Optimizing SVM Ensembles Using Genetic Algorithms in Bankruptcy Prediction

Myoung-Jong Kim, Hong-Bae Kim and Dae-Ki Kang, *Member, KIMICS*

Abstract— **Ensemble learning is a method for improving the performance of classification and prediction algorithms. However, its performance can be degraded due to multicollinearity problem where multiple classifiers of an ensemble are highly correlated with. This paper proposes genetic algorithm-based optimization techniques of SVM ensemble to solve multicollinearity problem. Empirical results with bankruptcy prediction on Korea firms indicate that the proposed optimization techniques can improve the performance of SVM ensemble.**

Index Terms— **Bankruptcy Prediction, Coverage Optimization, Ensemble, Genetic Algorithm, Support Vector Machines.**

I. INTRODUCTION

Since bankruptcy has a significant impact on management, stockholders, employees, customers and nation, bankruptcy prediction has been an important and widely studied topic in accounting and finance.

The traditional techniques used in bankruptcy prediction are statistical techniques. Numerous statistical techniques have been applied to developing bankruptcy prediction models. Many empirical studies have proposed statistical bankruptcy prediction models using multiple regression, discriminant analysis, logistic models, and probit [1]-[7]. However, strict assumptions of traditional statistics, such as the linearity, normality, independence among predictor variables and pre-existing functional form relating the criterion variable and the predictor variable, have limited their application to the real world.

Another technique is machine learning including decision trees (DT), case-based reasoning, neural network (NN), and Support Vector Machine (SVM) [8]-[16].

Recent technique applied in bankruptcy prediction is ensemble learning [17]-[19]. Ensemble learning is a method for improving the performance of classification and prediction algorithms. It is a method for finding a highly accurate classifier on the training set by constructing and combining an ensemble of weak classifiers, each of which needs only to be moderately accurate on the training set [20],[21]. Ensemble learning has received considerable attention from machine learning and artificial intelligence fields because of its remarkable performance improvement and flexible integration with the traditional learning algorithms such as DT, NN, and SVM, etc.

In those researches, all of DT ensemble studies have demonstrated impressive improvements in the generalization behavior of DT, while NN and SVM ensemble studies have not shown remarkable performance as shown in DT ensembles [22]-[25]. Recently, several works have reported that the performance of ensemble can be degraded where multiple classifiers of an ensemble are highly correlated with, and thereby result in multicollinearity problem, which leads to performance degradation of the ensemble. They have also proposed the differentiated learning strategies to cope with performance degradation problem [19],[22],[26].

This paper proposes a hybrid system for coverage optimization of SVM ensemble (CO-SVM) in order to improve the performance of SVM ensemble. Genetic Algorithm (GA), which has been widely used for various optimization problems, is used to deal with the coverage optimization problem. The proposed CO-SVM is applied to bankruptcy prediction task to validate its performance.

Experimental results with the bankruptcy prediction on Korean firms indicate that CO-SVM system has shown significant performance improvement of SVM ensemble by solving multicollinearity problem.

The remainder of this paper is organized as follows: The next section describes two popular ensemble algorithms Bagging and AdaBoost. Section 3 presents the previous researches on DT and SVM ensemble. Section 4 explains the algorithm of the proposed hybrid system. Section 5 presents data descriptions and experimental design process. Section 6 discusses the results of the experiment. The final section presents several concluding remarks and future research issues.

II. ENSEMBLE LEARNING

Several ensemble methods of constructing and combining an ensemble of classifiers have been proposed. Two methods which have been widely used are Bagging [26] and AdaBoost [21],[27].

A. Bagging algorithm

Bagging is a bootstrap aggregation method that creates and combines multiple classifiers, each of which is trained

Manuscript received July 16, 2010; revised July 17, 2010; accepted July 28, 2010.

Corresponding Author, Hong-Bae Kim is with the Division of Business, Dongseo University, Pusan, 617-716, Korea (Tel : +82-51-320-1913, Email: rfctogether@gdsu.dongseo.ac.kr)

on a bootstrap replicate of the original training set. The bootstrap data is created by resampling examples uniformly with replacement from the original training set. Each classifier is created by training on corresponding bootstrap replicate. The classifiers could be trained in parallel and the final classifier C is generated by combining ensemble of classifier.

