Adapting MultiBoost Ensemble for Class Imbalanced Learning

No Author Given

No Institute Given

Abstract. Learning with class imbalanced data sets is a challenging undertaking by the common learning algorithms. These algorithms favor majority class due to imbalanced class representation, noise and their inability to expand the boundaries of minority class in concept space. To improve the performance of minority class identification, ensembles combined with data resampling techniques have gained much popularity. However, these ensembles attain higher minority class performance at the cost of majority class performance. In this paper, we adapt the MultiBoost ensemble to deal with the minority class identification problem. Our technique inherits the power of its constituent and therefore improves the prediction performance of the minority class by expanding the concept space and overall classification performance by reducing bias and variance in the error. We compared our technique with seven existing simple and ensemble techniques using thirteen data sets. The experimental results show that proposed technique gains significant performance improvement on all tested metrics. Furthermore, it also extends inherited advantage over other ensembles of fitting parallel computation.

1 Introduction

Unexplored data keeps piling up due to flourishing of social media, e-commerce, and bioinformatic, etc. Finding useful information from this overgrowing pile of data is a challenging task for scientists and practitioners in machine learning community. Among these challenges, one well established challenge to machine learning community is known as class imbalanced learning. In binary class imbalanced problem, one class dominates other class by large numbers of instances. The class with substantially large numbers of instances is known as majority class and the class with substantially less numbers of instances is known as minority class. For example, detection of oil spills in radar images, finding cancerous cells among non-cancerous cells, finding documents of interest in text classification, fraud detection and so on [1],[2].

The common learning algorithms usually perform better on the majority class as compared to the minority class. This phenomenon is based on two observations. First, the classification rules that predict minority class have higher error rates as compared to classification rules that predict majority class in decision trees. Second, test instances of minority class are more often misclassified as compared to majority class. These two observations are different from each other

as minority class prediction rules's poor performance leads to poor classification performance on majority class instances. The insight for this poor performance by minority class prediction rules is due to the difference between numbers of majority class instances and minority class instances. This imbalance in class representation results from lessened minority class boundaries in concept space and thus, contributes to poor performance of common learning algorithms [3].

It is argued that small disjuncts, overlapping, concept complexity, or unavailability of learning data are also important factors besides class imbalance. These problems are considered characteristic of noise in data. Noisy data degrades the performance of learning algorithms by undermining the decision boundaries or overfitting the model by incorporating incorrect data points. This leads to increase in concept complexity and creation of small disjuncts in the data [4].

Furthermore, ensembles learning is a popular approach to improve the performance of weak learning algorithms. Generally, we can divide ensembles learning into boosting based and bagging based ensembles. Furthermore, there is advanced formation of bagging and boosting based ensembles known as hybrid ensembles. These techniques make hierarchical formation of bagging and boosting ensembles [5]. These ensembles of ensembles have vitalized the power of both boosting and bagging and hence, more effective than any constituent method.

For class imbalanced problem, ensembles have been combined with the data resampling. For example, EasyEnsemble and BalanceCascade have combined hybrid ensembles with exploratory undersampling [6]. However, these techniques also suffered information loss problem. This is due to the fact that merely balanced distribution of minority to majority class is not the optimum solution which causes valuable information loss for majority class in undersampling and repetition of minority class instances results in low diversity among constituent weak learning algorithms in ensemble learners. Our research is motivated by these facts and therefore, we have adapted an ensemble method known as MultiBoost [5] with Synthetic Minority Over-sampling TEchnique(SMOTE) [2] to address the class imbalance, noise and complexity problem in minority class identification. The proposed technique exploits the power of underlying methods by expanding the concept space of minority class and keeping focus on the difficult to learn majority class instances. While, MulitBoost is an ensemble that consists of wagging[7] and AdaBoost, where wagging subcommittees are constituted by AdaBoost. Furthermore, new weights to instances are assigned using the continuous Poisson distribution. In this method, decision committees are formed and produces lower error rate as compared to AdaBoost or wagging alone. In addition, SMOTE is an oversampling method that generates new synthetic instances from minority class instances. These new synthetic instances broaden the decision regions and increase the coverage of minority class. Thus, it increases the recall of minority class instances.

