-1$. For example, where there are four classes, we can make a decision using only 2 classifiers ($\because \log_2 4=2$) because they can generate 4 different cases—$(+1,+1)$, $(+1,-1)$, $(-1,+1)$, and $(-1,-1)$. However, in most cases, the more classifiers are used, the better the prediction performance is. Thus, in the previous example, it is better to apply all seven classifiers ($\because 2^{4-1}-1=7$) as indicated in Fig. 2 in order to improve the prediction accuracy of 4-class classification problems. However, in this case, the computational time for training and testing may be extended [41].

Directly considering all the data at once—method of Weston and Watkins: This approach may be interpreted as a natural extension of the binary SVM classification problem. Here, in the case of k classes, one has to solve a single quadratic optimization problem of size $(k-l)n$, which is identical to a binary SVM for the case when $k=2$ [47]. In a slightly different formulation of the QP problem, a bounded-formulation decomposition technique can significantly speed-up the solution of the optimization problem [16,48].

Directly considering all the data at once—method of Crammer and Singer: This method is similar to the previous one (i.e., the method of Weston and Watkins). It requires solving a single QP problem of size $(k-l)n$; however, it uses fewer slack variables in the constraints of the optimization problem [14]. Similar to the method of Weston and Watkins, the use of decompositions can significantly speed-up the solution of the optimization problem [16,48].

Fig. 3 graphically shows the differences between One-Against-All, One-Against-One, DAGSVM, and the techniques such as those of 'Weston and Watkins' and 'Crammer and Singer'. As shown in Fig. 3, the classifiers constructed by either the method of Weston and Watkins or the method of Crammer and Singer are very complex; therefore, these methods are generally more intensive in terms of computational resources [16,40]. As a result, the approach that decomposes the multi-class problem into binary subproblems is more frequently used in practice [40]. Therefore, the proposed technique in our study is also based on this approach.

2.3. Credit rating using data mining techniques

Major studies that use data-mining techniques for bond-rating predictions can be found in the literature [1–3,22–24,49–54]. Table 2 summarizes various studies on credit rating and their proposed techniques. Published research can be conceptualized as evolving in three phases; correspondingly, the studies listed in Table 2 can be categorized into three groups. Early investigations of credit rating techniques mainly focused on the applicability of statistical techniques, such as multiple discriminant analysis (MDA) and logistic regression analysis (LogR). The second phase of research on credit rating featured the application of typical techniques of AI, such as artificial neural networks (ANNs) and case-based reasoning (CBR). From Table 2, we can also find that in this phase, backpropagation neural networks (BPNs), a kind of ANN, was most frequently applied. However, BPNs suffer from issues relating to the selection of a large number of control parameters that pertain to the relevant input variables, hidden layer size, learning rate, and momentum term. In addition, they require a large amount of data for the training model due to the constraint on degrees of freedom. To overcome these limitations, recent studies have sought to apply MSVMs for credit rating.

The study of Huang et al. [3] pioneered the adoption of MSVMs for building prediction models of credit rating. Huang et al. experimented with various techniques of MSVM, including One-Against-One and the method of Crammer and Singer. They also experimented with different parameters for finding the optimal MSVM model. Finally, they opted for the method of Crammer and Singer, using an RBF kernel function with $\sigma^2=10$ and $C=1000$. They found that this MSVM model outperformed not only BPNs, but also LogR, in predicting the bond ratings for Taiwan and the US.

Cao et al. [1] applied One-Against-All, One-Against-One, and DAGSVM to predict the S&P's bond ratings. For the kernel function, the Gaussian RBF was applied and the optimal parameters of σ^2 and C were derived from a grid-search strategy. Cao et al. [1] found that DAGSVM performed the best among all three methods, and that all kinds of MSVM approaches outperformed other comparable multi-class classification techniques, including LogR, ordered probit regression (OPR), and BPN.

[2] The hamming distance between two strings is calculated by counting the number of positions for which the corresponding symbols are different. For example, the hamming distance between '110' and '101' is 2.

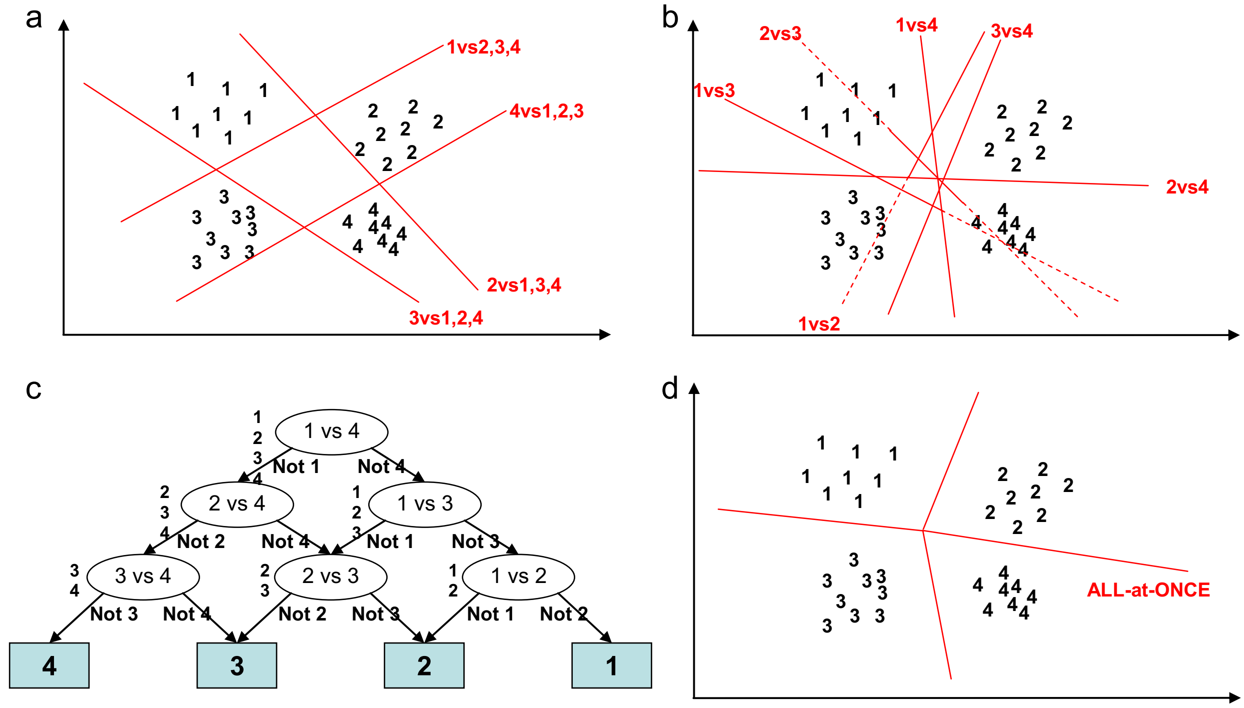

Fig. 3. Multi-class support vector machines applied to a four-class classification problem.