Bagging has been considered as a variance reduction technique for a given base procedure such as decision trees. Bagging is known to be particularly effective when the classifiers are unstable, that is, when perturbing the learning set can cause significant changes in the classification behavior because Bagging improves generalization performance due to a reduction in variance while maintaining or only slightly increasing bias.

B. Boosting algorithm

Boosting constructs a composite classifier by sequentially training classifiers while increasing weight on the misclassified observations through iterations. The observations that are incorrectly predicted by previous classifiers are chosen more often than examples that were correctly predicted. Thus Boosting attempts to produce new classifiers that are better able to predict examples for which the current ensemble's performance is poor. Boosting combines predictions of ensemble of classifiers with weighted majority voting by giving more weights on more accurate predictions.

AdaBoost is one of the most widely used Boosting methods. The algorithm of AdaBoost can be described as follows. Let $\{(x_1, y_1),(x_2, y_2)\ldots,(x_n, y_n)\}$ be a training set, where x is a vector of predictor variables and y is a two-class response variable such that $y \in \{-1, 1\}$. The weight $w_b(i)$ is assigned to each observation x_i and is initially set to $1/n$. The bth classifier, C_b, is learned on this new training set, T_b and applied to each training observation. The error of this classifier, ε_b, is calculated as

$$\varepsilon_b = \sum_{i=1}^{n} w_b(i)\varepsilon_b(i) \text{ where } \varepsilon_b = \begin{cases} 0 & C_b(i) = y_i \\ 1 & C_b(i) \neq y_i \end{cases} \cdots(1)$$

and ε_b is slightly better than random guessing. α_b, which indicates the importance of C_b, is defined as $\alpha_b=\ln((1-\varepsilon_b)/\varepsilon_b)$. The weight for the b+1th classifier is calculated as $w_{b+1}(i)=w_b(i)\exp(\alpha_b\varepsilon_b(i))$ and the calculate weights are normalized to sum one. Consequently, the weight of the incorrectly classified observation is increased, and the weight of correctly classified is decreased. Thus each classifier is forced to concentrate on the training examples that are misclassified by the previous classifier. AdaBoost is repeatedly apply to the training set with modified weights, producing a sequence of classifiers C_b, where b =1, 2, …, B. Finally, the ensemble classifier calculates the final predicted output as the weighted sum of its votes as

$$C(x) = sign(\textstyle\sum_{b=1}^{B}\alpha_b C_b(x)). \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots(2)$$

III. DT ENSEMBLE AND SVM ENSEMBLE

All of empirical studies using DT ensemble have been shown that DT ensemble can enhance the prediction performance of unstable learning algorithms such as Classification and Regression Tree (CART) and C4.5 [28]-[31]. Recently, several studies on bankruptcy prediction have applied AdaBoost to bankruptcy classification trees. They have shown that AdaBoost decreases the generalization error and improve the accuracy [17]. An empirical comparison has shown that AdaBoost with classification tree decreases the generalization error by about 30% percent with respect to error produced with NNs [18].

On the other hand, theoretical background of SVM ensemble learning is explored by Valentini and Ditterich [32]. They have introduced a theoretical foundation of SVM ensemble by showing that SVM ensemble effectively reduces the bias and the variance of SVMs through bias-variance decomposition and have proposed a variety of ensemble methods such as bagged ensemble SVMs of selected low-bias SVMs and heterogeneous ensembles of SVMs. Valentini [33] has proposed a quantitative model of bias-variance decomposition of SVM ensemble in terms of its error rate. SVM ensemble is shown to be able to enable a single classifier to provide stable classification performance by cross-validation optimization of the classifier [34]. Kim et al. [35] have proposed ensemble construction techniques such as weighted average methods and hierarchical combining methods based on bagging algorithms, AdaBoost algorithms, majority vote, and least squares estimators (LSE). They have shown that SVM ensemble is more accurate than a single SVM classifier based on the statistically significant experimental results on University of California-Irvine (UCI) data sets and cellular phone fraud detection data sets. Besides those theoretical researches, there have been numerous experimental researches explored on diverse fields. Those fields include various natural science areas such as prediction of bacterial transcription start sites, voice recognition, fault diagnosis of roller bearings, cardiovascular disease level prediction, and business and financial areas such as credit scoring systems [24],[36]-[39].