2 Related Work

Owing to the ubiquitous nature of imbalanced data sets in many sensitive domains, different state-of-the-art approaches have proposed over the years. Data resampling is a widely used approach in class imbalanced learning. Data resampling techniques can be divided into undersampling and oversampling. The most basic resampling techniques are random undersampling and random oversampling. In random undersampling, instances from the majority class are randomly discarded to achieve the desired class ratio. The major drawback of this technique is loss of valuable information.

On the other hand, oversampling adds the artificial instances to the minority class. Random oversampling is the simplest method to randomly add the instances to minority class. In addition, different advanced techniques have been proposed to minimize the negative impact of this naive oversampling. Synthetic oversampling is an approach that intelligently generates the minority class instances. For example, Synthetic Minority Over-sampling TEchnique(SMOTE) successfully minimizes the negative effects of random oversampling.

The second approach to handle class imbalanced problem is known as cost sensitive learning. This approach operates based on the assumption that all errors are equal by assigning different costs for classifying instances. This approach assigns high costs to minority class instances, thus learning algorithms may give more attention to minority class instances to learn an effective class boundary [8]. In addition, these learning methods have drawbacks. The misclassification cost of instances is domain dependent and not all learning algorithms are flexible enough to incorporate cost into their mechanism.

The third approach to deal with class imbalanced problem is known as kernel based. This approach provides sophisticated techniques to solve the class imbalance problem. For example, the kernel classifier construction [9] have used orthogonal forward selection and a regularized orthogonal weighted linear squares estimator. Furthermore, a kernel boundary alignment algorithm [10] was suggested for adjusting the SVM class boundary.

In addition, class imbalanced learning approaches can be cross-functionally divided according to ensemble approaches. Ensembles learning algorithms show promising results in error reduction from weak learning algorithms. Ensembles approaches make a committee of weak learning algorithms to form a strong learning algorithm. Weak learning algorithms are applied as the member of a committee to classification task and their results are aggregated as a strong single learning algorithm. Aggregation may be performed using the weighted or majority voting. Two main ensembles learning approaches are Adaboost and bagging.

Ensembles learning approaches can be categorized into cost sensitive and data resampling ensembles. In the former category, cost sensitive ensemble methods combine both algorithms and data level approaches. Cost sensitive learning algorithms integrate different misclassification costs for each class in the learning mechanism [11]. The later category can be divided into three further subcategories. The first subcategory is boosting based. This includes SMOTEBoost

[1], RUSBoost [12], etc. In this kind of approach a data resampling technique is merged into a boosting ensemble to address the class imbalanced problem. In the second subcategory, data resampling techniques are combined with bagging. UnderBagging, OverBagging and SMOTEBagging [13] are such methods that use undersampling, oversampling and SMOTE techniques with bagging, respectively. In the third subcategory, hybrid ensembles are combined with data resampling techniques. In hybrid ensemble approaches, two ensembles are combined to take advantage of constituent. For example, EasyEnsemble, BalanceCascade, etc. EasyEnsemble technique uses different independent training sub sets of majority class and then combines them with minority class instances for an ensemble. Then, a constituent AdaBoost ensemble is trained on each of these independent sub sets. Finally, the results of all constituent ensembles are aggregated.

Hence, it is an established fact that boosting reduces both bias and variance and bagging reduces variance in error. In addition, SMOTE intelligently generates new minority class instances to extend and broaden the concept space boundaries. To utilize the powers of these techniques, we adapted the MultiBoost ensemble with the SMOTE to effectively address the class imbalanced problem. We have conducted extensive experiments with our proposed technique and different related techniques using different frequently used class imbalanced data sets to assert the effectiveness of our approach.

