GEOMETRIC MEAN BASED BOOSTING ALGORITHM TO RESOLVE DATA IMBALANCE PROBLEM

Myoung-Jong Kim, School of Business, Pusan National University, Busan, 63 Beon-gil 2, Busandaehag-ro, Geumjeong-gu, Busan 609-735, Republic of Korea, mjongkim@pusan.ac.kr

Dae-Ki Kang, Division of Computer & Information Engineering, Dongseo University, 47, Churye-Ro, Sasang-Gu, Busan, 617-716, Republic of Korea, dkkang@dongseo.ac.kr

Abstract

In classification or prediction tasks, data imbalance problem is frequently observed when most of samples belong to one majority class. Data imbalance problem has received a lot of attention in machine learning community because it is one of the causes that degrade the performance of classifiers or predictors. In this paper, we propose geometric mean based boosting algorithm (GM-Boost) to resolve the data imbalance problem. GM-Boost enables learning with consideration of both majority and minority classes because it uses the geometric mean of both classes in error rate and accuracy calculation. We have applied GM-Boost to bankruptcy prediction task. The results indicate that GM-Boost has the advantages of high prediction power and robust learning capability in imbalanced data as well as balanced data distribution.

Keywords: Data imbalance, GM-Boost, Bankruptcy prediction.

Preferred journals:

Journal of the Association for Information Systems (JAIS)

European Journal of Information Systems (EJIS)

1 INTRODUCTION

Data imbalance problem is frequently observed in various classification and prediction tasks when most of training samples belong to one majority class. Although most classification algorithms are trained under the assumption that the ratio of the classes is almost equal, in real classification, this assumption is frequently violated. Data imbalance problem is reported in a wide range of classification tasks, such as oil spill detection (Kubat et al. 1998), response modeling (Shin & Cho 1997), remote sensing (Bruzzone & Serpico 1997), and scene classification (Yan et al. 2003). It is also pervasive in business applications including card fraud detection (Fawcett & Provost 1997) and credit rating (Kwon et al. 1997).

Recently, data imbalance problem has received a lot of attention in the machine learning community because it is one of the main causes that degrade the performance of machine learning algorithms in classification tasks. There are two main reasons why data imbalance causes degradation in performance of machine learning algorithms (Kang & Cho 2006; Kotsiantis et al. 2007; Wang & Japkowicz 2009).

The first reason is associated with the objective function of classification algorithms. One of widely used objective functions for classification algorithms is the arithmetic mean based accuracy (hereafter, arithmetic accuracy) which is a ratio of the number of correctly classified instances over the number of total instances. However, in the presence of data imbalance, arithmetic accuracy can be inappropriate because the accuracy is highly dependent on the classification accuracy of majority class samples. For example, bankruptcy is a very rare event. Credit rating agencies such as Moody's anticipate long term average bankruptcy rates of Korean audited companies to be about three to five percent. If all of audited companies are used as a training data set, then arithmetic accuracy of the generated classifier will tend to be abnormally high due to the high accuracy for majority class samples (non-bankrupt companies) despite the low accuracy for minority class samples (bankrupt companies). More specifically, in very imbalanced domains, most standard classifiers will tend to learn how to predict the majority class. While these classifiers can obtain higher predictive accuracies than those that also try to consider the minority class more, this seemingly good performance can be argued as being meaningless (Wang & Japkowicz 2009).

Recently there have been research works to apply receiver operating characteristic (ROC) curve or geometric mean based accuracy (hereafter, geometric accuracy) in measuring performance, because these measures have advantages of reflecting both the accuracy on the majority and minority classes at the same time (Fawcett 2006; Kubat et al. 1997).

The second reason for the degradation in performance is the distortion of decision boundaries resulting from imbalanced distribution of the classes. As the imbalance of data is getting severe, the decision (classification) boundary of majority class tends to invade the decision boundary of the minority class, so that the decision boundary of majority class is gradually expanded while the decision boundary of minority class is gradually reduced. This problem eventually causes the decrease in the accuracy for minority class.

For the alternatives to solve this problem, various methods have been proposed including under-sampling, over-sampling, cost adaptive strategies, and boosting algorithms. Under-sampling method decreases the number of samples from majority class to that of minority class in order to make the number of both classes the same. Over-sampling method, which is opposite to under-sampling method, increases the number of samples in minority class to meet the number of samples in majority class. In cost adaptive strategies, the penalty is assigned to misclassified instances from minority class. Cost adaptive strategies have a merit that they do not distort data distribution, while their effects are marginal when data imbalance is severe. Recently, various boosting algorithms have been proposed as alternatives for data imbalance problems including SMOTEBoost (Chawla et al. 2003) and RUSBoost (Seiffert et al. 2008). In particular, SMOTEBoost is an application of boosting techniques to over-

sampled data generated by synthetic minority over-sampling technique (SMOTE) (Chawla 2002). SMOTE effectively creates a new synthetic minority class sample by combining a certain sample with k similar minority class observations multiplied by Gaussian random distances where k is the number of the nearest neighbours, while boosting algorithm proceeds training on over-sampled data through repetitive sampling process which focuses on misclassified observations. In this way, SMOTEBoost can reinforce the training over samples from minority class to be likely misclassified. However, the boosting algorithm can be inappropriate as for over-fitting problem because its objective function is still measured in terms of arithmetic accuracy and arithmetic errors. New minority class samples, which are generated from SMOTE, are likely to have the higher similarity than majority data samples. Most standard learning algorithms will tend to generate classifiers focusing on samples with higher similarity because that strategy is helpful to maximize the objective function, i.e. arithmetic accuracy. This drawback might increase generalization errors when classifiers are applied to new validation data set which is not trained.

