

SMOTEBoost: Improving Prediction of the Minority Class in Boosting

Nitesh V. Chawla¹, Aleksandar Lazarevic²,
Lawrence O. Hall³, and Kevin W. Bowyer⁴

¹ Business Analytic Solutions, Canadian Imperial Bank of Commerce (CIBC)
BCE Place, 161 Bay Street, 11th Floor, Toronto, ON M5J 2S8, Canada
nitesh.chawla@cibc.ca

² Department of Computer Science, University of Minnesota
200 Union Street SE, Minneapolis, MN 55455, USA
aleks@cs.umn.edu

³ Department of Computer Science and Engineering, University of South Florida
ENB 118, 4202 E. Fowler Avenue, Tampa, FL 33620, USA
hall@csee.usf.edu

⁴ Department of Computer Science and Engineering
384 Fitzpatrick Hall, University of Notre Dame, IN 46556, USA
kwb@cse.nd.edu

Abstract. Many real world data mining applications involve learning from imbalanced data sets. Learning from data sets that contain very few instances of the minority (or interesting) class usually produces biased classifiers that have a higher predictive accuracy over the majority class(es), but poorer predictive accuracy over the minority class. SMOTE (Synthetic Minority Over-sampling TEchnique) is specifically designed for learning from imbalanced data sets. This paper presents a novel approach for learning from imbalanced data sets, based on a combination of the SMOTE algorithm and the boosting procedure. Unlike standard boosting where all misclassified examples are given equal weights, SMOTEBoost creates synthetic examples from the rare or minority class, thus indirectly changing the updating weights and compensating for skewed distributions. SMOTEBoost applied to several highly and moderately imbalanced data sets shows improvement in prediction performance on the minority class and overall improved *F-values*.

1 Motivation and Introduction

Rare events are events that occur very infrequently, i.e. whose frequency ranges from say 5% to less than 0.1%, depending on the application. Classification of rare events is a common problem in many domains, such as detecting fraudulent transactions, network intrusion detection, Web mining, direct marketing, and medical diagnostics. For example, in the network intrusion detection domain, the number of intrusions on the network is typically a very small fraction of the total network traffic. In medical databases, when classifying the pixels in mammogram images as cancerous or not

[1], abnormal (cancerous) pixels represent only a very small fraction of the entire image. The nature of the application requires a fairly high detection rate of the minority class and allows for a small error rate in the majority class since the cost of misclassifying a cancerous patient as non-cancerous can be very high.

In all these scenarios when the majority class typically represents 98-99% of the entire population, a trivial classifier that labels everything with the majority class can achieve high accuracy. It is apparent that for domains with imbalanced and/or skewed distributions, classification accuracy is not sufficient as a standard performance measure. ROC analysis [2] and metrics such as *precision*, *recall* and *F-value* [3, 4] have been used to understand the performance of the learning algorithm on the minority class. The prevalence of class imbalance in various scenarios has caused a surge in research dealing with the minority classes. Several approaches for dealing with imbalanced data sets were recently introduced [1, 2, 4, 9-15].

A confusion matrix as shown in Table 1 is typically used to evaluate performance of a machine learning algorithm for rare class problems. In classification problems, assuming class “C” as the minority class of the interest, and “NC” as a conjunction of all the other classes, there are four possible outcomes when detecting class “C”.

Table 1. Confusion matrix defines four possible scenarios when classifying class “C”

	Predicted Class “C”	Predicted Class “NC”
Actual class “C”	True Positives (TP)	False Negatives (FN)
Actual class “NC”	False Positives (FP)	True Negatives (TN)

From Table 1, *recall*, *precision* and *F-value* may be defined as follows:

$$Precision = TP / (TP + FP)$$

$$Recall = TP / (TP + FN)$$

$$F-value = \frac{(1 + \beta^2) \cdot Recall \cdot Precision}{\beta^2 \cdot Recall + Precision},$$

where β corresponds to relative importance of *precision* vs. *recall* and it is usually set to 1. The main focus of all learning algorithms is to improve the *recall*, without sacrificing the *precision*. However, the *recall* and *precision* goals are often conflicting and attacking them simultaneously may not work well, especially when one class is rare. The *F-value* incorporates both *precision* and *recall*, and the “goodness” of a learning algorithm for the minority class can be measured by the *F-value*. While ROC curves represent the trade-off between values of TP and FP, the *F-value* basically incorporates the relative effects/costs of *recall* and *precision* into a single number.