Table 2
Prior studies on credit rating using data mining techniques.

Stage	Study	Proposed technique(s)	Benchmark
The 1st stage	Pinches and Mingo [49]	MDA	N/A
	Belkaoui [50]	MDA	N/A
	Ederington [51]	ProR, LogR	N/A
The 2nd stage	Kim [52]	BPN, RBS	LinR, MDA, LogR
	Moody and Utans [53]	BPN	N/A
	Kwon et al. [24]	BPN	MDA
	Chaveesuk et al. [54]	BPN, RBF, LVQ	LogR
	Shin and Han [2]	CBR	MDA
The 3rd stage	Huang et al. [3]	MSVM	LogR, BPN
	Cao et al. [1]	MSVM	LogR, OPR, BPN
	Chen and Shih [22]	MSVM	BPN
	Lee [23]	MSVM	MDA, CBR, BPN

BPN: backpropagation neural networks, RBS: rule-based systems, LinR: linear Regression, LogR: logistic regression, ProR: ordered probit regression, RBF: radial basis function, LVQ: learning vector quantization, MDA: multiple discriminant analysis, and MSVM: multi-class support vector machines.

Chen and Shih [22] adopted the One-Against-One approach to build an automatic-classification model for credit ratings of issuers in Taiwan. Similar to the study of Cao et al. [1], they also adopted a Gaussian RBF kernel function and a grid-search strategy to determine the optimal parameters. They found that the MSVM model was statistically superior to the BPN and LogR models.

Lee [23] applied the same approach as [22] for building a corporate credit-rating prediction model for Korean companies. He also adopted a Gaussian RBF kernel function and a grid-search strategy. The experimental results showed that the MSVM model significantly outperformed BPNs, MDA, and CBR.

3. A novel approach of MSVMs for credit rating: OMSVM

As mentioned in the previous section, MSVMs have recently been receiving attention from researchers, who investigate achieving effective and efficient classifiers for credit ratings. Several articles that validate the applicability of MSVMs to the credit-rating domain have already been published. However, it is worth noting that most of these studies have merely adopted those MSVM techniques that had been developed by other researchers, without any modification or improvement. Furthermore, the techniques that were applied in these studies were designed for those multi-class classification problems where classes were either nominal or categorical but not ordinal. Thus, an appropriate modification of conventional MSVM techniques, which takes into account ordinality, may improve the performance of classifiers for credit rating.

Consequently, we propose a new type of MSVM technique that is optimized for the ordinal multi-class classification problems, such as the credit-rating problem. Our approach, named ordinal multi-class support vector machines (OMSVM), is a hybrid algorithm that applies the ordinal pairwise partitioning (OPP) technique to MSVMs.

OPP is an approach proposed by Kwon et al. [24] for enhancing the performance of ANN models of ordinal multi-class classifications. Kwon et al. [24] observed that ANNs that are designed for prediction in multi-class classification problems generally perform worse than when several binary ANN classifiers are combined. Thus, they proposed a new method, called ordinal pairwise partitioning (OPP), which considers the order of classes while combining several binary ANN classifiers. As explained before, SVMs were originally binary classifiers. Hence, they are well-suited to the use of OPP in the creation of a new type of MSVM model.

Thus, in this study, we propose the ordinal pairwise partitioning (OPP) approach as a tool for upgrading conventional multi-class SVM models to appropriately deal with ordinal classes. In an ordinal and pairwise manner that is in accordance with the output classes, the OPP approach partitions the dataset into data-subsets with reduced classes. As shown in Table 3, depending on the methods applied for partitioning and fusion, there are four types of OMSVM.

For the partitioning method, there are the One-Against-The-Next and One-Against-Followers approaches. The One-Against-The-Next

Table 3
Four types of OMSVM and their example processes for 4-class classification.

		Partitioning method	
		One-Against-The-Next	One-Against-Followers
Fusing method	Forward	– *Trains the following classifiers* : (1vs2), (2vs3), (3vs4) – *Apply the classifier (1vs2)* • Determine the class 1 – *Apply the classifier (2vs3)* • Determine the class 2 – *Apply the classifier (3vs4)* • Determine the class 3 and 4	– *Trains the following classifiers* : (1vs2,3,4), (2vs3,4), (3vs4) – *Apply the classifier (1vs2,3,4)*→Determine the class 1 – *Apply the classifier (2vs3,4)* • Determine the class 2 – *Apply the classifier (3vs4)* • Determine the class 3 and 4
	Backward	– *Trains the following classifiers* : (1vs2), (2vs3), (3vs4) – *Apply the classifier (3vs4)* • Determine the class 4 – *Apply the classifier (2vs3)* • Determine the class 3 – *Apply the classifier (1vs2)* • Determine the class 1 and 2	– *Trains the following classifiers* : (1vs2), (1,2vs3), (1,2,3vs4) – *Apply the classifier (1.2,3vs4)* • Determine the class 4 – *Apply the classifier (1,2vs3)* • Determine the class 3 – *Apply the classifier (1vs2)* • Determine the class 1 and 2.

method is similar to One-Against-One, but much more efficient. In the case of One-Against-One, all the classifiers for each pair of classes should be developed. However, in One-Against-The-Next, the binary classifiers are constructed only for the pairs $\{(i,i+1): i=1,2,\ldots,k-1\}$, where k is the total number of classes. Consequently, One-Against-The-Next constructs only $k-1$ binary classifiers, when there are k classes.

In contrast to One-Against-The-Next, One-Against-Followers is similar to – but slightly more efficient than – One-Against-All. In the case of One-Against-Followers, the binary classifiers are constructed for the pairs $\{(i,j): i=1,2,\ldots,k-1, j=\cup_{m=i+1}^{k} m\}$, where k is the total number of classes. As a result, where there are k classes, One-Against-Followers also constructs only $k-1$ binary classifiers, although One-Against-All constructs k classifiers.