Most of the studies on SVM ensemble have reported that ensemble learning can improve individual SVM’s accuracy. However, some studies have indicated that SVM ensemble have shown the improved results in a few limited areas and that NN ensemble’s performance is often even worse than that of a single classifier, compared with the promising results of DT ensemble [22]-[25]. Recently, several works have investigated the cause of performance degradation and differentiated learning strategy for performance improvement [26],[40]. They have reported that the performance of ensemble can be degraded where multiple classifiers of an ensemble are highly correlated with, and thereby result in multicollinearity problem, which leads to performance degradation of the ensemble.

Hansen and Salamon [40] insisted that it is necessary and sufficient for the performance enhancement of an ensemble that the ensemble should contain diverse classifiers and each classifier in the ensemble needs to be more accurate than random guess. This means that the accuracy of each classifier in the ensemble should be over 50%, and the classifiers in the ensemble should be diverse to minimize miss-classification rate. For example, in the ensemble with diverse classifiers, if one classifier can predict wrong but other classifiers can predict right, then the final classification result can be right. However, when the classifiers in the ensemble lack diversity, it is highly possible that most classifiers predict similarly on one example.

Breiman [26] explored that bagging (and to a lesser extent, boosting) can increase the performance of unstable learning algorithms, but does not show remarkable performance improvement on stable learning algorithms. Ensemble learning applies various sampling techniques such as bagging and boosting, etc to guarantee the diversity in a classifier pool. Unstable learning algorithms such as decision tree learners are sensitive to the change of the training data, and thus small changes in the training data can yield large changes in the generated classifiers. Therefore, ensemble with unstable learning algorithms can guarantee some diversity among the classifiers. To the contrary, stable learning algorithms such as SVM generate similar classifiers in spite of small changes of the training data, and thus the correlation among the resulting classifiers is very high. This high correlation results in multicollinearity problem, which leads to performance degradation of the ensemble.

Kim's work [19] shows the performance comparison in bankruptcy prediction on Korea firms using tradition prediction algorithms such as NN, DT, and SVM. It reports that stable learning algorithms such as NN and SVM have higher predictability than the unstable DT. Meanwhile, with respect to their ensemble learning, DT ensemble shows the more improved performance than NN and SVM ensemble. Further analysis with variance inflation factor (VIF) analysis empirically proves that performance degradation of ensemble is due to multicollinearity problem. It also proposes that optimization of ensemble is needed to cope with such a problem.

IV. COVERAGE OPTIMIZATION

Ho [41] has defined the classifier selection problem as coverage optimization. Coverage optimization is a problem of choosing a sub-ensemble of d ($K \geq d$) classifiers from an ensemble of K classifiers so that the chosen sub-ensemble has the maximum classification performance. For this problem, the size of the possible search space is ${}_KC_d$, which means the search space is an exponential search space. Recently, GA are popularly used as a effective tool to solve such a local search operations because GA can prevent local optima by using cross-over and mutation operators, and can search rapidly a vast and complicated search space to find an optimal or near-optimal solution using probabilistic search methods.

Zhou et al. [42] have formally proved that a sub-ensemble of selected classifiers can be superior to an ensemble composed of all the classifiers in terms of performance. They also have experimentally demonstrated their proposal by generating a neural network ensemble and adapting GA for choosing an optimal sub-ensemble. Oliverira et al. [43] have used GA to select a sub-ensemble from ensembles with multiple classifiers. Those previous works have shown the promising results in selecting sub-ensemble which can improve prediction accuracy, however, they have not proposed the process for selecting diverse classifiers.

Unlike previous works, the proposed CO-SVM has advantage to deal with accuracy and diversity together. The GA learning process for coverage optimization is performed through three stages as followings.

A. Population Setup

A solution can be encoded to chromosome form in order to solve the coverage optimization problem. We set the size of search space for the chromosomes of the coverage optimization as the number of classifiers in the ensemble, K, and assign the weight (d_k) of each classifier as either 0 or 1, where 0 means the classifier is excluded and 1 means the classifier is selected. Thus, the GA chromosomes for the coverage optimization are encoded into binary strings. For example, when the binary number is 1100100011, the number of classifiers in the ensemble is 10 since the binary number has 1 digit and the classifiers #1, #2, #5, #9, and #10 are chosen as a sub-ensemble.