It is well known in machine learning that a combination of classifiers can be an effective technique for improving prediction accuracy. As one of the most popular combining techniques, boosting [5] uses adaptive sampling of instances to generate a highly accurate ensemble of classifiers whose individual global accuracy is only moderate. There has been significant interest in the recent literature for embedding cost-sensitivity in the boosting algorithm. CSB [6] and AdaCost boosting algorithms [7] update the weights of examples according to the misclassification costs. Karakou-

las and Shawe-Taylor’s ThetaBoost adjusts the margins in the presence of unequal loss functions [8]. Alternatively, Rare-Boost [4, 9] updates the weights of the examples differently for all four entries shown in Table 1.

In this paper we propose a novel approach for learning from imbalanced data sets, SMOTEBoost, that embeds SMOTE [1], a technique for countering imbalance in a dataset, in the boosting procedure. We apply SMOTE during each boosting iteration in order to create new synthetic examples from the minority class. SMOTEBoost constructs focuses on the minority class examples sampled for each boosting iteration, and constructs new examples. Experiments performed on data sets from several domains have shown that SMOTEBoost is able to achieve a higher *F-value* than SMOTE applied to a classifier, standard boosting algorithm, AdaCost [7] and first smote then boosting for each of the datasets. We also provide a *precision-recall* analysis of the approaches.

2 Synthetic Minority Oversampling Technique - SMOTE

SMOTE (Synthetic Minority Oversampling Technique) was proposed to counter the effect of having few instances of the minority class in a data set [1]. SMOTE creates synthetic instances of the minority class by operating in the “feature space” rather than the “data space”. By synthetically generating more instances of the minority class, the inductive learners, such as decision trees (e.g. C4.5 [16]) or rule-learners (e.g. RIPPER [17]), are able to broaden their decision regions for the minority class. We deal with nominal (or discrete) and continuous attributes differently in SMOTE. In the nearest neighbor computations for the minority classes we use Euclidean distance for the continuous features and the Value Distance Metric (with the Euclidean assumption) for the nominal features [1, 18, 19]. The new synthetic minority samples are created as follows:

- For the continuous features
 - Take the difference between a feature vector (minority class sample) and one of its k nearest neighbors (minority class samples).
 - Multiply this difference by a random number between 0 and 1.
 - Add this difference to the feature value of the original feature vector, thus creating a new feature vector
- For the nominal features
 - Take majority vote between the feature vector under consideration and its k nearest neighbors for the nominal feature value. In the case of a tie, choose at random.
 - Assign that value to the new synthetic minority class sample.

Using this technique, a new minority class sample is created in the neighborhood of the minority class sample under consideration. The neighbors are proportionately utilized depending upon the amount of SMOTE. Hence, using SMOTE, more general regions are learned for the minority class, allowing the classifiers to better predict

unseen examples belonging to the minority class. A combination of SMOTE and under-sampling creates potentially optimal classifiers as a majority of points from the SMOTE and under-sampling combination lie on the convex hull of the family of ROC curves [1, 2].

3 SMOTEBoost Algorithm

In this paper, we propose a SMOTEBoost algorithm that combines the Synthetic Minority Oversampling Technique (SMOTE) and the standard boosting procedure. We want to utilize SMOTE for improving the prediction of the minority classes, and we want to utilize boosting to not sacrifice accuracy over the entire data set. Our goal is to better model the minority class in the data set, by providing the learner not only with the minority class instances that were misclassified in previous boosting iterations, but also with a broader representation of those instances. We want to improve the overall accuracy of the ensemble by focusing on the difficult minority (positive) class cases, as we want to model this class better, with minimal accuracy degradation for the majority class. The goal is to improve our True Positives (TP).

The standard boosting procedure gives equal weights to all misclassified examples. Since boosting algorithm samples from a pool of data that predominantly consists of the majority class, subsequent samplings of the training set may still be skewed towards the majority class. Although boosting reduces the variance and the bias in the final ensemble, it might not be as effective for data sets with skewed class distributions.. Boosting algorithm (Adaboost) treats both kinds of errors (FP and FN) in a similar fashion, and therefore sampling distributions in subsequent boosting iterations could have a larger composition of majority class cases.