Regarding methods of fusion, there are forward and backward methods, named in accordance with the 'direction of reasoning'. The forward method fuses the binary classifiers in the 'forward' direction, i.e., it determines the highest level of classes first, and the lowest level last. By contrast, the backward method combines the binary classifiers in the reverse direction, i.e., it determines the lowest level of classes first, and the highest level last.

The process of OMSVM consists of two phases: (1) preparation and (2) interpretation. In the preparation phase, OMSVM constructs the individual binary classifiers using the training dataset. To elaborate, OMSVM first divides the whole training dataset into $k-1$ groups, in accordance with the partitioning method (either One-Against-The-Next or One-Against-Followers). Then, OMSVM trains $k-1$ binary SVM models with each of the above data-subsets. For example, when the One-Against-The-Next approach is used for four-level classification problems, the first phase of OMSVM produces three binary classification models: Model 1 for the pair of classes (1, 2); Model 2 for the pair of classes (2, 3); Model 3 for the pair of classes (3, 4).

In the interpretation phase, OMSVM determines the class for the input data using the binary classifiers that are built in the first phase. To do this, it fuses the binary classifiers either in the forward or backward direction. In the case of the above example, the forward method begins with Model 1. If a test datum is put into class 1 by Model 1, then it is deemed 'class 1'. Otherwise, the test datum is passed on to Model 2. If it is put into class 2 by Model 2, then it is deemed 'class 2'. Otherwise, Model 3 applies. In Model 3, the test datum is finally classified as either 'class 3' or 'class 4'. Using the same reasoning, the backward method starts with Model 3. That is, if a test datum is put into class 4, then we regard it as belonging to 'class 4'. Otherwise, the test datum is passed on to the next SVM model. The remaining procedure is the same as that of the forward method but in the reverse order.

Table 3 includes examples of the binary classifiers and the order of their application for a four-class classification problem. Fig. 4 graphically shows the mechanism of each type of OMSVM. The figure helps to clarify the differences between the four types of OMSVM.

4. Experimental design

4.1. Research data

To validate our model, we applied it to a real-world case of credit rating in Korea. Our application is in bond rating, which is the most frequently studied area of credit rating for specific debt issues or other financial obligations. The research data were collected from National Information and Credit Evaluation, Inc., a major bond-rating company in Korea. We obtained the bond-ratings for the year 2002 and various financial variables for 1295 companies from the manufacturing industry in Korea. In the Korean bond-rating market, bond ratings are divided into five classes: A1, A2, A3, B, and C. However, we adjusted our data to four classes by combining the B and C ratings into one group because the respective numbers of companies were very small. Moreover, both B and C ratings are usually treated similarly, i.e., as being equivalent to junk bonds in the market.

The original data consisted of 39 financial-ratio variables that were known to affect bond ratings, as documented in the existing literature. Among them, we selected 36 variables by applying the

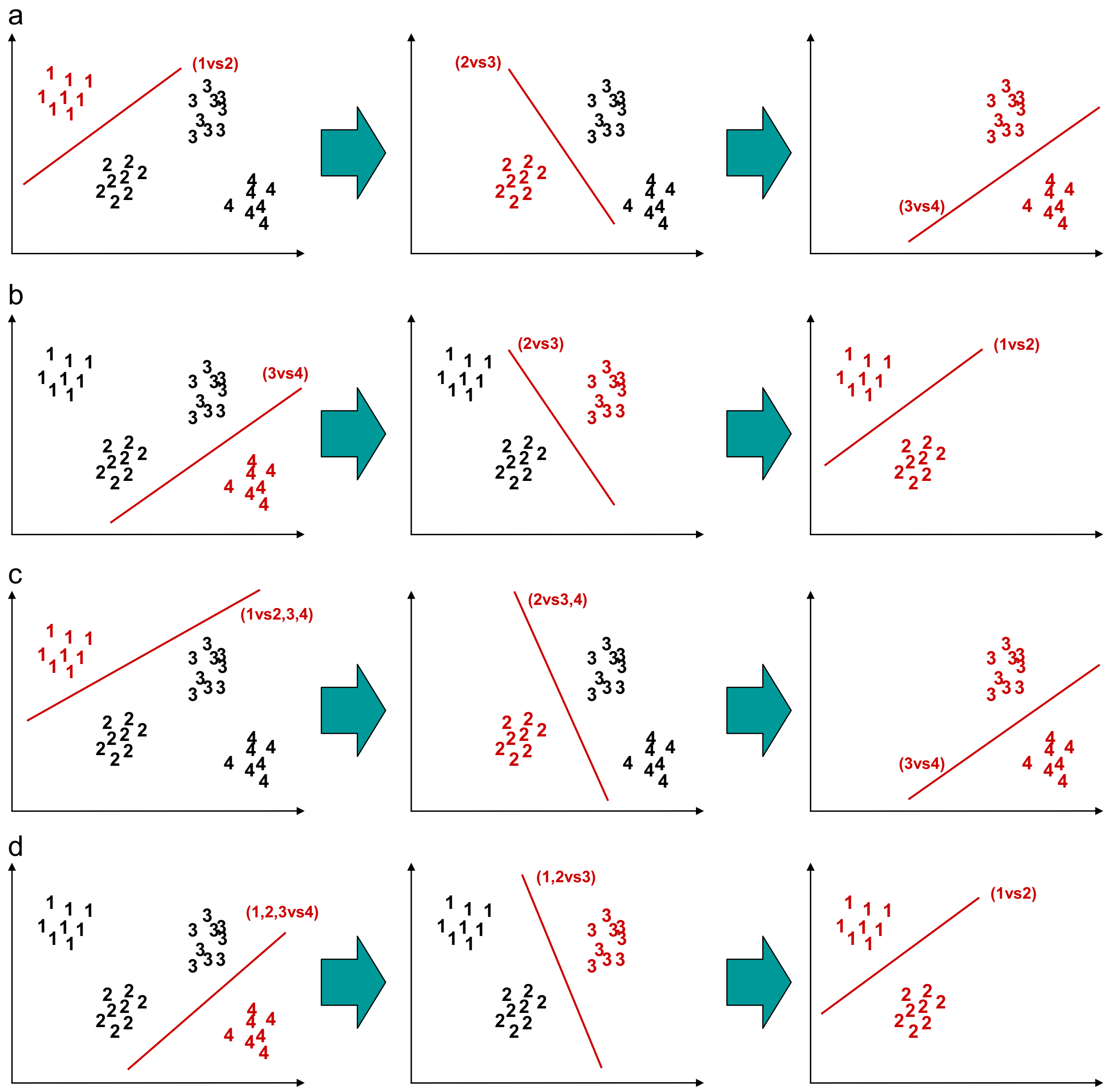

Fig. 4. Differences between four types of OMSVM: graphical presentation. (a) One-Against-The-Next + Forward approach, (b) One-Against-The-Next + Backward approach, (c) One-Against-Followers + Forward approach and (d) One-Against-Followers + Backward approach.