This paper proposes geometric mean based boosting (GM-Boost) which is a novel boosting algorithm applying the concept of geometric accuracy to AdaBoost algorithm (Freund and Schapire 1997). It has the advantage of enabling balanced learning against both majority and minority classes. The proposed GM-Boost algorithm is applied to bankruptcy prediction task which is one of the typical data imbalance problems in business domains. Two different data samples are constructed to verify the performance of GM-Boost algorithm. At the first stage, five sample groups is constructed according to different data balance rates (1:1(denoted as A), 1:3(B), 1:5(C), 1:10(D), and 1:20(E)) and perform classification experiments using AdaBoost and GM-Boost for performance verification in imbalanced data.

At the second stage, SMOTE algorithm is applied to generate new bankrupt company data sets for B, C, D, and E of the first stage, and thus bankrupt companies to normal companies are in the ratio of 1:1. We apply the newly sampled sets to AdaBoost and to GM-Boost experiments for the performance verification of GM-Boost in balanced data.

Experimental results show that GM-Boost has the advantages of high prediction power and robust learning capability in imbalanced data distribution as well as in balanced data distribution.

This paper is organized as follows. The problems of data imbalance and the previous methods to solve these problems are briefly described in section two. Three algorithms including SMOTE, AdaBoost and GM-Boost, which are used in this research, are explained in section three. In section four, we explain the processes of data collection and experimental design. Experimental results are presented in section five. We conclude with future research directions in section six.

2 DATA IMBALANCE PROBLEM IN BINARY CLASSIFICATION PROBLEMS

In this section, we will describe data imbalance problems and then previously proposed methods to resolve data imbalance problem.

2.1 Data Imbalance Problem

Kang and Cho (2006) constructed six sample groups according to different data balance rates (1:1, 1:3, 1:5, 1:10, 1:30, and 1:50) in order to analyze the effects of data imbalance on classification accuracy of SVM. Their experimental results show that for the two sample groups with little or no data imbalance problem (1:1, and 1:3), the sizes of decision boundary areas of the two classes are similar to each other. However, for the sample groups with serious data imbalance problems (1:5, and 1:10), the area of minority class is reduced because the area of the majority class invades the area of minority class, and thus the classification accuracy for minority class samples gets degraded. Especially, for the

sample groups with extreme data imbalance (1:30, and 1:50), it is reported that the decision boundary area for minority class is excessively small, which makes the classification for minority class meaningless. Also, they reported that, as the data imbalance is getting severe, arithmetic accuracy over total samples steadily increases due to the high accuracy over samples of majority class, while the arithmetic accuracy for minority class is dramatically reduced, and thereby geometric accuracy over total samples gradually decreases. They argued that these results demonstrate that arithmetic accuracy is not a suitable objective function for imbalanced data.

Wu and Chang (2003) asserted that data imbalance leads to skewing the boundaries of SVM. Firstly, the decision boundary area of majority class is expanded and the decision boundary area of minority class is reduced, as the data imbalance is getting severe. This problem causes the distortion of decision boundary area. Secondly, data imbalance induces that samples of minority class do not reside in the decision boundary area of minority class, as the decision boundary area of minority class is getting small. Consequently, the possibility becomes very high that the classifier will classify a sample as a majority class.

2.2 The Approaches to Resolve Performance Measure Problem

		Predicted category	
		Positive	Negative
Actual class	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

Table 1. Confusion Matrix.

Let us assign minority class as positive and majority class as negative in order to explain the concept of accuracy. Simple arithmetic mean based accuracy which is widely used is calculated as (TP+TN)/(TP+FN+FP+TN) as shown in the confusion matrix as of Table 1. As described before, arithmetic accuracy is a proper performance measure for classifiers in balanced data set. However, under data imbalance, it is not a proper performance measure anymore because it is highly influenced by the classification accuracy of majority class (Kang & Cho 2006; Kotsiantis et al. 2007; Wang & Japkowicz 2009).

Geometric accuracy and ROC analysis are proposed to resolve this problem. The geometric accuracy is calculated as a square root of sensitivity multiplied by specificity where sensitivity and specificity are TP/(TP+FN) and TN/(FP+TN) respectively (Kubat et al. 1997).

In ROC analysis, we usually plot and connect each sample to generate a polyline ordered by their classification score in two dimensional Cartesian coordinate system where x axis denotes 1- specificity and y axis denotes sensitivity. The accuracy of the classifier is calculated as an area under the ROC curve (AUROC). AUROC is 1.0 in a perfect model and AUROC is 0.5 in a random guess model. Most models generally have AUROC which is higher than 0.5 and lower than 1.0. As AUROC becomes closer to 1.0, the model is regarded as more accurate (Fawcett 2006).