3 Preliminaries

For common expressions related to class imbalanced problem, we reserve special symbols and characters. Let take a base classification learning algorithm L and training set S and represents a committee of learning algorithms with H^*. Let T subcommittees of the size $\sqrt{T}$ are created, where $H^* = T\sqrt{T}$. Suppose, S is a sample vector of n labeled features-class pairs. Each pair (x_i, y_i) associates features $x_i \in$ X and class $y_i \in$ Y. Let t is a counter that moves through a rang from one to maximum number of iterations T, H_t be the weak hypothesis (trained using some classification algorithm, L) trained on iteration t, and $H_t(x_i)$ be the output of hypothesis H_t, for instance, x_i. Let $D_t(i)$ represents the weight of the ith instance on iteration t. The committee's classification $H^*(x)$ is the class that obtains the most votes from the committee members when applies to x. Subcommittee termination index which holds information about iteration at which subcommittee should terminate is represented by vector I_i. The percentage of oversampling, N and the number of nearest neighbors, nn, to be used with SMOTE.

3.1 Synthetic Minority Over-sampling TEchnique (SMOTE).

Synthetic Over-Sampling TEchnique(SMOTE) is an intelligent oversampling technique to deals with the class imbalanced problem. In Synthetic Over-Sampling TEchnique(SMOTE), new synthetic instances are created of minority class by

interpolating the existing one. These new synthetic instances are created near the minority class instances. SMOTE creates synthetic instances for minority class based on information from nn-nearest neighbors randomly. This technique generalized the decision boundaries for the minority class and thus address overfitting problem effectively.

The procedure for SMOTE is defined as follows: For continuous features and each observation of the minority class, identify and select its nn-nearest neighbors (number is user provided). The new synthetic instances are laying between the original instance and its nearest neighbors. First, take the difference between the minority class instance and its neighbors. Then, multiply previously computed difference with random number between 0 and 1. Then, add this feature vector into original feature vector. For the nominal features, take majority votes between the feature vector used for oversampling and its neighbors, in case of tie take at random and then assign the value to newly created instance. Through this method a new synthetic instance is created by using the existing instances. In the nearest neighbors computation, SMOTE uses different way for discrete and continue features. Nearest neighbors for discrete features are calculated with Value Distance Metric and Euclidean Distance is used for the continuous features.

3.2 MultiBoost.

MultiBoost is an ensemble method that combines a variant of bagging i.e. wagging with the AdaBoost. In bagging, different training sets are created by sampling with replacement which are equal to the size with original training set but some instances might repeat. This is known as bootstrap samples. Multiple weak learning algorithms are trained on these samples. At the end, these weak learning algorithms are combined by voting. However, wagging requires learning algorithms to use instances with different weights. Wagging assigns random weights to instances in sample and select all instances as contrast to the bagging where random bootstrapping is used to change the probability of instances. The instances weights are assigned by continuous Poisson distribution.

On the other hand, AdaBoost combines weak learning algorithms to form a strong learning algorithm. At each iteration, a weak learning algorithm is added and instance weights are updated based on their classification decisions. Therefore, succeeding weak learning algorithm emphasis more on the instances that preceding weak algorithm misclassified. MultiBoost consists of T wagged subcommittees of the size $\sqrt{T}$ which are formed by AdaBoost. MultiBoost sets a target subcommittee member index, I_i which allows premature termination of boosting subcommittee due to low or high error. On abnormal termination of current subcommittee, it increases the size of next subcommittee and this process repeats until the target committee size is accomplished. The final result is a weighted sum of wagging subcommittees as contrast to bagging. It is to mention that current implementation differs from strict wagging of boosted subcommittees by using same instance weights for the first subcommittee rather than continuous Poisson distribution. It is done in the notion that first

subcommittee which uses initial equal weights for instances instead of continuous Poisson distribution would increase the diversity of MultiBoost ensemble. MultiBoost not only inherits bias and variance reduction properties from its constituent but also offers computational edge over AdaBoost such that it is amenable to parallel execution.