independent-samples t-test and, finally, selected 14 variables that proved, through the stepwise discriminant analysis method, to be the most influential in bond rating. As the stepwise method, we used "Wilk's lambda', and we used 'F-value' as the criteria for determining the entry or removal of the input variables. The selected variables are presented in Table 4.

In addition, we applied min–max normalization to all input variables in order to mitigate the size effect. Min–max normalization performs a linear transformation on the original data. Suppose that min_X and max_X are the minimum and maximum values of attribute X. Min–max normalization maps value x of X–x' in the range [new_min$_X$, new_max$_X$] by computing Eq. (9):

$$x' = \frac{x - min_X}{max_X - min_X}(\text{new\_max}_X - \text{new\_min}_X) + \text{new\_min}_X \tag{9}$$

Min–max normalization is often applied to enhance the performance of the prediction model because it ensures that the larger value input features do not overwhelm smaller value input features [55]. In this study, all features of our datasets range from 0 to 1.

Twenty percent of the data for each class was used for validation, and the remaining eighty percent was used for training. To overcome the scarcity of samples, we adopted the five-fold cross-validation. As a result, we produced five experimental datasets #1 to #5.

4.2. Experimental design

To thoroughly validate the superiority of our model's performance, we applied our proposed model (OMSVM) as well as six MSVM techniques, namely: (1) One-Against-All; (2) One-Against-One; (3) DAGSVM; (4) ECOC; (5) the method of Weston and Watkins; (6) the method of Crammer and Singer. In the case of the MSVM models, the linear kernel, polynomial kernel, and Gaussian radial basis function were used as the kernel functions. Tay and Cao [56] showed that the upper bound C and the kernel parameter play important roles in determining the performance

Table 4
List of the selected variables.

Variables	Description	Definition
SHEQ	Shareholder's equity	A firm's total assets minus its total liabilities
SALE	Sales	Sales
DEBT	Total debt	Total debt
SAPE	Sales per employee	Sales/the number of employees
NIPS	Net income per share	Net income/the number of issued shares
YEAR	Years after foundation	Years after foundation
AETA	Accumulated earning to total asset	Accumulated earning/total asset
BDRA	Borrowings-dependency ratio	Interest cost/sales
FCTC	Financing cost to total cost	Financing cost/total cost
FIRA	Fixed ratio	Fixed assets/(total assets − debts)
IACA	Inventory assets to current assets	Inventory assets/current assets
SBTB	Short-term borrowings to total borrowings	Short-term borrowings/total borrowings
CFTA	Cash flow to total assets	Cash flow/total assets
OACF	Cash flow from operating activity	Cash flow from operating activity

Definition of terms: Accumulated earnings: revenue or earnings that are received by a company, but are not paid out as dividends to the investors of the company. Sometimes they are referred to as 'earned surplus'. Financing cost: the price of obtaining loan capital. Inventory assets: the value of the assets in the form of inventory. Current assets: the value of all assets that are reasonably expected to be converted to cash within one year in the normal course of business.

of SVMs. The improper selection of these two parameters can cause the problems of over-fitting or under-fitting. The literature offers few pointers on determining the parameters of SVM. Therefore, for selecting the optimal values for achieving the best prediction performance, this study varied the parameters including the following: the upper bound, C; the degree of the polynomial kernel function, d; the bandwidth of the Gaussian RBF kernel function, σ^2. For implementing One-Against-All, One-Against-One, DAGSVM, and ECOC, we developed post-processing software that combines, in predefined ways, the results generated from multiple binary SVM classifiers. It was written in Microsoft Visual Basic for Applications for Excel 2003. We adopted LIBSVM 2.6 provided by Chang and Lin [57] as a library for binary SVM classification. In addition, we used BSVM 2.06 software to execute multi-class classification as per the methods of Weston and Watkins and of Crammer and Singer [58].

To validate the usefulness of our model as a general tool for credit rating, we also experimented with four statistical or AI techniques that had been adopted in prior studies on credit rating: (1) multiple discriminant analysis (MDA); (2) multinomial logistic regression (MLOGIT); (3) case-based reasoning (CBR); (4) artificial neural networks (ANNs). For each of MDA and MLOGIT, we used trial-and-error to select the best model. For CBR, the 1-NN (one nearest-neighbor) approach was applied, and the similarity between the cases was calculated using the Euclidean distance. In the case of ANNs, we adopted a standard three-layer back-propagation network and set the number of nodes in the hidden layer as being one of 7, 14, 21, and 28. For the stopping criteria of ANNs, this study allowed 50 learning epochs and set both the learning rate and momentum term to 0.1. The hidden nodes used the sigmoid transfer function, and the output node used the linear transfer function. This study allowed 14 input nodes because 14 input variables were employed. We used SPSS for Windows 14.0 for MDA and MLOGIT, and Neuroshell R4.0 for ANN. For the experiment with CBR, we used our own software, which is written in Microsoft Visual Basic for Applications for Excel 2003.

Finally, we also experimented with the multi-class classification model that applied the main idea (ordinal pairwise partitioning) of our proposed model to ANN: it is the model in the literature proposed by [24]. By comparing the prediction results of this comparative model (for simplicity, we refer to it as 'OMANN'—ordinal, multi-class, artificial neural network) to those of our proposed model, we sought to investigate the effects of OPP on SVM and ANN.

Table 5
Experimental results of OMSVM.