2.3 The Approaches to Resolve Data Distribution Problem

The previously proposed methods to resolve data imbalance can be divided into twofold: data sampling and the assignments of weights (penalties) to misclassified instances (Kang & Cho 2006).

There have been two types of data sampling strategies, under-sampling and over-sampling, which is generally used to resolve data imbalance problems. Data sampling balances the class distribution in the training data by either removing examples from the majority class or adding examples to the

minority class. Under-sampling removes a portion of majority class samples in accordance with the number of minority class samples. The primary drawback of under-sampling is the loss of information associated with deleting examples from the training data. However, it has the benefit of decreasing the time required to train models since the training dataset size is reduced (Seiffert et al. 2008). It can also successfully resolve data imbalance problems when adequate rules to select and remove samples are adopted (Japkowicz & Stephen 2002; Kubat et al. 1998; Laurikkala 2002).

Over-sampling uses techniques of data duplication or data generation to increases the number of minority class samples (Chawla et al. 2003; Japkowicz & Stephen 2002). Over-sampling, on the other hand, does not result in the loss of information. However, it can lead to over-fitting (Drummond & Holte 2003) and increased model training times due to increased training dataset size (Japkowicz & Stephen 2002; Kubat et al. 1998; Laurikkala 2002; Seiffert et al. 2008).

In weights assignments methods, cost adaptive learning strategies are generally used to impose different penalties on misclassified patterns. That is, the higher penalty is imposed on the misclassification when a sample in minority class is misclassified than the penalty is when a sample in majority class is misclassified (Elkan 2001; Provost & Fawcett 2001). This method has a merit that it can avoid the problems like as information loss of under-sampling or generalization error of over-sampling, while it can cause to generate unstable classifiers due to the excessive sensitivity about the samples.

Recently, a variety of hybrid data sampling/boosting algorithms such as SMOTEBoost (Chawla et al. 2003), RUSBoost (Seiffert et al. 2008), etc. have been applied to data imbalance problem and have shown successful results. Both SMOTEBoost and RUSBoost introduce data sampling into the AdaBoost algorithm. SMOTEBoost using an over-sampling technique called SMOTE which creates new minority class examples by extrapolating between existing examples. RUSBoost applies random under-sampling (RUS), a technique which randomly removes examples from the majority class.

Another technique that can be used to improve classification performance is boosting. While data sampling is designed to resolve the class imbalance problem, boosting is a technique that can improve the performance of any weak classifier (whether or not the data is imbalanced). The most common boosting algorithm is AdaBoost (Freund & Schapire 1997). AdaBoost sequentially generate ensemble of classifiers. It assigns higher weights to misclassified observations than to correctly classified observations for each iteration. In the next iteration, AdaBoost leads to more learning opportunity to misclassified observations with higher weights because it selects training samples from the sample of the previous iteration according to the weights assigned to observations. Upon completion, all constructed classifiers participate in a weighted vote to classify untrained samples. In this way, AdaBoost has a merit that it can strengthen learning on minority class samples with the high probabilities of misclassification.

3 GM-BOOST ALGORITHM

In this section, we will explain SMOTE, AdaBoost, and GM-Boost algorithms which are used in this research.

3.1 The Approaches to Resolve Data Distribution Problem

SMOTE algorithm is used to generate new samples for minority class data. SMOTE algorithm combines a certain observation with k similar minority class samples to generate a new sample according to the following calculation: $X_{new} = X + rand(0,1) \times (X_n - X)$ where X_{new}, X, and X_n respectively means newly generated sample, the original sample, and the nearest k samples to the original sample. SMOTE algorithm consists of three steps as followings; Firstly, the nearest k samples to the original sample is chosen, secondly the distances of the original sample and k samples is

multiplied by a random number between zero and one, and finally, the average of the multiplied distances is added to the original sample in order to generate a new sample. In this way, we repeat SMOTE sampling to increase the samples of minority class until both the numbers of the minority class and majority class become same.

3.2 AdaBoost Algorithm

AdaBoost algorithm is one of the most widely used boosting algorithms in binary classification among ensemble learning algorithms, and has been proposed by Freund and Schapire (1997). Basically, boosting is an algorithm that generates a strong learner that is highly accurate by linearly combining multiple weak learners which is more accurate than random guess. In boosting, a new classifier is generated based on the result of the previously generated classifiers focusing on misclassified samples. To explain AdaBoost, we assume an ensemble $C = \{C_1, C_2, \dots, C_K\}$ composed of K base classifiers from n training samples. Then the error rate for k^{th} base classifier (e_k) is calculated as an arithmetic mean, which is as follows.

$$e_k = \sum_{i=1}^{n} w_k(i) L(C_k(x_i), y_i)$$
$$\text{where, } L(C_k(x_i), y_i) = \begin{cases} 1 & C_k(x_i) \neq y_i \\ 0 & C_k(x_i) = y_i \end{cases} \text{ and } \sum_i w_k(i) = 1$$