Our goal is to reduce the bias inherent in the learning procedure due to the class imbalance. Introducing SMOTE in each round of boosting will enable each learner to learn from more of the minority class cases, thus learning broader decision regions for the minority class. We only SMOTE for the minority class examples in the distribution D_t at the iteration t . This has an implicit effect of increasing the sampling weights of minority class cases, as new examples are created in D_t . The synthetically created minority class cases are discarded after learning a classifier at iteration t . That is, they are not added to the original training set, and new examples are constructed in each iteration t , by sampling from D_t . The error-estimate after each boosting iteration is on the original training set. Thus, we try to maximize the margin for the skewed class dataset, by adding new minority class cases before learning a classifier in a boosting iteration. We also conjecture that introducing the SMOTE procedure also increases the diversity amongst the classifiers in the ensemble, as in each iteration we produce a different set of synthetic examples, and therefore different classifiers. The amount of SMOTE is a parameter that can vary for each data set. It will be useful to know a priori the amount of SMOTE to be introduced for each data set. We believe that utilizing a validation set to set the amount of SMOTE before the boosting iterations can be useful.

The combination of SMOTE and the boosting procedure that we present here is a variant of the AdaBoost.M2 procedure [5]. The proposed SMOTEBoost algorithm, shown in Fig. 1, proceeds in a series of T rounds. In every round a weak learning algorithm is called and presented with a different distribution D_t , altered by emphasizing particular training examples. The distribution is updated to give wrong classifications higher weights than correct classifications. Unlike standard boosting, where the distribution D_t is updated uniformly for examples from both the majority and minority classes, in the SMOTEBoost technique the distribution D_t is updated such that the examples from the minority class are oversampled by creating synthetic minority class examples (See Line 1, Fig. 1). The entire weighted training set is given to the weak learner to compute the weak hypothesis h_t . At the end, the different hypotheses are combined into a final hypothesis h_{fn} .

- Given: Set $S = \{(x_1, y_1), \dots, (x_m, y_m)\}$ $x_i \in X$, with labels $y_i \in Y = \{1, \dots, C\}$, where C_p , ($C_p < C$) corresponds to a minority (positive) class.
- Let $B = \{(i, y) : i = 1, \dots, m, y \neq y_i\}$
- Initialize the distribution D_1 over the examples, such that $D_1(i) = 1/m$.
- For $t = 1, 2, 3, 4, \dots, T$
 1. Modify distribution D_t by creating N synthetic examples from minority class C_p using the SMOTE algorithm
 2. Train a weak learner using distribution D_t
 3. Compute weak hypothesis $h_t: X \times Y \rightarrow [0, 1]$
 4. Compute the pseudo-loss of hypothesis h_t :

$$\varepsilon_t = \sum_{(i,y) \in B} D_t(i,y)(1 - h_t(x_i, y) + h_t(x_i, y))$$
 5. Set $\beta_t = \varepsilon_t / (1 - \varepsilon_t)$ and $w_t = (1/2) \cdot (1 - h_t(x_i, y) + h_t(x_i, y_i))$
 6. Update D_{t+1} : $D_{t+1}(i, y) = (D_t(i, y) / Z_t) \cdot \beta_t^{w_t}$
where Z_t is a normalization constant chosen such that D_{t+1} is a distribution.
- Output the final hypothesis: $h_{fn} = \arg \max_{y \in Y} \sum_{t=1}^T (\log \frac{1}{\beta_t}) \cdot h_t(x, y)$

Fig. 1. The SMOTEBoost algorithm

We used RIPPER [17], a learning algorithm that builds a set of rules for identifying the classes while minimizing the amount of error, as the classifier in our SMOTEBoost experiments. RIPPER is a rule-learning algorithm based on the separate-and-conquer strategy. We applied SMOTE with different values for the parameter N that specifies the amount of synthetically generated examples.