Dataset	One-Against-The-Next				One-Against-Followers			
	Forward		Backward		Forward		Backward	
	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)
1	65.57	65.50	65.57	65.50	73.96	66.28[a]	73.19	65.89
2	78.30	64.34	80.14	65.12[a]	82.45	65.12[a]	80.62	64.73
3	66.35	69.77	66.06	69.38	66.15	71.32[a]	66.06	70.93
4	74.64	68.22	74.25	68.22	82.35	68.60[a]	80.14	68.60[a]
5	74.16	68.99[a]	73.87	67.44	73.96	68.60	67.89	67.83
Avg.	**71.80**	**67.36**	**71.98**	**67.13**	**75.77**	**67.98**[a]	**73.58**	**67.60**

[a] The best performance for each data set.

5. Experimental results

To compare the performance of each algorithm, we adopted the hit-ratio as the performance measure. Simply put, the hit-ratio means the ratio of the corrected cases over all cases. The ratio is defined in

$$CR = \frac{1}{n}\sum_{k=1}^{n} CA_i;\ CA_i = 1 \text{ if } PO_i = AO_i,\ 0 \text{ otherwise} \tag{10}$$

In Eq. (10), CR is the classification accuracy rate of the test-set, CA_i is the classification accuracy of the ith case of the test-set denoted by either 1 or 0 ('correct' = 1, 'incorrect' = 0), PO_i is the predicted outcome for the ith case, and AO_i is the actual outcome for the ith case.

Table 5 shows the hit-ratios of the proposed model, OMSVM. As shown in this table, the One-Against-Followers + Forward approach yielded the best performance (67.98%) on average, and also yielded the best performance for all the datasets, except for 'dataset #5'. When the method of fusion was fixed, the prediction accuracy of the One-Against-Followers approach was always greater than that of One-Against-The-Next (67.98% > 67.36% for the forward strategy, and 67.60% > 67.13% for the backward strategy). Further, for the same partitioning method, the forward method of fusion always outperformed the backward method (67.36% > 67.13% for One-Against-The-Next, and 67.98% > 67.60% for One-Against-Followers).

Table 6
Experimental results of conventional MSVM algorithms.

Dataset	Constructing several binary classifiers								Considering all the data at once			
	One-Against-All		One-Against-One		DAGSVM		ECOC		Weston and Watkins		Crammer and Singer	
	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)	Train (%)	Valid (%)
1	73.10	65.50	75.80	65.12	65.19	64.73	73.00	66.28	67.12	65.89[a]	81.58	65.12
2	76.37	63.18	84.09	65.12[a]	69.62	65.12[a]	79.85	61.63	75.99	64.73	63.74	62.79
3	63.36	64.34	67.21	69.38	67.60	69.77[a]	62.97	65.12	66.44	69.38	63.26	64.73
4	80.04	65.12	76.47	68.22	76.18	68.60	73.10	65.12	81.20	68.99[a]	86.40	66.67
5	64.80	63.57	75.31	68.22[a]	75.60	68.22[a]	62.01	64.34	67.31	66.28	64.51	65.12
Avg.	**71.53**	**64.34**	**75.78**	**67.21**	**70.84**	**67.29[a]**	**70.18**	**64.50**	**71.61**	**67.05**	**71.90**	**64.89**

[a] The best performance for each data set.

In Table 6, the results of other MSVM algorithms are presented. The techniques that used the strategy of constructing several binary classifiers yielded an accuracy of 64.34–67.29%. Among them, with regard to the validation dataset, DAGSVM (67.29%) yielded the best performance, while One-Against-One (67.21%) came second. By contrast, ECOC (64.50%) and One-Against-All (64.34%) performed the worst. The ordering of prediction accuracies for MSVMs was exactly the same as that revealed in Cao et al. [1]. We believe this phenomenon is due to the differences in the numbers of classifiers. That is, when there are k classes, DAGSVM and One-Against-One determine the class for new data by referencing as many as ${}_kC = k(k-1)/2$ classifiers, but One-Against-All refers only to fewer (just k) classifiers. This gap in information seems to affect the prediction accuracy. The methods that simultaneously consider all data yielded accuracies between 64.89% and 67.05%. In our experiment, the method of Weston and Watkins (67.05%) outperformed that of Crammer and Singer (64.89%) but underperformed the DAGSVM (67.29%) and One-Against-One (67.21%) approaches.

Table 7 presents the prediction results of MDA, MLOGIT, CBR, and ANN. As shown in this table, the prediction accuracy of ANN (65.66%) was the highest, and that of CBR (51.40%) was the lowest. The result of MLOGIT (65.43%) was slightly lower than that of ANN, but the accuracy of MDA (63.10%) was quite a bit lower than those of ANN and MLOGIT.

In Table 8, the results of OMANN – the ANN model that adopted OPP – are presented. OMANN yielded a prediction accuracy that ranged between 66.43% and 67.05%. Considering that the prediction accuracy of conventional ANN was 65.66% for the validation dataset, the ordinal pairwise partitioning approach does seem to improve the performance of SVMs as well as that of ANNs. Nevertheless, OMANN's best prediction accuracy (67.05%) was lower than even the worst result under OMSVM (67.13%). Thus, it is believed that for bond rating, OMSVM would be more effective than OMANN.

It is also interesting that the partitioning approach that yields the best performance differs between OMSVM and OMANN. As indicated in Table 5, One-Against-Followers outperformed One-Against-The-Next in OMSVM. However, in the case of OMANN, One-Against-The-Next outperformed One-Against-Followers. In [24], we can also find the same pattern. We believe this phenomenon is caused by the fundamental difference between ANN and SVM. As mentioned earlier, ANN seeks to minimize empirical risk, i.e., it is designed to minimize the training error by repeated learning. Thus, ANN is easily affected by disproportionate sample sizes. By contrast, SVM is basically free from the problem of sample disproportionateness because SVM just refers to a small subset of training samples, which are called support vectors. In other words, SVM seeks to minimize the structural risk [1]. While the One-Against-Followers approach may provide more detailed information for classification, it distorts the proportion of the training sample. As a result, SVM, by its very nature, can fully exploit the One-Against-Followers approach. However, under the same approach in the case of ANN, training may be misled, resulting in low prediction performance.

Table 7
Experimental results of other comparative algorithms.