4 SMOTEMultiBoost

To adapt MultiBoost for class imbalanced learning, it is necessary to consider the following important factors. Multiboosting reduces both variance and bias in classification tasks but it is not adequate for imbalanced data sets. The objective is to reduce the bias towards majority class by increasing the weights of the minority instances while keeping the performance of majority class. We introduce the SMOTE in boosting iterations to give minority class instances higher priority and learn broader regions for minority class. Synthetic Minority Over-sampling TEchnique (SMOTE) is an oversampling technique that indirectly changes the instances weights and thus incompatible with the MultiBoost because it uses reweighting instead of resampling. As, MultiBoost uses continuous Poisson distribution for resetting the instances weights. To adapt the SMOTE, we assign the weights to new synthetic instances created by the SMOTE. SMOTEBoost outperforms competing methods by implementing boosting by reweighing [14]. So, considering the potential edge of reweighting instead of resampling and to adjust with the MultiBoost, we assign the average weight of nearest neighbors weights to the new synthetic instances. We used this method because average weighting performs better than other weighting approaches. The number of subcommittees formed and their sizes are determined by user parameter. The SMOTEMultiBoost algorithm is shown in Algorithm 1.

In summary, the SMOTEMultiBoost algorithm is a combination of MultiBoost and SMOTE. SMOTEMultiBoost takes an argument T subcommittees. Continuous Poisson distribution is used by each wagged subcommittee for setting instances weights. An AdaBoost constituent is called for each subcommittee having size equals to $\sqrt{T}$. For each iteration of AdaBoost, SMOTE technique synthetically generates the minority class instances. To adapt with the MultiBoost, new synthetic instances generated by SMOTE are assigned average weights calculated from their nearest neighbors weights. A weak hypothesis H_t is formed and evaluated. If classification error is too big or zero, current AdaBoost constituent is aborted and a next subcommittee is formed with increased size to compensate early termination of previous subcommittee. Finally, all subcommittees are combined for weighted vote H^*.

5 Experimental Setup

We employed thirteen binary class imbalanced data sets in our experiments. We applied different competing methods including SMOTEMultiBoost on these data sets to compare and evaluate the effectiveness of our proposed method.

Algorithm 1 SMOTEMultiBoost

Input Data set, S. A weak learning algorithm, L. Number of iterations, T. Vector, I_i where $i \geq 1$. SMOTE percentage, N.

1. $S' = S$ with instance weights assigned to be 1.
2. Set k = 1.
3. For t = 1 to T {
4. If $I_k = t$ then
5. Reset S' to random weights obtained using continuous Poisson distribution.
6. Normalize S' to sum to 1.
7. Set k=k+1.
8. Create temporary training data set S'_t with distribution D'_t by generating N synthetic instances from minority class C_m using SMOTE.
9. Normalize S'_t to sum to n.
10. $H_t = L(S'_t)$
11. $e_t = \frac{\sum_{x_j \in S'_t : H_t(x_j) \neq y_j} D'_t(x_j)}{n}$
12. If $e_t > 0.5$ or $e_t = 0$ then
13. Go to step 5
14. $\beta_t = \frac{e_t}{1-e_t}$
15. For each $x_j \in S'_t$,
16. $D_{t+1}(x_j) = \frac{D_t(x_j)}{Z_t} \times \begin{cases} \beta_t & \text{if } H_t(x_j) \neq y_j \\ 1 & otherwise \end{cases}$

where Z_t is a normalization constant which enable D_{t+1} to be a distribution

17. }

Output the final classifier:
$H^*(x) = argmax_{y \in Y} \sum_{t:H_t(x)=y} log \frac{1}{\beta_t}$.

5.1 Data sets.

We used binary class data sets in our experiments with different ratios of the majority to minority class. We used publicly available data sets from KEEL data repository [15] along with some used in [1]. Detail statistics of these data sets are described in table 5.1. In the Satimage data set all classes are collapsed except smallest class into one so that we can get binary skewed data set.

5.2 Evaluation Metrics.

In this paper, we employed different performance evaluation metrics to evaluate the performance of the proposed method. For classification problems contingency table or confusion matrix is widely used. Accuracy is a commonly used measure in classification problems. However, we can not measure rigorously the performance of learning algorithms on skewed data sets with accuracy. Therefore, first, we used Geometric Mean(G-mean) in our experiments. True positive rate (Acc^+),

Table 1. Statistics of the data sets are used in our experiments include data set names, sizes, imbalance ratios (IR) and number of attributes. The table is sorted according to IR.