Note that x_i is a vector of predictor variables for i^{th} observation, y_i is a category of i^{th} observation, and $C_k(x_i)$ is a classification result of k^{th} classifier on the predictor variable vector x_i. For the $(k+1)^{th}$ classifier, the weight for i^{th} observation is adjusted as follows, which impose higher weights on misclassified observations.

$$w_{k+1}(i) = \frac{w_k(i) exp(-\alpha_k C_k(x_i) y_i)}{Z_k}$$
$$\text{where } Z_k = \sum_i w_k(i) exp(-\alpha_k C_k(x_i) y_i)$$

Note that α_k is conceptually interpreted as an importance or accuracy of the classifier, and calculated as $\alpha_k = \frac{1}{2}\ln\left((1 - e_k)/e_k\right)$. When the training samples are constructed for $(k+1)^{th}$ classifier, since higher weights are assigned to misclassified observations, the boosting algorithm can proceed training focused on misclassified observations. The ensemble learning algorithm stops when $e_k > 0.5$. The classification result of the ensemble for i^{th} observation is a weighted mean of base classifiers' classification expressed as follows:

$$C(x_i) = sign\left(\sum_{k=1}^{K} \alpha_k C_k(x_i)\right)$$

Having an advantage of providing learning opportunity to minority class samples, various boosting algorithms based on AdaBoost are frequently applied to data imbalance problem as an alternative solution. As data imbalance is more severe, the error rate for minority class is higher whereas the error rate for majority class is lower. Since higher weights are assigned to minority class samples in the process of constructing training samples for new classifier, the new classifier will strengthen its

learning for minority class. In this way, although learning algorithm is concentrated on majority class samples in the beginning stage of ensemble learning, gradually there become more learning opportunities for minority class samples. Upon such characteristic, AdaBoost has an advantage of yielding robust learning performance even under data imbalance.

However, the boosting algorithms can exhibit the over-fitting and generalization problems because they try to maximize arithmetic accuracy. The error rate of the classifier e_k and the performance of the classifier, α_k, are measures based on arithmetic mean. As mentioned before, measures based on arithmetic accuracy might not be valid as a useful objective function because the objective function based on arithmetic measures tends to generate a strongly biased classification function towards majority class or class with high similarity among samples. Especially, when the boosting algorithms are applied after SMOTE algorithm, which generates a new data sample from a group of adjacent data samples weighted with their inter-distances, it will increase the inductive bias due to the increased similarity among the group of data samples and will eventually aggravate the over-fitting effects. The notion of geometric accuracy, which can consider predictive performances of both majority class and minority class, is introduced to alleviate these problems.

3.3 GM-Boost Algorithm

In addition to the aforementioned assumptions for AdaBoost algorithm, we assume that, out of n training samples, n^+ samples are in minority class and n^- samples are in majority class. Let e_k^+ be the error rate for minority class of k^{th} classifier and e_k^- be the error rate for majority class of k^{th} classifier. Then the geometric mean based error rate e_k, can be defined as follows:

$$e_k = \sqrt{e_k^+ \cdot e_k^-},$$

$$\text{where } e_k^+ = \frac{\sum_{i=1}^{n^+} w_k(i) L(C_k(x_i), y_i)}{\sum_{i=1}^{n^+} w_k(i)} \text{ and } e_k^- = \frac{\sum_{i=1}^{n^-} w_k(i) L(C_k(x_i), y_i)}{\sum_{i=1}^{n^-} w_k(i)}$$

Accordingly, α_k which means classification accuracy of the classifier is calculated as a geometric mean based accuracy of classification accuracies of minority class and majority class.

$$\alpha_k = ln\left(\sqrt{\mu \cdot \alpha_k^+ \cdot \alpha_k^-}\right),$$

$$\text{where } \alpha_k^+ = \frac{1 - e_k^+}{e_k} \text{ and } \alpha_k^- \frac{1 - e_k^-}{e_k}$$

Note that μ is a weighting degree that controls the weight value multiplied to each instance. Following AdaBoost, the weight imposed on the samples for $(k+1)^{th}$ classifier is calculated as follows:

$$w_{k+1}(i) = \frac{w_k(i) \exp(-\alpha_k C_k(x_i) y_i)}{Z_k}$$

$$\text{where } Z_k = \sum_i w_k(i) \exp(-\alpha_k C_k(x_i) y_i)$$

And the final classification result for i[th] observation is calculated as a linear combination of ensemble results and α_k.

$$C(x_i) = sign\left(\sum_{k=1}^{K} \alpha_k C_k(x_i)\right)$$

Since GM-Boost algorithm is based on AdaBoost algorithm, it has an advantage of having more opportunities to strengthen its learning for minority class. Moreover, since it is based on geometric mean based error rate and geometric mean based accuracy, it can carry on its learning with consideration of both majority class and minority class. Figure 1 concisely depicts the procedure of GM-Boost algorithm.