4 Experiments

Our experiments were performed on the four data sets summarized in Table 2. For all data sets, except for the KDD Cup-99 intrusion detection data set [20, 21], the reported (averaged) values for *recall*, *precision* and *F-value* were obtained by performing 10-fold cross-validation. For the KDDCup-99 data set however, the separate intrusion detection test set was used to evaluate the performance of proposed algorithms. Since the original training and test data sets have totally different distributions due to novel intrusions introduced in the test data, for the purposes of this paper, we modified the data sets in order to have similar distributions in the training and test data. Therefore, we first merged the original training and test data sets and then sampled 69,980 network connections from this merged data set in order to reduce the size of the data set. The sampling was performed only from majority classes (normal background traffic and the DoS attack category), while other classes (Probe, U2R, R2L) remained intact. Finally, the new train and test data sets used in our experiments were obtained by randomly splitting the sampled data set into equal size subsets. The distribution of network connections in the new test data set is given in Table 2. Unlike the KDDCup-99 intrusion data set that has a mixture of both nominal and continuous features, the remaining data sets (mammography [1], satimage [22], phoneme [23]) have all continuous features. For the satimage data set we chose the smallest class as the minority class and collapsed the remaining classes into one class as was done in [24]. This procedure gave us a skewed 2-class dataset, with 5809 majority class examples and 626 minority class examples.

Table 2. Summary of data sets used in experiments

Data set	Number of majority class instances			Number of minority class instances		Number of classes
	DoS	Probe	Normal	U2R	R2L	
KDDCup-99 Intrusion	13027	2445	17400	136	1982	5
Mammography	10923			260		2
Satimage	5809			626		2
Phoneme	3818			1586		2

When experimenting with SMOTE and the SMOTEBoost algorithm, different values for the SMOTE parameter N , ranging between 100 and 500, were used for the minority classes. Since the KDD Cup'99 data set has two minority classes U2R and R2L that are not equally represented in the data set, different combinations of SMOTE parameters were investigated for these two minority classes (values 100, 300, and 500 were used for the U2R class while the value 100 was used for the R2L class). The values of the SMOTE parameters for U2R class were higher than the SMOTE parameter values for R2L class, since the U2R class is rarer than the R2L class in KDD-Cup 1999 data set (R2L has a larger number of examples). Our experimental results showed that the higher values of SMOTE parameters for the R2L

class could lead to over-fitting and decreasing the prediction performance on that class (since SMOTEBoost achieved only minor improvements for the R2L class, these results are not reported here due to space limitations).

The experimental results for all four data sets are presented in Tables 3 to 6 and in Figures 2 to 4. It is important to note that these tables report only the prediction performance for the minority classes from four data sets, since prediction of the majority class was not of interest in this study. Moreover, precision captures the FP's introduced in the classification. So, F-value includes the estimate for majority class examples wrongly classified. Due to space limitations, the figures with *precision* and *recall* trends over the boosting iterations, along with the *F-value* trends for the representative SMOTE parameter were not shown for the R2L class from KDDCup'99 data as well as for the satimage data set. In addition, the left and the right parts of the reported Figures do not have the same scale due to the fact that the range of changes in *recall* and *precision* shown in the same graph is much larger than the change of the *F-value*.

Table 3. Final values for *recall*, *precision* and *F-value* for minority U2R class when proposed methods are applied on KDDCup-99 intrusion data set. (N_{u2r} corresponds to the SMOTE parameter for U2R class, while N_{r2l} corresponds to the SMOTE parameter for R2L class)

<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>	<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
Standard RIPPER		57.35	84.78	68.42	Standard Boosting		80.15	90.083	84.83
	N_{u2r} N_{r2l}	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>		N_{u2r} N_{r2l}	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
SMOTE	100 100	80.15	88.62	84.17	SMOTE -Boost	100 100	84.2	93.9	88.8
	300 100	74.26	92.66	82.58		300 100	87.5	88.8	88.15
	500 100	68.38	86.11	71.32		500 100	84.6	92.0	88.1
First SMOTE then Boost	N_{u2r} N_{r2l}	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>	Ada- Cost	<i>Cost factor</i>	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
	100 100	81.6	90.92	86.01		$c = 2$	83.1	96.6	89.3
	300 100	82.5	89.30	85.77		$c = 5$	83.45	95.29	88.98
	500 100	82.9	89.12	85.90					

Table 4. Final values for *recall*, *precision* and *F-value* for minority class when proposed methods are applied on *mammography* data set