Data Set	Size	IR	#attri
Ionosphere	351	1.79	34
Glass1	214	1.82	9
Wisconsin	683	1.86	9
Pima	768	1.87	8
Phoneme	5403	2.4	5
Yeast1	1484	2.46	8
Vehicle2	846	2.88	18
Vehicle1	846	2.88	18
Hepatitis	155	3.84	19
Satimage	6435	9.28	36
Glass-0-1-6-vs-2	192	10.29	9
Ecoli-0-1-4-7-vs-2-3-5-6	336	10.59	7
Mammography	11183	42	7

true negative rate (Acc^-) and G-mean can be represented as:

$$TruePositiveRate(Acc^+) = \frac{TP}{TP + FN} \tag{1}$$

$$TrueNegativeRate(Acc^-) = \frac{TN}{TN + FP} \tag{2}$$

$$G - mean = \sqrt{(Acc^+) \times (Acc^-)} \tag{3}$$

Second, we used F-measure in our experiments. F-measure uses both precision(p) and recall(r) to calculate the score. Recall(r), precision(p) and F-measure are calculated as:

$$recall(r) = \frac{TP}{TP + FN} \tag{4}$$

$$precision(p) = \frac{TP}{TP + FP} \tag{5}$$

$$F - measure = \frac{2pr}{p + r} \tag{6}$$

Finally, we used ROC curve in our experiments. A receiver operating characteristic (ROC) curve is a two-dimensional representation of classifier performance using false positive rate(fpr) on x-axis and true positive rate(tpr) on y-axis.

5.3 Evaluation Algorithms.

We used eight different learning algorithms for our experiments. These are CART, MultiBoost, SMOTE, BalanceCascade, EasyEnsemble, RUSBoost, SMOTEBoost and our proposed SMOTEMultiBoost(SMB). We used CART as

baseline with default parameters except pruning set to false. Three different class distributions with 35%, 50% and 65% are used in our experiments. Five nearest neighbors are used in SMOTE. The number of subcommittees and size of those subcommittees for MultiBoost and SMOTEMultiBoost(SMB) are set to three. Thus, it became total nine classifiers. Similarly, nine classifiers are used for RUSBoost, SMOTEBoost, EasyEnseble and BalanceCascade, for fair comparison. Experiments are conducted using WEKA. Ten fold cross validation is employed as the evaluation mechanism and each experiment is repeated ten times.

6 Experimental Results and Analysis

The class imbalanced learning algorithms are designed specifically for imbalanced data sets. These algorithms try to improve the prediction performance of minority class while keeping the prediction performance of the majority class. It is preferable for learners in class imbalance problem to have higher value for true positive rate which also known as accuracy over minority class. On the other hand, it is also preferable to maintain accuracy over majority class known as true negative rate. Our proposed approach focuses on both true positive rate and true negative rate represented by G-mean measure. We report only performance of learning algorithms using 65% minority class distribution on all data sets using G-mean and F-measure in Tables 2, and 3, respectively.

G-mean values are depicted for all eight techniques in Table 2. From the table, it is observable that our technique performs better than all other competing techniques. EasyEnsemble and BalanceCascade are failed by performing nowhere close to the oversampling methods even RUSBoost. The poor performance of these methods can be characterized as the rarity of minority class instances in the data sets. The instances of minority class are already rare, naive undersampling of a majority class to achieve a balanced class distribution with the minority class causes the majority class instances rare as well. However, both SMOTE and SMOTEBoost preform well and their performance is closest to our method.

Similarly, values of F-measure are depicted in Table 3. From the table, it is apparent that our technique outperformed all other techniques except in Hepatitis and Mammography data sets. Additionally, SMOTE and SMOTEBoost are competing methods with no clear winner on given metrics. We assert the statistical significance of the performance gain using Friedman test[16]. In our experiments, we used SMOTEMultiBoost as base learner. SMOTEMultiBoost(SMB) shows significant ($p < 0.05$) performance gain over all other methods using G-mean and F-measure metrics. These statistical significance gains have been depicted with stars in Tables 2, 3.