1. Initialize weights $w_i(x_i) = {}^{1}/_{n}, i = 1,2,\dots,n.$
2. Repeat generating classifiers $C_k,\ k = 1,2,\dots,K$ until the stopping criterion(e_k >0.5) is satisfied
a) Learn classifiers $C_k(x_i) \in \{-1,1\}$ based on the training data and their weights
b) Calculate geometric mean based error rates:
$e_k = \sqrt{e_k^+ \cdot e_k^-},$ where $e_k^+ = \frac{\sum_{i=1}^{n^+} w_k(i) L(C_k(x_i), y_i)}{\sum_{i=1}^{n^+} w_k(i)}$ and $e_k^- = \frac{\sum_{i=1}^{n^-} w_k(i) L(C_k(x_i), y_i)}{\sum_{i=1}^{n^-} w_k(i)}$
c) Calculate geometric mean based accuracies:
$\alpha_k = \ln\left(\sqrt{\mu \cdot \alpha_k^+ \cdot \alpha_k^-}\right)$, where $\alpha_k^+ = \frac{1 - e_k^+}{e_k}, \alpha_k^- = \frac{1 - e_k^-}{e_k}$, and μ is a weighting degree
d) Adjust the weights:
$w_{k+1}(i) = \frac{w_k(i) exp(-\alpha_k C_k(x_i) y_i)}{Z_k}$, where $Z_k = \sum_i w_k(i) exp(-\alpha_k C_k(x_i) y_i)$
3. Generate the final result:
$C(x_i) = sign\left(\sum_{k=1}^{K} \alpha_k C_k(x_i)\right)$

Figure 1. GM-Boost algorithm

4 RESEARCH DESIGN

The experimental data used for this research is given from a Korean commercial bank. The bankrupt companies are 500 audited manufacturing companies during year 2002 to year 2005, while the non-bankrupt companies are 2,500 audited manufacturing companies during 2002-2005. We collected 10,000 firm-year financial statements of the non-bankrupt companies during 2001-2004. In this way, we collected a total of 10,000 financial statements based on firms-year standard, and the average bankrupt rate for the four years is about five percent, which falls in the expected range of bankruptcy rate (three to five percent) estimated by professional credit rating agencies.

As for the financial ratios for bankruptcy prediction, we collected thirty financial ratios, which have been usefully applied in the previous corporate bankruptcy prediction researches. The collected ratios are divided into seven financial ratio groups including profitability, debt coverage, leverage, capital structure, liquidity, activity, and size. Consequently, the seven final input variables, each of which has the highest AUROC in each group, are selected. The original thirty financial ratios and the seven selected financial ratios are described in the Table 2.

Variables	AUROC

Profitability	**Ordinary income to total assets***	51.7
	Net income to total assets	44.7
	Financial expenses to sales	49.0
	Financial expenses to total debt	47.3
	Net financing cost to sales	49.2
	Ordinary income to sales	44.5
	Net income to sales	49.1
	Ordinary income to capital	47.2
	Net income to capital	47.1
Debt Coveraage	**EBITDA to Interest expenses***	51.2
	EBIT to Interest expenses	48.2
	Cash operating income to interest expenses	47.5
	Cash operating income to total debt	47.1
	Cash flow after interest payment to total debt	50.5
	Cash flow after interest payment to total debt	50.1
	Debt repayment coefficient	48.8
Leverage	**Total debt to total assets***	50.9
	Current assets to total assets	50.3
Capital Structure	**Retained earning to total assets***	52.5
	Retained earning to total debt	51.7
	Retained earning to current assets	50.1
Liquidity	**Cash ratio***	45.5
	Quick ratio	45.1
	Current assets/current Liabilities	42.2
Activity	**Inventory to sales***	30.5
	Current liabilities to sales	28.3
	Account receivable to sales	27.2
Size	**Total assets***	24.2
	Sales	21.4
	Fixed assets	22.6

*Table 2. Area under the ROC curve (AUROC) values of the 31 financial ratios. * Note that the chosen seven financial ratios are in boldface and are denoted with '*'.*

Although multicollinearity test is not directly related to model's predictive power, it is a check-point of the model. Variance inflation factor (VIF) analysis is performed to check for multicollinearity among the seven financial ratios. Generally, it is suspected that multicollinearity will present if VIF lies between four and ten, and there is very severe multicollinearity if VIF is higher than ten. Table 3 shows that the chosen variables do not exhibit any substantial multicollinearity because all the VIFs are below four.

Variables	VIF
Ordinary income to total assets	1.36
EBITDA to Interest expenses	2.11
Total debt to total assets	1.77
Retained earning to total assets	2.53
Cash ratio	1.34
Inventory to sales	1.59
Total assets	1.31

Table 3. The result of variance inflation factor analysis on the chosen variables.

5 EXPERIMENTAL RESULTS AND DISCUSSIONS

In this section, we summarize and present the overall performance evaluation results of GM-Boost in imbalanced data distribution as well as balance data distribution. Sequential minimal optimization (SMO) is used as a SVM base classifier and the radial basis function (RBF) is used as a kernel function. There are two parameters in RBF kernels: acceptable error C and kernel parameter δ2. We made up various configurations of the two parameters: varying C from 1 to 250, and δ2 from 1 to 200.