B. Fitness Function

Majority voting is used to combine the results of multiple classifiers and generate the final class of each observation, x_i. In case of a tie, x_i belongs to a class of a classifier with the highest prediction accuracy on training set. Fitness function is defined as average prediction accuracy because the purpose of this paper is to find bankruptcy prediction model with high prediction performance.

However, with only GA search, it is not sufficient to solve the multicollinearity problem owing to the correlation among the classifiers; we add more constraints with respect to the measuring multicollinearity. VIF is one of the generally used methods to measure multicollinearity. VIF of the *k*th classifier is calculated as following:

$$\mathrm{VIF}(k) = \frac{1}{1 - R_k^2} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots(3)$$

Here, R_K^2 is a value of the coefficient of determination obtained when we performed regression analysis on the other classifiers with the *k*th classifier as a response parameter. If the *k*th classifier is closely related with other classifiers, then R_K^2 will be close to 1, and therefore VIF(*k*) will increase. If 5< VIF(*k*)<10 then it is possible that the *k*th classifier has multicollinearity, and if VIF(*k*)>10 then we can evaluate the *k*th classifier has serious multicollinearity. Thus, the classifiers included in the sub-ensemble should have VIF less than 5. Therefore, the fitness function including the constraints can be expressed as following:

$$\mathrm{Fitness} = \frac{1}{n}\sum CR_i \quad \text{where}\, CR_i = \begin{cases} 0 & C_b(i) = y_i \\ 1 & C_b(i) \neq y_i \end{cases} \cdots(4)$$
$$\text{Subject to VIF}(k) < 5$$

C. Genetic Operations

At this stage, GA generates new candidates from the initial chromosome set using the various search methods including crossover, mutation, etc. The algorithm calculates a fitness score for each candidate and replaces the organisms with low scores by the organisms with high scores. This process is repeated until the stopping conditions are satisfied.

IV. EXPERIMENTAL DESIGN

A. Data description

The data is obtained from a commercial bank in Korea. The data set contains 1,200 externally audited manufacturing firms, half of which went bankrupt during 2002-2005 while healthy firms were selected from active companies at the end of 2005.

Initially 31 financial ratios categorized as profitability, debt coverage, leverage, capital structure, liquidity, activity, and size are investigated through literature review and basic statistical methods. We have chosen final input variables by Receiver operating characteristic (ROC) curve analysis. In ROC curve analysis, we assess the performance of a classifier based on ROC curve, in which 1 – specificity and sensitivity of the classifier are graphically plotted. Sensitivity and specificity is measured as TP/(TP+FN) and TN/(FP+TN) shown in Table 1.

TABLE I
CONFUSION MATRIX

		Predicted Class	
		Bankrupt (Positive)	Non-Bankrupt (Negative)
Actual Class	Bankrupt (Positive)	True Positive (TP)	False Negative (FN)
	Non-Bankrupt (Negative)	False Positive (FP)	True Negative (TN)

The performance criterion of ROC curve is calculated an area under the ROC curve (AUROC), which is the probability that a classifier will rank a randomly chosen positive instance higher than a randomly chosen negative one and equals to. AUROC of the perfect model is 1, and AUROC of a random guess model is 0.5. Generally, a classifier has an AUROC value between 0.5 and 1, and is considered accurate if AUROC is close to 1. We have chosen seven financial ratios with the highest AUROC values from the 31 financial ratios, presented in Table 2.