Furthermore, Figure 1 depicts ROC curves produced by different techniques including our technique on thirteen data sets. For the mammography data set, CART, EasyEnsemble and BalanceCascade do not perform any better than random prediction. For all data sets the performance of our methods is better than

Table 2. Performance comparison of different methods including our proposed SMOTEMultiBoost(SMB) on various data sets using G-mean. In table, text with star shows statistical significance at ($p < 0.05$) using the Friedman test with respect to SMOTEMultiBoost(SMB) as Base learner and all other methods

Data Set	CART	MultiBoost	SMOTE	BalanceCascade	EasyEnsemble	RUSBoost	SMOTEBoost	SMB
Ionosphere	0.8774(8)	0.9102(6)	0.9490(3)	0.9226(5)	0.9063(7)	0.9600(2)	0.9455(4)	0.9779(1)
Glass1	0.7259(8)	0.7616(7)	0.9204(2)	0.8154(5)	0.7770(6)	0.9169(4)	0.9194(3)	0.9762(1)
Wisconsin	0.9451(8)	0.9627(7)	0.9842(2)	0.9707(5)	0.9660(6)	0.9820(4)	0.9868(3)	0.9975(1)
Pima	0.6701(8)	0.6952(7)	0.8855(4)	0.7270(6)	0.7325(5)	0.8873(3)	0.8992(2)	0.9633(1)
Phoneme	0.8372(8)	0.8701(7)	0.9402(3)	0.9060(5)	0.8737(6)	0.9389(4)	0.9508(2)	0.9831(1)
Yeast1	0.6420(8)	0.6493(7)	0.8935(3)	0.5565(6)	0.6840(5)	0.8662(4)	0.9030(2)	0.9650(1)
Vehicle2	0.9396(8)	0.9685(7)	0.9889(3)	0.9856(4)	0.9752(6)	0.9772(5)	0.9904(2)	0.9980(1)
Vehicle1	0.6498(7)	0.6577(6)	0.9176(3)	0.5565(8)	0.7181(5)	0.8771(4)	0.9252(2)	0.9790(1)
Hepatitis	0.5910(8)	0.6560(6)	0.8357(4)	0.7556(5)	0.6114(7)	0.9072(1)	0.8721(3)	0.8858(2)
Satimage	0.7302(7)	0.7395(6)	0.9628(2)	0.4124(8)	0.8097(5)	0.9162(4)	0.9627(3)	0.9919(1)
Glass-0-1-6...	0.4124(6)	0.3626(7)	0.9321(2)	0.3526(8)	0.4671(5)	0.8040(4)	0.9268(3)	0.9675(1)
Ecoli-0-1-4-7...	0.8084(7)	0.8266(6)	0.9440(2)	0.4662(8)	0.8372(5)	0.9092(4)	0.9377(3)	0.9790(1)
Mammography	0.0273(8)	0.5011(5)	0.7685(1)	0.4569(7)	0.4775(6)	0.6762(4)	0.7432(3)	0.7511(2)
Average Rank	7.61	6.46	2.61	6.15	5.92	3.61	2.69	1.15
Friedman Test	⋆ 3.11491E-4	⋆ 3.11491E-4	⋆ 0.00228	⋆ 3.11491E-4	⋆ 3.11491E-4	⋆ 0.00228	⋆ 3.11491E-4	Base

Table 3. Performance comparison of different methods including our proposed SMOTEMultiBoost(SMB) on various data sets using F-measure. In table, text with star shows statistical significance at ($p < 0.05$) using the Friedman test with respect to SMOTEMultiBoost(SMB) as Base learner and all other methods