In order to compare and analyze the performance of classifiers from different learning algorithms under data imbalance, we prepared samples through two stages. At the first stage, we chose samples from the total of 10,500 cases, with the ratio of bankrupt companies to normal companies as 1:1(A), 1:3(B), 1:5(C), 1:10(D), and 1:20(E). Then we set 60% of each of them as training samples, and the rest 40% of each of them as test samples. Table 4 shows these configurations of samples in the experiment. We repeated these steps of the first stage fifty times to generate fifty training sample sets and fifty test sample sets for each of the five configurations (A, B, C, D, and E).

Set		**Training**			**Validation**		
		Bankrupt	***Normal***	***Total***	***Bankrupt***	***Normal***	***Total***
A	1:1	300	300	600	200	200	400
B	1:3	300	900	1,200	200	600	800
C	1:5	300	1,500	1,800	200	1,000	1,200
D	1:10	300	3,000	3,300	200	2,000	2,200
E	1:20	300	6,000	6,300	200	4,000	4.200

Table 4. Configurations of imbalanced data samples.

At the second stage, we used SMOTE algorithm, where k is set to five, to generate new bankrupt companies, so that we obtained the number of bankrupt companies same with that of normal companies. Table 5 shows these configurations of samples. We repeated the same sampling process fifty times to generate fifty training sample sets and fifty test sample sets for each of four configurations (B, C, D, and E).

Set		**Training**			**Validation**		
		Bankrupt	***Normal***	***Total***	***Bankrupt***	***Normal***	***Total***
A	1:1	300	300	600	200	200	400
B	1:3	900	900	1,200	200	600	800
C	1:5	1,500	1,500	1,800	200	1,000	1,200
D	1:10	3,000	3,000	3,300	200	2,000	2,200
E	1:20	6,000	6,000	6,300	200	4,000	4.200

Table 5. Configurations of balanced data samples.

5.1 Results on Imbalanced Data

We run fifty validations per each configuration, since we have fifty different pairs of training sample sets and test sample sets per each configuration. Table 6 shows the results of average accuracy of fifty validations. The second and third column of Table 6 show average accuracy for majority and minority class, and the fourth and fifth column show prediction accuracy based on arithmetic and geometric mean, respectively. In case of AdaBoost, as the data imbalance is getting severe, arithmetic accuracy over total samples is steadily increased due to the high accuracy over samples of majority class, while the arithmetic accuracy for minority class is dramatically reduced, and thereby geometric accuracy

over total samples is gradually decreased. In particular, average accuracy for minority class of sample groups C, D, and E is 7%, 4.5%, and 3.5%, respectively. It indicates that the classification for minority class is meaningless and that arithmetic accuracy cannot be a suitable performance measure for imbalanced data. Those results are caused by arithmetic error and accuracy calculation of AdaBoost.

Comparing to AdaBoost, however, GM-Boost shows stable arithmetic accuracy for minority class and geometric accuracy over total samples. The arithmetic accuracy for minority class lies between 0.420 and 0.780 and geometric accuracy lies between 0.619 and 0.800. T-test is performed to analyze the difference of geometric accuracy between both boosting algorithms for the five configurations (A, B, C, D, and E). The results of T-test show that the prediction accuracy between two training algorithms for sample group A is significantly different at 5% level and for sample group B, C, D, and E is different at 1% level, respectively. The difference in geometric accuracies becomes higher, as the data imbalance becomes more severe. These results imply that GM-Boost successfully balances minority category and majority category. It reduces distortion of decision boundary and thus enhances decision support.

Set	**AdaBoost**				**GM-Boost**				**t-value**
	Majority	**Minority**	**Arithmetic**	**Geometric**	**Majority**	**Minority**	**Arithmetic**	**Geometric**	
A	0.820	0.755	0.788	0.787	0.820	0.780	0.800	0.800**	1.851*
B	0.960	0.330	0.803	0.563	0.893	0.630	0.828	0.750*	2.435**
C	0.990	0.070	0.837	0.263	0.891	0.610	0.844	0.737*	2.704**
D	0.999	0.045	0.912	0.212	0.916	0.505	0.879	0.680*	3.291**
E	0.998	0.035	0.952	0.187	0.912	0.420	0.889	0.619*	3.557**

*Table 6. Prediction accuracy and the t-test for the five configurations of imbalanced data samples. ** and * represent significance levels at 1% and 5%, respectively.*

5.2 Results on Balanced Data

We apply the final sampled sets generated from SMOTE to AdaBoost and GM-Boost experiments. Table 7 shows the results of average accuracy of fifty validations. Comparing to classification accuracy shown in Table 6, more robust classifiers can be driven from balanced data set through SMOTE than those from imbalanced data set.