TABLE II
AUROC

Variables		AUROC
Profitability	Ordinary income to total assets*	52.5
	Net income to total assets	45.9
	Financial expenses to sales	49.7
	Financial expenses to total debt	48.9
	Net financing cost to sales	50.8
	Ordinary income to sales	45.9
	Net income to sales	49.9
	Ordinary income to capital	48.8
	Net income to capital	48.1
Debt coverage	EBITDA to Interest expenses*	53.7
	EBIT to Interest expenses	40.1
	Cash operating income to interest expenses	48.9
	Cash operating income to total debt	48.8
	Cash flow after interest payment to total debt	52.3
	Cash flow after interest payment to total debt	53.1
	Debt repayment coefficient	51.7
	Borrowings to Interest expenses	53.4
Leverage	Total debt to total assets*	51.9
	Current assets to total assets	50.9
Capital structure	Retained earning to total assets*	53.5
	Retained earning to total debt	52.7
	Retained earning to current assets	51.1
Liquidity	Cash ratio*	46.5
	Quick ratio	45.5
	Current assets/current Liabilities	43.2
Activity	Inventory to sales*	30.8
	Current liabilities to sales	29.2
	Account receivable to sales	27.7
Size	Total assets*	24.8
	Sales	22.4
	Fixed assets	22.6

Although less directly related to model predictive power, the potential presence of multicollinearity is an important check-point of the model. Variance information factors (VIF) among 7 financial ratios are performed to check for multicollinearity. Table 3 shows that the estimated VIF values are below the threshold levels of 4 and 10 that are commonly used in VIF analysis when testing for presence of multicollinearity. The findings indicate that the model variables do not present any substantial multicollinearity.

TABLE III
VARIANCE INFLATION FACTORS

Variables	VIF
Ordinary income to total assets	1.64
EBITDA to Interest expenses	2.34
Total debt to total assets	1.95
Retained earning to total assets	2.77
Cash ratio	1.54
Inventory to sales	1.73
Total assets	1.67

B. Experimental design

In this paper, we define the bankruptcy problem as a non-linear problem and use the radial basis function (RBF) kernel as a kernel function to optimize the hyperplane. There are two parameters while using RBF kernels: acceptable error *C* and kernel parameter δ^2. We make up various topologies, varying *C* from 1 to 250 and δ^2 from 1 to 200. On the other hand, *C*4.5 is used as a base classifier for creating DT ensemble.

We have limited the maximal number of ensemble creations to 25, following the research results by Opitz and Maclin [44] that error rate reduction is marginal when the number of ensemble creations is over 25.

The crossover and mutation rates are changed to prevent the output from falling into the local optima. The crossover rate ranges 0.5–0.7 and the mutation rate ranges 0.06–0.1 for this experiment. The stop condition is set to 1000 iterations and thus genetic search repeated by 50 generation. We use Microsoft Excel and the GAs software package called Evolver.

V. EXPERIMENTAL RESULTS

We repeated 10-fold cross validations for five times with different random seeds in order to ensure that the comparison among three different classifiers does not happen by chance [24]. For each of 10-fold cross validation, the entire data set (1200 firms) is first partitioned into 10 equal-sized sets, and then each set is in turn used as the test set while the classifier trains on the other nine sets. That is, cross-validated folds were tested independently of each algorithm. This way we obtained the results for three classifiers on each of the 50 experiments.

VIF analysis is performed to investigate the effect of multicollinearity on performance of three classifiers. The size of classifiers and the measure of multicollinearity are shown in Table 4. DT ensemble do not present any substantial multicollinearity, while there exists substantial multicollinearity problem in 3 classifiers of bagged SVM and 4 classifiers of boosted SVM. This result indicates that the performance improvement of SVM ensemble can be deteriorated due to multicollinearity problem. On the other hand, boosted CO-SVM has 12 classifiers and bagged CO-SVM has 15 classifiers. Also, the measurements of VIF which reflects multicollinearity shows that there is no significant multicollinearity present among the classifiers. Since multicollinearity is one of the main criteria about the diversity of classifiers, those results indicate that CO-SVM consists of diverse classifiers.

TABLE IV
MULTICOLLINEARITY BY VIF ANALYSIS

No. of Classifiers	DT		SVM		CO-SVM	
	Boosted	Bagged	Boosted	Bagged	Boosted	Bagged
Total	21	25	17	25	12	15
VIF>5	-	-	3	4		

Table 5 describes the average prediction accuracy of each classifier. All ensemble classifiers show higher performance than a single classifier. Duncan test is used to examine whether the average performance of single classifier and two ensemble classifiers for 50 folds is significantly different. The results indicate that DT ensemble outperforms a single DT classifier at the 1% statistical significance level. This result mean ensemble with classification tree decreases the generalization error. Meanwhile, the performance of SVM ensemble is not significantly different from that of a single SVM classifier. This means that the combination of SVM and ensemble techniques do not have substantial impact on SVM in terms of performance improvement, and differentiated learning strategy is needed for performance improvement of SVM ensemble.