<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>	<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
Standard RIPPER		48.12	74.68	58.11	Standard Boosting		59.09	77.05	66.89
SMOTE	$N = 100$	58.04	64.96	61.31	SMOTE -Boost	$N = 100$	61.73	76.59	68.36
	$N = 200$	62.16	60.53	60.45		$N = 200$	62.63	74.54	68.07
	$N = 300$	62.55	56.57	58.41		$N = 300$	64.16	69.92	66.92
	$N = 500$	64.51	53.81	58.68		$N = 500$	61.37	70.41	65.58
First SMOTE then Boost	$N = 100$	60.22	76.16	67.25	Ada- Cost	<i>Cost factor</i>	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
	$N = 200$	62.61	72.10	67.02		2	59.83	69.07	63.01
	$N = 300$	63.92	70.26	66.94		5	68.45	55.12	59.36
	$N = 500$	64.14	69.80	66.85					

Table 5. Final values for *recall*, *precision* and *F-value* for minority class when proposed methods are applied on *Satimage* data set

<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>	<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
Standard RIPPER		47.43	67.92	55.50	Standard Boosting		58.74	80.12	67.78
SMOTE	<i>N</i> = 100	65.17	55.88	59.97	SMOTE -Boost	<i>N</i> = 100	63.88	77.71	70.12
	<i>N</i> = 200	74.89	48.08	58.26		<i>N</i> = 200	65.35	73.17	69.04
	<i>N</i> = 300	76.32	47.17	57.72		<i>N</i> = 300	67.87	72.68	70.19
	<i>N</i> = 500	77.96	44.51	56.54		<i>N</i> = 500	67.73	69.5	68.6
First SMOTE then Boost	<i>N</i> = 100	64.69	72.53	68.65	Ada- Cost	<i>Cost factor</i>	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
	<i>N</i> = 200	69.23	67.10	68.15		2	64.85	54.58	58.2
	<i>N</i> = 300	67.25	69.92	68.56		5	60.85	56.01	57.6
	<i>N</i> = 500	67.84	68.02	67.93					

Table 6. Final values for *recall*, *precision* and *F-value* for minority class when proposed methods are applied on *phoneme* data set

<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>	<i>Method</i>		<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
Standard RIPPER		62.28	69.13	65.15	Standard Boosting		76.1	77.07	76.55
SMOTE	<i>N</i> = 100	82.18	59.91	68.89	SMOTE -Boost	<i>N</i> = 100	81.86	73.66	77.37
	<i>N</i> = 200	85.88	58.51	69.59		<i>N</i> = 200	84.86	76.47	76.47
	<i>N</i> = 300	89.79	56.15	69.04		<i>N</i> = 300	86	66.76	75.16
	<i>N</i> = 500	94.2	50.22	65.49		<i>N</i> = 500	88.46	65.16	75.04
First SMOTE then Boost	<i>N</i> = 100	82.05	72.34	76.89	Ada- Cost	<i>Cost factor</i>	<i>Recall</i>	<i>Precision</i>	<i>F-value</i>
	<i>N</i> = 200	85.25	68.97	76.25		2	76.83	75.71	75.99
	<i>N</i> = 300	87.37	66.38	75.44		5	85.05	68.71	75.9
	<i>N</i> = 500	89.21	64.73	75.03					

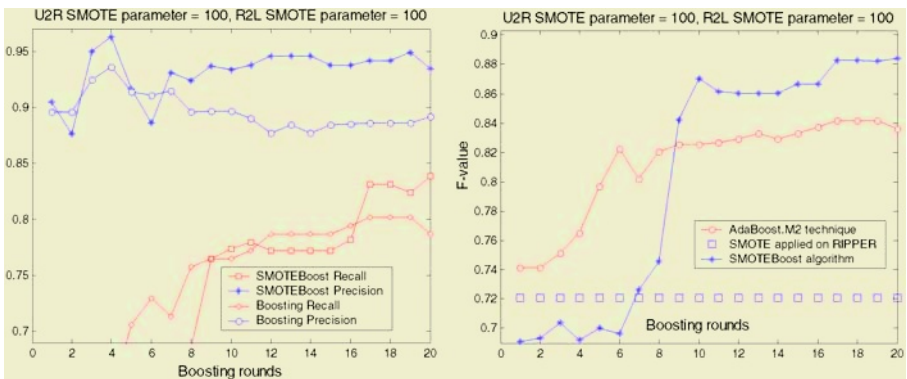


Fig. 2. Precision, Recall, and F-values for the minority U2R class when the SMOTEBoost algorithm is applied on the KDDCup 1999 data set