Set	**AdaBoost**				**GM-Boost**				**t-value**
	Majority	**Minority**	**Arithmetic**	**Geometric**	**Majority**	**Minority**	**Arithmetic**	**Geometric**	
A	0.830	0.765	0.798	0.797	0.820	0.780	0.800	0.800	0.552
B	0.750	0.750	0.750	0.750	0.747	0.770	0.753	0.758	1.301*
C	0.775	0.720	0.766	0.747	0.808	0.745	0.798	0.776**	1.685**
D	0.755	0.720	0.751	0.737	0.775	0.765	0.774	0.770*	2.423***
E	0.775	0.695	0.771	0.734	0.776	0.785	0.776	0.780*	2.997***

*Table 7. Prediction accuracy and the t-test for the five configurations of balanced data samples. ***, **, and * represent significance levels at 1%, 5%, and 10%, respectively.*

As noted, the higher is the proportion of new generated samples in minority class, the higher is the similarity among minority class samples. SVM, the base classifier of AdaBoost, will tend to learn focusing on minority samples with high similarity because this strategy is helpful maximizing arithmetic accuracy. Boosting algorithms also try to modify the weight of each instance based on misclassification, but do not try to balance majority class error and minority class error. This problem leads to over-fitting problem and deteriorates the performance of SMOTEBoost in the perspectives of generalization and prediction for novel samples.

In our case, since data set E has the higher proportion of new generated samples and the higher similarity among data samples than any other data sets, it is likely to show the lower prediction performance for novel samples. Hence, while the accuracy of AdaBoost for majority class samples consistently lies on the interval between 0.750 and 0.830, its accuracy for minority class samples becomes lower as the degree of data imbalance is higher. Thus, arithmetic accuracy of AdaBoost stably lies between 0.750 and 0.798, but geometric accuracy continues to deprecate from 0.797 to 0.734. On the contrary, GM-Boost, that employs geometric accuracy, systematically avoids this over-fitting problem because it considers both accuracies of majority class category and minority class category. Consequently, GM-Boost exhibits more robustness and generalization than AdaBoost does for novel test samples.

T-test is performed to compare the prediction accuracy between AdaBoost and GM-Boost for the five configurations (A, B, C, D, and E). The results show that significant difference between two algorithms in classification accuracies for all configurations except the configuration A.

As for the analysis of the results presented in the Table 6 and 7, it is worth noting the following observations. Firstly, the classification accuracies between AdaBoost and GM-Boost for imbalanced data samples are significantly different. Especially, as the data imbalance becomes more severe, the difference in classification accuracies becomes higher. Those results are caused by the difference in the objective function between both boosting algorithms. Most boosting algorithms including AdaBoost try to focus on arithmetic accuracy when they update the weight distribution of instances, but GM-Boost focuses on geometric accuracy.

Secondly, GM-Boost algorithm shows significant difference with AdaBoost in terms of classification accuracies for all configurations which are data-imbalanced except the configuration A which is data-balanced. The T-test results also indicate that there is a significant difference between two algorithms. Consequently, these results indicate that GM-Boost algorithm has highly accurate and robust learning capability in imbalanced data distribution as well as in balanced data distribution.

6 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

Data imbalance problem has received a lot of attention in machine learning community because it is one of the causes that degrade the performance of classifiers or predictors. In our research, we proposed GM-Boost algorithm to resolve data imbalance problem. The proposed GM-Boost algorithm is applied to bankruptcy prediction task which is one of typical data imbalance problems in business domains. Two different data samples are constructed to verify the performance of GM-Boost algorithm. At the first stage, five sample groups are constructed according to different data balance rates (1:1, 1:3, 1:5, 1:10, and 1:20) and classification experiments using AdaBoost and GM-Boost are performed against those imbalanced data sets. At the second stage, SMOTE algorithm is used to generate new bankrupt company data sets and the newly sampled sets are applied to AdaBoost and GM-Boost experiments for the performance verification of GM-Boost in balanced data. Experimental results show that GM-Boost has the advantages of high prediction power and robust learning capability in imbalanced data distribution as well as balanced data distribution. These results on actual bankruptcy data mean that the usefulness of GM-Boost is promising on broader range of real-world problems of decision making.

We expect the following future researches to be conducted to cope with the limitations of GM-Boost.

Firstly, boosting algorithms have drawbacks that degrade classification accuracy when outliers are included in the learning samples or when there is high correlation between the classifiers in the ensemble. Various methods have been proposed to compensate these shortcomings (Cover & Thomas 1991; Darbellay 1999; Maia 2008), and we plan to conduct researches to develop algorithms coupled with those methods.

Secondly, the ensemble algorithm we propose in this research is a modification of a boosting algorithm to solve data imbalance problem. However, it can be possible to solve the data imbalance problem by combining our results with SVM kernel management (Hong 2007; Wu & Chang 2005), so we anticipate future researches in this direction.

Thirdly, we note that our GM-Boost algorithm can be applied to multi-class classification tasks which can have severe data imbalance problem. Especially, credit rating is one of typical multi-class classification tasks in financial area, so we plan to extend our research to this task.

Fourthly, in this research we have applied GM-Boost algorithm to bankruptcy prediction task. We plan to apply GM-Boost algorithm to broader range of more real-world applications such as intrusion detection data (Cieslak et al. 2006), Healthcare data (Joshi et al. 2010), etc. to further verify GM-Boost algorithms.