Dataset	MDA		MLOGIT		CBR	ANN		
	Train (%)	Valid (%)	Train (%)	Valid (%)	Valid (%)	Train (%)	Test (%)	Valid (%)
1	65.86	59.30	67.70	63.95	48.45	70.22	64.73	64.73[a]
2	64.90	62.02	67.98	63.57	47.67	71.76	65.50	65.12[a]
3	64.51	69.38[a]	65.96	68.99	54.26	69.45	63.18	66.67
4	64.80	65.50	67.79	67.44[a]	56.98	69.19	64.73	67.44[a]
5	65.09	59.30	66.15	63.18	49.61	67.14	68.60	64.34[a]
Avg.	**65.03**	**63.10**	**67.12**	**65.43**	**51.40**	**69.55**	**65.35**	**65.66[a]**

[a] The best performance for each data set.

Table 9 summarizes the averages of the results of Tables 5–7. As we can see from Table 9, our proposed model (OMSVM) exhibited the best performance among all the algorithms, including the conventional MSVM algorithms and other AI or statistical multi-class classification techniques. In particular, our model used the smallest number of classifiers among the approaches that construct several binary classifiers. In this experiment, OMSVM used just three binary classifiers, which are smaller than those used by ECOC (seven), One-Against-One (six), DAGSVM (six), and even One-Against-All (four). Thus, we may conclude that our proposed model is both effective and efficient for ordinal multi-class classification.

The McNemar tests were used to examine whether or not the predictive performance of OMSVM was significantly greater than those of other algorithms. The McNemar test is run on matched pair data. Its null hypothesis is the equality of the discordant pairs in hit or non-hit frequencies of the two different prediction methods. The McNemar test requires the statistic that is distributed as χ^2 with one degree of freedom. This test is used with nominal data, and it is particularly useful for before-and-after measurement of the same subjects. The advantage of the test over the paired t-test is a lower Type-I error; the test also has good power[3] [59]. Table 10 shows the results of the McNemar test for comparing the performance of five algorithms on the test data. As shown in the table, OMSVM was better than MDA, CBR,

[3] The power of a statistical test represents the probability that it will reject a false null hypothesis. That is, a statistical test has good power denotes that it makes fewer Type II errors.

Table 8
Experimental results of OMANN.

Dataset	One-Against-The-Next						One-Against-Followers					
	Forward			Backward			Forward			Backward		
	Train (%)	Test (%)	Valid (%)	Train (%)	Test (%)	Valid (%)	Train (%)	Test (%)	Valid (%)	Train (%)	Test (%)	Valid (%)
1	68.55	67.05	66.67[a]	68.55	67.05	66.67[a]	68.42	66.28	65.89	68.42	66.28	66.28
2	69.32	64.34	63.18	69.32	64.34	63.18	68.42	65.12	65.12[a]	68.55	65.12	64.73
3	67.39	63.57	69.38[a]	67.27	63.57	69.38[a]	66.75	65.50	67.05	66.75	65.12	67.05
4	68.68	67.83	68.22[a]	68.55	67.83	68.22[a]	70.09	68.22	67.05	69.06	67.05	67.44
5	67.78	69.38	67.83[a]	67.65	69.38	67.44	64.70	69.38	66.67	64.83	69.38	66.67
Avg.	**68.34**	**66.43**	**67.05[a]**	**68.27**	**66.43**	**66.98**	**67.68**	**66.90**	**66.36**	**67.52**	**66.59**	**66.43**

[a] The best performance for each data set.

Table 9
Summarized results of the experiments.

Type	Technique	Train (%)	Test (%)	Valid (%)
Statistical/AI approaches	MDA	65.03		63.10
	MLOGIT	67.12		65.43
	CBR			51.40
	ANN	69.55	65.35	65.66
Conventional MSVMs	One-Against-All (OAA)	71.53		64.34
	One-Against-One (OAO)	75.78		67.21
	DAGSVM	70.84		67.29
	ECOC	70.18		64.50
	Weston and Watkins (WW)	71.61		67.05
	Crammer and Singer (CS)	71.90		64.89
Proposed method	OMSVM	75.77		67.98

Table 10
McNemar values for the validation data set.

	MLOGIT	CBR	ANN	OAA	OAO	DAGSVM	ECOC	WW	CS	OMSVM
MDA	4.163**	49.020***	5.044**	0.611	10.859***	11.903***	0.781	10.917***	1.238	15.315***
MLOGIT		73.469***	0.028	0.693	2.766*	4.069**	0.513	3.150*	0.131	5.198**
CBR			76.111***	65.766***	98.117***	96.111***	67.040***	96.193***	68.331***	109.587***
ANN				1.020	1.861	2.073	0.800	1.700	0.285	3.823*
OAA					5.918**	5.801**	0.010	5.279**	0.275	8.430***
OAO						0.000	5.479**	0.007	3.115*	0.880
DAGSVM							5.326**	0.030	3.321*	0.430
ECOC								4.433**	0.088	7.533***
WW									2.641	0.903
CS										5.394**

* Significant at the 10% level.
** Significant at the 5% level.
*** Significant at the 1% level.

One-Against-All (OAA), and ECOC at the 1% level of significance, and OMSVM performed better than MLOGIT and Crammer and Singer's method (CS) at the 5% significance level. However, OMSVM barely outperformed ANN at the 10% significance level, and it did not outperform One-Against-One (OAO), DAGSVM, and the method of Weston and Watkins (WW) at any level of statistical significance. The performance gap between OMSVM and DAGSVM is about 0.7% (=67.98%−67.29%), which implies that our proposed model just correctly classifies nine additional cases compared to DAGSVM. Although predicting nine more cases correctly can be important in real businesses, there was no statistical significant difference between the two methods. Nevertheless, we believe that our proposed model is valuable because it solves ordinal multi-class problems as accurately as comparative MSVM algorithms, such as DAGSVM and One-Against-One, but with fewer classifiers.

6. Conclusions and directions for future research

In this study, we proposed a novel MSVM algorithm that was optimized for credit rating. In contrast to prior studies that just applied conventional MSVMs to credit ratings, we suggested a new MSVM algorithm, called OMSVM, which is designed to use order-information in ordinal multi-class classification problems. To validate the applicability of the proposed algorithm, we applied it to a real case of bond rating. As a result, we found that OMSVM outperformed many kinds of MSVM approaches proposed in the literature as well as other AI techniques including MDA, MLOGIT, CBR, and ANN, for solving the ordinal multi-class classification problem. It is also impressive that our approach requires the fewest classifiers among the MSVM techniques that are designed to combine several binary classifiers. In addition, we provided the empirical outcome that the One-Against-Followers approach and the forward strategy of fusion might be more effective when applying OMSVM. As a result, we may conclude that OMSVM is an effective and efficient classifier for solving ordinal multi-class classification problems.