As shown in Table 5, SVM ensembles have shown marginal improvements over SVM in validation data with about 0.41%~0.63%, but the improvements of CO-SVM are as high as about 5.09%~5.39%. Compared with SVM ensembles, CO-SVM ensembles have fewer classifiers as shown in Table 5, but their accuracies are higher by 4.46% for boosting ensemble and 4.98% for bagging ensemble, which indicates performance improvement.

TABLE V
PREDICTION ACCURACY IN VALIDATION SET(%)

Classifier	Single	Boosted	Bagged
DT	70.30	75.09*	75.79*
SVM	72.44	73.07	72.85
CO-SVM	-	77.53	77.83

* indicates that the prediction accuracies between a single classifier and ensemble are significantly different at 1% significance level

To investigate if one algorithm shows significant improvement in terms of performance over another, we have conducted t-test verification on the 50 cross-validated classification results. As shown in Table 6, SVM ensemble learning has not shown significant performance improvement over a single SVM classifier when there is high correlation among the classifiers in the ensemble. To the contrary, bagging ensemble and boosting ensemble in which the highly correlated classifiers are removed by CO-SVM have shown significant performance improvements over a single SVM classifier and SVM ensemble by 1% significance level.

TABLE VI
T-TEST RESULTS ON VALIDATION SET

Single		Boosted			Bagged		
	SVM		SVM	CO-SVM		SVM	CO-SVM
DT	2.058**	DT	-1.942**	2.268**	DT	-2.827*	2.012**
		SVM		3.854*	SVM		3.945*

* 1% significance level, ** 5% significance level

From those experiments, we conclude that CO-SVM can solve the problem of the lack in diversity among the classifiers and thereby it shows more significant performance enhancement than SVM ensemble. Therefore, CO-SVM proposed in this research can be effective in stable performance enhancement of SVM and SVM ensemble through the classifier selection by considering correlation.

VI. CONCLUDING REMARKS

In this work, we have proposed CO-SVM to solve coverage optimization to enhance the stable performance enhancement of SVM ensemble. CO-SVM uses GA to guarantee the diversity of classifiers in coverage optimization process.

We use VIF analysis to measure multicollinearity as a degree of diversity in order to select various classifiers, which is the goal of coverage optimization, and we use VIF as a constraint of GA search to remove high correlation among the classifiers to insure the diversity of classifiers.

Experiments on company failure prediction to verify the performance of CO-SVM have shown that CO-SVM is effectively applied in the stable performance enhancement of SVM ensembles through the choice of classifiers by considering the correlations of the ensemble.

However, we expect to continue the following further researches to solve the limitations of this research. First of all, we mainly have focused on resolving coverage optimization. But, one of the most fundamental problems in ensemble is decision optimization process to find optimal combination function. We expect to have more advanced further researches on decision optimization process in the future.

Secondly, ensemble learning has a problem with noise. The noises in the learning samples distort the classification boundary of learning algorithms like SVM and degrade the learning performance. Especially, in boosting ensemble which focuses on the learning of miss-classified observations, the noise will repeatedly affect newly generated classifiers. To deal with those outliers, various SVM ensemble methods such as Probabilistic Roulette Selection, Karush–Kuhn–Tucker Condition-based Heuristic Selection, Automatic Feature Selection, etc. have been proposed [45]. Coupled with those researches, we expect to have more advanced further researches in the future.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea Grant funded by the Korean Government (NRF-2009-332-B00103).

REFERENCES

[1] Meyer, P. A., & Pifer, H. (1970). Prediction of bank failures. The Journal of Finance, 25, 853–868.

[2] Altman, E. L. (1968). Financial ratios, discriminant analysis and the prediction of corporate bankruptcy. The Journal of Finance, 23(4), 589–609.

[3] Altman, E. L., Edward, I., Haldeman, R., & Narayanan, P. (1977). A new model to identify bankruptcy risk of corporations. Journal of Banking and Finance, 1, 29–54.