Data Set	CART	MultiBoost	SMOTE	BalanceCascade	EasyEnsemble	RUSBoost	SMOTEBoost	SMB
Ionosphere	0.8493(8)	0.8944(7)	0.9552(2)	0.9537(5)	0.9388(6)	0.9574(4)	0.9551(3)	0.9794(1)
Glass1	0.6522(8)	0.7014(7)	0.9384(2)	0.8841(5)	0.8465(6)	0.9117(4)	0.9365(3)	0.9802(1)
Wisconsin	0.9276(8)	0.9501(7)	0.9885(3)	0.9766(5)	0.9741(6)	0.9806(4)	0.9901(2)	0.9981(1)
Pima	0.5800(8)	0.6123(7)	0.9127(3)	0.7907(5)	0.7800(6)	0.8810(4)	0.9190(2)	0.9697(1)
Phoneme	0.7777(8)	0.8228(7)	0.9572(3)	0.9357(4)	0.8990(6)	0.9347(5)	0.9601(2)	0.9860(1)
Yeast1	0.5190(8)	0.5353(7)	0.9062(3)	0.8430(5)	0.8161(6)	0.8596(4)	0.9103(2)	0.9707(1)
Vehicle2	0.9123(8)	0.9553(7)	0.9921(4)	0.9924(3)	0.9866(5)	0.9753(6)	0.9927(2)	0.9984(1)
Vehicle1	0.5098(8)	0.5330(7)	0.9416(3)	0.8681(5)	0.8515(6)	0.8728(4)	0.9421(2)	0.9843(1)
Hepatitis	0.4300(8)	0.5201(7)	0.8316(6)	0.8811(4)	0.8837(2)	0.9051(1)	0.8723(5)	0.8827(3)
Satimage	0.5692(8)	0.6365(7)	0.9722(2)	0.9574(5)	0.9631(4)	0.9118(6)	0.9688(3)	0.9931(1)
Glass-0-1-6...	0.2319(7)	0.2164(8)	0.9468(2)	0.9414(4)	0.9440(3)	0.8057(6)	0.9349(5)	0.9684(1)
Ecoli-0-1-4-7...	0.6863(8)	0.7545(7)	0.9253(4)	0.9685(3)	0.9741(2)	0.9070(6)	0.9226(5)	0.9803(1)
Mammography	0.0035(7)	0.0023(8)	0.8666(1)	0.4900(6)	0.6754(4)	0.6442(5)	0.8121(2)	0.7236(3)
Average Rank	7.84	7.15	2.92	4.53	4.76	4.53	2.92	1.30
Friedman Test	⋆ 3.11491E-4	⋆ 3.11491E-4	⋆ 0.00228	⋆ 3.11491E-4	⋆ 0.00228	⋆ 0.00228	⋆ 0.00228	Base

any other method. The performance of SMOTE, RUSBoost and SOTEBoost is better than BalanceCascade and EasyEnsemble on all data sets. SMOTE and SMOTEBoost are competing with each other followed by RUSBoost. It is generally observed that resampling degrades the performance of the majority class. However, the domination of our technique in ROC space over all other techniques indicates that inherited power of wagging, boosting and SMOTE reduce the error and thus increase in true positive rate and decrease in false positive rate. We conducted experiments with different techniques including our technique on thirteen data sets with three different minority to majority class distributions (35%, 50%, 65%). There is no clear trend which distribution produces better

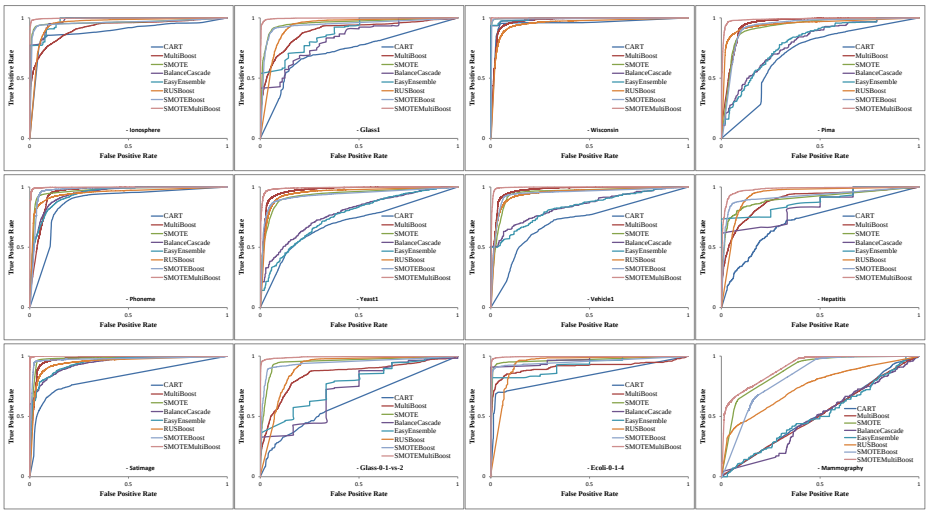

Fig. 1. Comparison of CART, MultiBoost, SMOTE, BalanceCascade, EasyEnsemble, RUSBoost, SMOTEBoost and SMOTEMultiBoost(SMB)on all data sets. SMOTE-MultiBoost(SMB) dominates over all other techniques in the ROC space.