Finally, we will test GM-Boost algorithm for an experimental setting with under-sampling method (Seiffert et al. 2008) to analyze the performance of different sampling methods when they are used with GM-Boost and other boosting algorithms.

References

Bruzzone, L. and Serpico, S. B. (1997). Classification of imbalanced remote-sensing data by neural networks. Pattern Recognition Letters, 18(11-13), 1323-1328.

Chawla, N., Bowyer, K., Hall, L., and Kegelmeyer, W. (2002). SMOTE: synthetic minority oversampling techniques. Journal of Artificial Intelligence Research 16, 321-357.

Chawla, N., Lazarevic, A., Hall, L., and Bowyer, K. (2003). SMOTEBoost: improving prediction of the minority class in boosting. In Proceedings of the 7th European conference on principles and practice of knowledge discovery in databases. pp. 107-119, Cavtat-Dubrovnik, Croatia.

Cieslak, D. A., Chawla, N. V., and Striegel A. (2006). Combating Imbalance in Network Intrusion Datasets. In Proceedings of IEEE International Conference on Granular Computing, pp. 732-737, Atlanta, GA, USA.

Cover, T. M., and Thomas, J. A. (1991). Element of information theory. John Wiley & Sons.

Darbellay, G. A. (1999). An estimator of the mutual information based on a criterion for independence. Computational Statistics and Data Analysis, 32(1), 1-17.

Drummond, C. and Holte, R. C. (2003). C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In Proceedings of Workshop on Learning from Imbalanced Data Sets II, International Conference on Machine Learning.

Elkan, C. (2001). The foundation of cost-sensitive learning. In Proceedings of the 17th International Joint Conference on Artificial Intelligence, pp. 973-978, Seattle, WA, USA.

Fawcett, T. (2006). An introduction to ROC analysis. Pattern Recognition Letters 27(8), 861-874.

Fawcett T. and Provost, F. (1997). Adaptive fraud detection. Data Mining and Knowledge discovery 1(3), 291-316.

Freund, Y. and Schapire, R. E. (1997). A decision theoretic generalization of online learning and an application to boosting. Journal of Computer and System Science, 55(1), 119-139.

Hong, X. (2007). A kernel-based two-class classifier for imbalanced data sets. IEEE Transactions on neural networks, 18(1), 28-40.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: a systematic study. Intelligent Data Analysis, 6(5), 429-250.

Joshi, A. J., Chandran, S., Jayaraman, V.K., and Kulkarni, B.D. (2010). Hybrid Support Vector Machine for imbalanced data in multiclass arrhythmia classification. International Journal of Functional Informatics and Personalised Medicine, 3(1), 29-47.

Kang, P. and Cho, S. (2006). EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems. In Proceedings of ICONIP 2006, Part I, pp. 837-846. LNCS 4232.

Kotsiantis, S., Tzelepis, D., Kounmanakos, E., and Tampakas, V. (2007). Selective costing voting for bankruptcy prediction. International Journal of Knowledge-based and Intelligent Engineering Systems, 11(2), 115-127.

Kubat, M., Holte, R., and Matwin, S. (1997). Learning when Negative example abound. In Proceedings of the 9th European Conference on Machine Learning (ECML'97).

Kubat, M., Holte, R., Matwin, S. (1998). Machine Learning for the detection of oil spills in satellite radar images. Machine Learning 30(2), 195-215.

Kubat, M. and Matwin, S. Addressing the curse of imbalanced training sets: one-sided selection. In Proceedings of the 14th International Conference on Machine Learning, pp. 179-186.

Kwon, Y. S., Han, I. G., and Lee, K. C. (1997). Ordinal pairwise partitioning (OPP) approach to neural networks training in bond rating. Intelligent Systems in Accounting, Finance and Management, 6(1), 23-40.

Laurikkala, J. (2002). Instance-based data reduction for improved identification of difficult small classes, Intelligent Data Analysis, 6(4), 311-322.

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

Provost, F. and Fawcett, T. (2001). Robust classification for imprecise environments. Machine Learning, 42(3), 203-231.

Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J. and Napolitano, A. (2008). RUSBoost: Improving classification performance when training data is skewed. In Proceedings of the 19th International Conference on Pattern Recognition, pp. 1-4.

Shin, H. J. and Cho, S. Z. (1997). Response modeling with support vector machine. Expert Systems with Applications, 30(4), 746-760.

Wang, B. X. and Japkowicz, N. (2009). Boosting support vector machines for imbalanced data sets. Knowledge and Information Systems, 25(1), 1-20.

Wu, G. and Chang, E. (2003). Adaptive feature-space conformal transformation for imbalanced data learning. In Proceedings of the 20th International Conference on Machine Learning.

Wu, G. and Chang, E. (2005). KBA: Kernel boundary alignment considering imbalanced data distribution. IEEE Transactions on knowledge and data engineering, 17(6), 786-795.

Yan, R., Liu, Y., Jin, R., and Hauptman, A. (2003). On predicting rare classes with SVM ensembles in scene classification. In Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP'03).