[4] Dimitras, A. I., Zanakis, S. H., & Zopounidis, C. (1996). A survey of business failure with an emphasis on prediction methods and industrial applications. European Journal of Operational Research, 90(3), 487–513.

[5] Ohlson, J. (1980). Financial ratios and the probabilistic prediction of bankruptcy. Journal of Accounting Research, 18(1), 109–131.

[6] Pantalone, C., & Platt, M. B. (1987). Predicting commercial bank failure since deregulation. New England Economic Review, 37–47.

[7] Zmijewski, M. E. (1984). Methodological issues related to the estimation of financial distress prediction models. Journal of Accounting Research, 22(1), 59-82.

[8] Han, I., Chandler, J. S., & Liang, T. P. (1996). The impact of measurement scale and correlation structure on classification performance of inductive learning and statistical methods. Expert System with Applications, 10(2), 209–221.

[9] Shaw, M., & Gentry, J. (1998). Using and expert system with inductive learning to evaluate business loans. Financial Management, 17(3), 45–56.

[10] Bryant, S. M. (1997). A case-based reasoning approach to bankruptcy prediction modeling. International Journal of Intelligent Systems in Accounting, Finance and Management, 6(3), 195–214

[11] Buta, P. (1994). Mining for financial knowledge with CBR. AI Expert, 9(10), 34–41.

[12] Laitinen, T., & Kankaanpaa, M. (1999) Comparative analysis of failure prediction methods: the Finish case. European Accounting Review, 8(1), 67–92.

[13] Odom, M., & Sharda, R. (1990). A neural network for bankruptcy prediction, Proceedings of the International Joint Conference on Neural Networks, IEEE Press, San Diego, CA.

[14] Ravi, P., & Ravi, K. V. (2007). Bankruptcy prediction in banks and firms via statistical and intelligent techniques-a review. European Journal of Operational Research, 180, 1–28.

[15] Min, S. H.; Lee, J. M., & Han, I. G. (2006). Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert Systems with Applications, 31, 652-660.

[16] Shin, K., Lee, T., & Kim, H. (2005). An application of support vector machines in bankruptcy prediction. Expert Systems with Applications, 28, 127-135.

[17] Alfaro, E., Gámez, M., & García, N. (2007). Multiclass corporate failure prediction by AdaBoost.M1. Advanced Economic Research, 13, 301-312.

[18] Alfaro, E., García, N., Gámez, M., & Elizondo, D. (2008). Bankruptcy forecasting: an empirical comparison of AdaBooost and neural networks. Decision Support Systems, 45, 110-122.

[19] Kim, M. J., (2009). A Performance Comparison of Ensembles in Bankruptcy Prediction, Entrue Journal of Information Technology, 8(2), 41-49.

[20] Perrone, M. E. (1994). Putting it all together: Methods for combining neural networks. In J. D. Cowan, G. Tesauro, & J. Alspector, (Eds.), Advances in Neural Information Processing Systems, 6, (pp. 1188-1189). San Mateo, CA: Morgan Kaufmann.

[21] Schapire, R. E. (1990). The strength of weak learnability. Machine Learning, 5(2), 197-227.

[22] Buciu, I., Kotrooulos, C., & Pitas, I. (2001). Combining support vector machines for accuracy face detection, Proc. ICIP, 1054-1057.

[23] Dong, Y. S., & Han, K. S. (2004). A comparison of several ensemble methods for text categorization, IEEE International Conference on Service Computing.

[24] Eom, J. H., Kim, S. C. & Zhang, B. T. (in press). AdaaCDSS-E: A classifier ensemble based clinical decision support systems for cardiovascular disease level prediction, Expert Systems with Applications.

[25] Valentini, G., Muselli, M. & Ruffino, F. (2003). Bagged ensembles of SVMs or gene expression data analysis, The IEEE-INNS-ENNS International Joint Conference on Neural Networks, 1844-1849

[26] Breiman, L. (1994). Bagging predictors. Machine Learning, 24(2), 123-140.

[27] Freund, Y. (1995). Boosting a weak learning algorithm by majority. Information and Computation, 121(2), 256-285.

[28] Banfield, R. E., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2007). A comparison of decision tree ensemble creation techniques. IEEE Transactions on Pattern Analysis and Machine Intelligence, 29(1), 173-180.