results than other. However, it is apparent that suitable distribution is the attribute of the particular data set. An interesting thing to mention is that the G-mean, F-measure and ROC curve of the RUSBoost is generally better than EasyEnsemble and BalanceCascade. This is probably due to the fact that only balanced distribution of minority to majority class is not suffice because majority class undersampling causes valuable information loss and repetition of minority class instances causes low diversity.

7 Conclusion

In this paper, we propose SMOTEMutliBoost learning algorithm for class imbalanced data sets. MultiBoost ensemble is combined with SMOTE oversampling to improve the prediction performance of minority as well as majority class. MultiBoost is an ensemble that combines the wagging with boosting to reduce the error due to bias and variance. Experiments with SMOTEMultiBoost on a wide collection of data sets and comparison with other learning algorithms show that our technique is able to accomplish higher G-mean, F-measure values and is dominate in ROC space. This endorses our hypotheses that combining SMOTE with MultiBoost ensemble is a good approach for dealing with class imbalance. Furthermore, SMOTEMultiBoost inherits the MultiBoost's parallel computational edge over other boosting techniques. In our future work, we will further investigate effectiveness of our approach with more exhaustive experimental setup that includes large data sets with high imbalance ratios and increased number of subcommittees and their sizes.

8 Acknowledgements

add acknowledgements here.

References

1. Chawla, Nitesh V., et al. SMOTEBoost: Improving prediction of the minority class in boosting. Knowledge Discovery in Databases: PKDD 2003. Springer Berlin Heidelberg, 2003. 107-119.
2. Chawla, Nitesh V., et al. SMOTE: Synthetic Minority Over-sampling TEchnique. Journal of Artificial Intelligence Research 16 (2002): 341-378.
3. Weiss, Gary M., and Foster Provost. "The effect of class distribution on classifier learning: an empirical study." Rutgers Univ (2001).
4. Khoshgoftaar T., Van Hulse J., Napolitano A.: Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man, and Cybernetics(2011).
5. Webb, Geoffrey I.:Multiboosting: A technique for combining boosting and wagging. Machine learning 40.2 (2000): 159-196.
6. Liu, Xu-Ying, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39.2 (2009): 539-550.
7. Bauer, Eric, and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. Machine learning 36.1-2 (1999): 105-139.
8. Y. Tang, Y.-Q. Zhang, N.V. Chawla. SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39:1 (2009) 0-288.
9. X. Hong, S. Chen, and C. Harris, A kernel-based two-class classifier for imbalanced data sets, IEEE Transactions on Neural Networks, vol. 18, no. 1, pp. 2841, 2007.
10. G. Wu and E. Chang, Kba: Kernel boundary alignment considering imbalanced data distribution, IEEE Transactions on Knowledge and Data Engineering, vol. 17, no. 6, pp. 786795, 2005.
11. N. Chawla, D. Cieslak, L. Hall, and A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Min. Knowl. Discov., vol. 17, pp. 225252, 2008.
12. Seiffert, Chris, et al. RUSBoost: A hybrid approach to alleviating class imbalance" Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on 40.1 (2010): 185-197.
13. Wang, Shuo, and Xin Yao. Diversity analysis on imbalanced data sets by using ensemble models. Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on. IEEE, 2009.
14. Seiffert, Chris, et al. "Resampling or reweighting: A comparison of boosting implementations." Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on. Vol. 1. IEEE, 2008.
15. Alcal-Fdez, Jess, et al. KEEL Data-Mining Software Tool: Data Set Repository, Integration of Algorithms and Experimental Analysis Framework. Journal of Multiple-Valued Logic and Soft Computing 17 (2011).
16. J. Demsar. Statistical comparisons of classifiers over multiple data sets. The Journal of Machine Learning Research, 7:30,2006.