Although we applied our model to the domain of credit rating, OMSVM is applicable to any kind of ordinal multi-class classification problem. For example, in medical diagnostics, doctors may want to build a prediction model that classifies patients by the level of disease severity. In the business domain, some marketers may want to build classification models that classify customers by the level of profitability, so that they can implement a customer relationship management (CRM) strategy [60–65,71]. Besides, there are many kinds of application areas that require accurate ordinal multi-class classification models. Thus, we expect that our proposed model will be able to contribute to other domains or business problems in future studies.

One of the limitations of our study is that we did not provide empirical evidence to support our assertion that our proposed model is the most efficient among various multi-class SVM techniques. To do this, we should have measured and reported the computational time of each experimental model [41,66]. However, we could not report meaningful computational times since the experiments were not performed on the same software platform, because we used different software solutions for each experimental model. Thus, in future work we must conduct experiments using the same conditions and compare the computational time between the proposed model and other comparative models.

In addition, the proposed OMSVM model is designed to seek the optimal kernel function and the optimal parameters using grid-search. However, some recent studies proposed an evolutionary approach, such as genetic algorithms (GA), for tuning SVM parameters [67]. Thus, in the future, a new OMSVM model with an evolutionary process for optimizing the parameters of SVM classifiers can be investigated.

References

[1] Cao L, Guan LK, Jingqing Z. Bond rating using support vector machine. Intelligent Data Analysis 2006;10:285–96.
[2] Shin KS, Han IA. Case-based approach using inductive indexing for corporate bond rating. Decision Support Systems 2001;32:41–52.
[3] Huang Z, Chen H, Hsu CJ, Chen WH, Wu S. Credit rating analysis with support vector machines and neural networks: a market comparative study. Decision Support Systems 2004;37:543–58.
[4] Liang L. Earnings forecasts in enterprise information systems environment. Enterprise Information Systems 2008;2:1–19.
[5] Huang W, Nakamori Y, Wang SY. Forecasting stock market movement direction with support vector machine. Computers & Operations Research 2005;32:2513–22.
[6] Yang B, Li L, Xu J. An early warning system for loan risk assessment using artificial neural networks. Knowledge-Based Systems 2001;14:303–6.
[7] Zhu X, Wang H, Xu L, Li H. Predicting stock index increments by neural networks: the role of trading volume under different horizons. Expert Systems with Applications 2008;34:3043–54.
[8] Wong WT, Hsu SH. Application of SVM and ANN for image retrieval. European Journal of Operational Research 2006;173:938–50.
[9] Kumar PR, Ravi V. Bankruptcy prediction in banks and firms via statistical and intelligent techniques. European Journal of Operational Research 2007; 180:1–28.
[10] Yang Y. Adaptive credit scoring with kernel learning methods. European Journal of Operational Research 2007;183:1521–36.
[11] Kim HS, Sohn SY. Support vector machines for default prediction of SMEs based on technology. European Journal of Operational Research 2010;201: 838–46.
[12] Paleologo G, Elisseeff A, Antonini G. Subagging for credit scoring models. European Journal of Operational Research 2010;201:490–9.
[13] Vapnik V. The nature of statistical learning theory. New York: Springer-Verlag; 1995.
[14] Crammer K, Singer Y. On the learnability and design of output codes for multiclass problems. In: Proceedings of the 13th annual conference on computational learning theory, Palo Alto, California; 2000. p. 35–46.
[15] Platt JC, Cristianini N, Shawe-Taylor J. Large margin DAG's for multiclass classification. In: Solla SA, Leen TK, Muller K-R, editors. Advances in neural information processing systems, vol. 12. Cambridge, MA: MIT Press; 2000. p. 547–53.
[16] Hsu CW, Lin CJA. Comparison of methods for multiclass support vector machines. IEEE Transactions on Neural Networks 2002;13:415–25.
[17] Shuibo Z, Houjun T, Zhengzhi H, Haoran Z. Solving large-scale multiclass learning problems via an efficient support vector classifier. Journal of Systems Engineering and Electronics 2006;17:910–5.
[18] Navia-Vázquez A. Compact multi-class support vector machine. Neurocomputing 2007;71:400–5.
[19] Übeyli ED. Multiclass support vector machines for diagnosis of erythemato-squamous disease. Expert Systems with Applications 2008;35:1733–40.
[20] Wang A, Yuan W, Liu J, Yu Z, Li H. A novel pattern recognition algorithm: combining ART network with SVM to reconstruct a multi-class classifier. Computers & Mathematics with Applications 2009;57:1908–14.
[21] Oladunni OO, Trafalis T. A regularized pairwise multi-classification knowledge-based machine and applications. European Journal of Operational Research 2009;195:924–41.
[22] Chen WH, Shih JY. A study of Taiwan's issuer credit rating systems using support vector machines. Expert Systems with Applications 2006;30: 427–35.
[23] Lee YC. Application of support vector machines to corporate credit rating prediction. Expert Systems with Applications 2007;33:67–74.
[24] Kwon YS, Han I, Lee KC. Ordinal pairwise partitioning (OPP) approach to neural networks training in bond rating. Intelligent Systems in Accounting Finance and Management 1997;6:23–40.
[25] Buta P. Mining for financial knowledge with CBR. AI Expert 1994;9: 34–41.
[26] Xu L. Case-based reasoning—a major paradigm of artificial intelligence. IEEE Potentials 1995;13:10–3.
[27] Xu L. Case-based reasoning for AIDS initial assessment. Knowledge-Based Systems 1995;8:32–8.
[28] Li H, Li L. Representing diverse mathematical problems using neural networks in hybrid intelligent systems. Expert Systems 1999;16:262–72.
[29] Zhou S, Xu L. Dynamic recurrent neural networks for a hybrid intelligent decision support system for the metallurgical industry. Expert Systems 1999;16:240–7.
[30] Zhou S, Xu LA. Neural network representation of linear programming. European Journal of Operational Research 2000;124:224–34.
[31] Zhou S, Xu LA. New type of recurrent fuzzy neural network for modeling dynamic systems. Knowledge-Based Systems 2001;4:243–51.
[32] Li H, Li L, Wang J. Interpolation representation of feedforward neural networks. Mathematical and Computer Modeling 2003;37:829–47.
[33] Kakousis K, Paspallis N, Papadopoulos G. A survey of software adaptation in mobile and ubiquitous computing. Enterprise Information Systems 2010;4: 355–89.
[34] Vapnik V. Statistical learning theory. New York: Wiley; 1998.
[35] Kim K. Financial time series forecasting using support vector machines. Neurocomputing 2003;55:307–19.
[36] Huang CL, Wang CJA. GA-based feature selection and parameters optimization for support vector machines. Expert Systems with Applications 2006;31: 231–40.
[37] Herbrich R. Learning kernel classifiers: theory and algorithms. Massachusetts: The MIT Press; 2002.
[38] Cristianini N, Shawe-Taylor J. An introduction to support vector machines. Cambridge England: Cambridge University Press; 2000.
[39] Moguerza JM, Muñoz A. Support vector machines with applications. Statistical Science 2006;21:322–36.
[40] Lorena AC, de Carvalho ACPLF. Investigation of strategies for the generation of multiclass support vector machines. In: Nguyen NT, Katarzyniak R, editors. New Challenges in Applied Intelligence Techniques. Berlin, Germany: Springer-Verlag; 2008. p. 319–28.
[41] Wu YC, Lee YS, Yang JC. Robust and efficient multiclass SVM models for phrase pattern recognition. Pattern Recognition 2008;41:2874–89.
[42] Kreβel U. Pairwise classification and support vector machines. In: Scholkopf B, Burges C, Smola AJ, editors. Advances in kernal methods: support vector learning. Cambridge MA: MIT Press; 1999. p. 255–68. [Chapter 15].
[43] Statnikov A, Aliferis CF, Tsamardinos I, Hardin D, Levy SA. Comprehensive evaluation of multicategory classification methods for microarray gene expression cancer diagnosis. Bioinformatics 2005;21:631–43.
[44] Friedman J. Another approach to polychtomous classfication. Technical Report, Stanford University; 1996.
[45] Dietterrich TG, Bakiri G. Solving multiclass learning problems via error-correcting output codes. Journal of Artificial Intelligence Research 1995;2: 263–86.
[46] Klautau A, Jevtic N, Orlitsky A. On nearest-neighbor error-correcting output codes with application to all-pairs multiclass support vector machines. Journal of Machine Learning Research 2003;4:1–15.
[47] Weston J, Watkins C. Support vector machines for multiclass pattern recognition. In: Proceedings of the seventh European symposium on artificial neural networks, Bruges, Belgium; 1999. p. 219–24.
[48] Hsu CW, Lin CJA. Simple decomposition method for support vector machines. Machine Learning 2002;46:291–314.
[49] Pinches GE, Mingo KAA. Multivariate analysis of industrial bond ratings. The Journal of Finance 1973;28:1–18.