[29] Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine Learning, 36, 105-139.

[30] Drucker, H., & Cortes, C. (1996). Boosting decision trees, Advanced Neural Information Processing Systems, 8.

[31] Quinlan, J. R. (1996). Bagging, boosting and C4.5. Machine Learning: Proceedings of the Fourteenth International Conference , 725-730).

[32] Valentini, G., & Dietterich, T. (2004). Bias-variance analysis of support vector machines for the development of SVM-based ensemble methods, Journal of Machine Learning Research, 5, 725-775.

[33] Valentini, G. (2005). An experimental bias-variance analysis of SVM ensembles based on resampling techniques. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, 35(6), 1252-1271.

[34] Pang, S., Kim, D., & Sung, Y. (2003). Membership authentication in the dynamic group by face classification using SVM ensemble, Pattern Recognition Letter, 24, 215-225.

[35] Pang, S., Kim, D., & Sung, Y. (2003). Membership authentication in the dynamic group by face classification using SVM ensemble, Pattern Recognition Letter, 24, 215-225.

[36] Gordon, J. J., Towsey, M. W., Hogan, J. M., Mathews S. A., & Timms, P. (2006). Improved prediction of bacterial transcription start sites, Bioinformatics, 22(2) 142-148

[37] Hu, Q., He, Z., Zhang, Z., and Zi, Y. (2007). Fault diagnosis of rotating machinery based on improved wavelet package transform and SVMs ensemble. Mechanical Systems and Signal Processing, 21, 688-705.

[38] Lei, Z, Yang, Y., & Wu, Z.(2006). Ensemble of support vector machine for text-independent speaker recognition, International Journal of Computer Science and Network Security 6(5A) 163-167.

[39] Wang, Y. Q. (2008). Building credit scoring systems based on support-based support vector machine ensemble, Fourth International Conference on Natural Computation, 323-326.

[40] Hansen, L., & Salamon, P. (1990). Neural network ensembles, IEEE Trans. PAMI, Vol. 12, 993-1001.

[41] Ho, T. K. (2002). Multiple classifier combination: lessons and next steps, in Hybrid Methods in Patter Recognition. (Ed. By H. Bubke & A. Kandel), World Scientific, 2002.

[42] Zhou, Z. H. Wu, J. X., & Tang, W. (2002). Ensembling neural networks: many could better than all, Artificial Intelligence, 137, 239-263.

[43] Oliveira, L. S., Sabourin, R., Bortolozzi, F., & Suen, C.Y. (2003). Feature selection for ensembles: a hierarchical multi-objective genetic algorithm approach, ICDAR 2003.

[44] Optiz, D., & Maclin, R. (1999). Popular ensemble methods: an empirical study. Journal of Artificial Intelligence, 11, 169-198.

[45] Maia, T. T., Braga, A. P., & Carvalho, A. F. (2008). Hybrid classification algorithms based on boosting and support vector machines, Kybernetes, 37(9), 1469-1491

Myoung-Jong Kim

He is a professor of Division of Business in Pusan National University. He received a BS and MS degree from Sungkyunkwan University, and a PhD from Korea Advanced Institute of Science and Technology in Korea. He has published many papers related to the business applications of Artificial Intelligence. His main research interests are Data Mining, e-business architecture, knowledge management and intelligent systems in accounting and finance fields

Hong-Bae Kim,

He is a professor of Division of Business in Dongseo University. He received a BS from Pusan National University, and a MS degree and PhD from Korea University in Korea. He has published many papers related to the financial and derivative markets. His main research interests are credit analysis and credit derivative market in finance.

Dae- Ki Kang

He is an assistant professor of Computer & Information Engineering at Dongseo University. He was a senior member of engineering staff at The Attached Research Institute of Electronics & Telecommunications Research Institute in South Korea. He earned a PhD in computer science from Iowa State University in 2006. His research interests include intrusion detection, security informatics, ontology learning and relational learning. Prior to joining Iowa State, he worked at a Bay-area startup company and at Electronics and Telecommunication Research Institute in South Korea. He received a science master degree in computer science at Sogang University in 1994 and a bachelor of engineering (BE) degree in computer science and engineering at Hanyang University in 1992.