[50] Belkaoui A. Industrial bond ratings: a new look. Financial Management 1980;9:44–51.
[51] Ederington LH. Classification models and bond ratings. The Financial Review 1985;20:237–62.
[52] Kim JW. Expert systems for bond rating: a comparative analysis of statistical rule-based and neural network systems. Expert Systems 1993;10:167–71.
[53] Moody J, Utans J. Architecture selection strategies for neural networks application to corporate bond rating. In: Refenes A, editor. Neural networks in the capital markets. Chichester: Wiley; 1995. p. 277–300.
[54] Chaveesuk R, Srivaree-Ratana C, Smith AE. Alternative neural network approaches to corporate bond rating. Journal of Engineering Valuation and Cost Analysis 1999;2:117–31.
[55] Han J, Kamber M. Data mining: concepts and techniques.2nd edition San Francisco: Morgan Kaufmann Publishers; 2001.
[56] Tay FEH, Cao LJ. Application of support vector machines in financial time series forecasting. Omega 2001;29:309–17.
[57] Chang CC, Lin CJ. LIBSVM: a library for support vector machines; 2001. Software available at; 〈http://www.csie.ntu.edu.tw/~cjlin/libsvm/〉.
[58] Hsu CW, Lin CJ. BSVM: a SVM library for the solution of large classification and regression problems; 2006. Software available at; 〈http://www.csie.ntu.edu.tw/~cjlin/bsvm/〉.
[59] Dietterrich TG. Approximate statistical tests for comparing supervised classification learning algorithms. Neural Computation 1998;10:1895–923.
[60] Chen Y, Li L. Deriving information from CRM for knowledge management—a note on a commercial bank. Systems Research and Behavioral Science 2006;23:141–6.
[61] Betheshti H, Hultman M, Jung M, Opoku R, Salehi-Sangari E. Electronic supply chain management applications by Swedish SMEs. Enterprise Information Systems 2007;1:255–68.
[62] Li H, Wang H. A multi-agent-based model for a negotiation support system in electronic commerce. Enterprise Information Systems 2007;1:457–72.
[63] Hou H, Xu S, Wang H. A study on X party material flow: the theory and applications. Enterprise Information Systems 2007;1:287–99.
[64] Wang S, Archer N. Electronic marketplace definition and classification: literature review and clarification. Enterprise Information Systems 2007;1:89–112.
[65] Millet P, Schmitt P, Botta-Genoulaz V. The SCOR model for the alignment of business processes and information systems. Enterprise Information Systems 2009;3:393–407.
[66] Lorena AC, de Carvalho ACPLF. Comparing techniques for multiclass classification using binary SVM predictors. Lecture Notes in Artificial Intelligence 2004;2972:272–81.
[67] Ahn H, Lee K, Kim Kj. Global optimization of support vector machines using genetic algorithms for bankruptcy prediction. Lecture Notes in Computer Science 2006;4234:420–9.
[68] Kim HJ, Shin KS. A hybrid approach using case-based reasoning and fuzzy logic for corporate bond rating. Journal of Intelligence and Information Systems 2004;10:91–109.
[69] Lee IH, Shin KS. A study on forecasting accuracy improvement of case based reasoning approach using fuzzy relation. Journal of Intelligence and Information Systems 2010;16:67–84.
[70] Ahn H, Kim KJ, Han I. Purchase Prediction model using the support vector machine. Journal of Intelligence and Information Systems 2005;11:69–81.
[71] Lee HY, Yang JH, Ryu CH. A model for effective customer classification using LTV and churn probability: application of holistic profit method. Journal of Intelligence and Information Systems 2006;12:109–26.