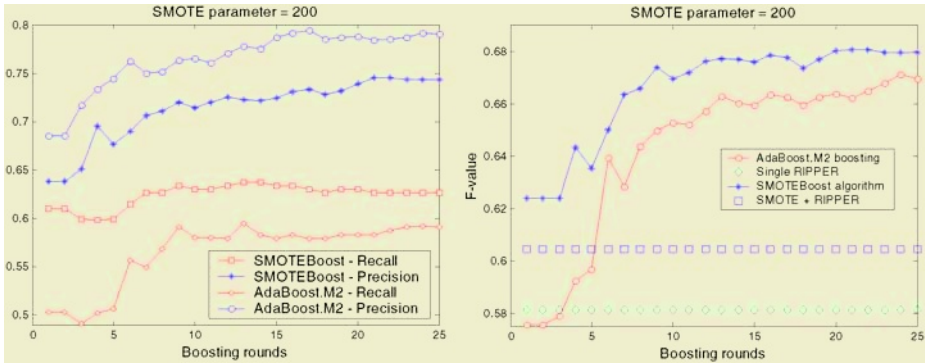


Fig. 3. Precision, Recall, and F -values for the minority class when the SMOTEBoost algorithm is applied on the Mammography data set

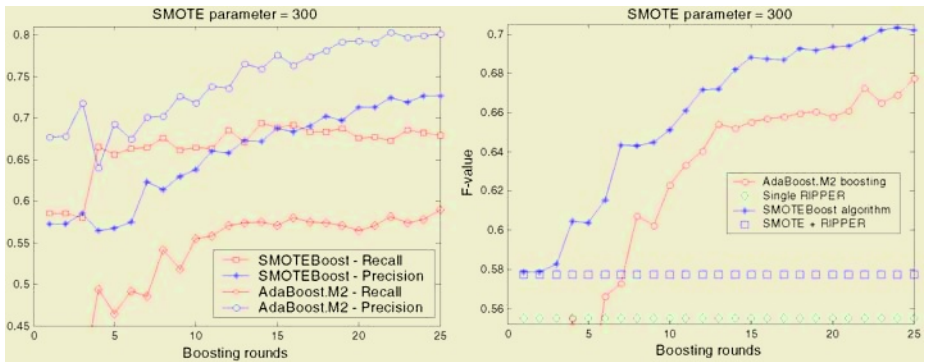


Fig. 4. Precision, Recall, and F -values for the minority class when the SMOTEBoost algorithm is applied on the Satimage data set

Analyzing Figures 2 to 4 and Tables 3 to 6, it is apparent that SMOTEBoost achieved higher F -values than the other presented methods including standard boosting, AdaCost, SMOTE with the RIPPER classifier and the standard RIPPER classifier, although the improvement varied with different data sets. We have also compared SMOTEBoost to the procedure “First SMOTE, then Boost” when we first apply SMOTE and then perform boosting in two separate steps. It is SMOTEBoost’s apparent improvement in *recall*, while not causing a significant degradation in *precision* that improves the over-all F -value. Tables 3 to 6 include the *precision*, *recall*, and F -value for the various methods at different amounts of SMOTE (best values are given in bold). These reported values indicate that SMOTE applied with the RIPPER classifier has the effect of improving the *recall* of the minority class due to improved coverage of the minority class examples, while at the same time SMOTE causes a decrease in *precision* due to an increased number of false positive examples. Thus, SMOTE is more targeted to the minority class than standard boosting or RIPPER. On the other hand, standard boosting is able to improve both the *recall* and *precision* of a single classifier, since it gives all errors equal weights. SMOTE embedded within the

boosting procedure additionally improved the *recall* achieved by the boosting procedure, and did not cause a significant degradation in *precision*, thus increasing the *F-value*. SMOTE as a part of SMOTEBoost allows the learners to broaden the minority class scope, while boosting on the other hand aims at reducing the number of false positives and false negatives.

Tables 3 to 6 show the precision, recall, and F-values achieved by varying the amount of SMOTE for each of the minority classes for all four data sets used in our experiments. We report the aggregated result of 25 boosting iterations in the tables. The improvement was generally higher for the data sets where the skew among the classes was also higher. Comparing SMOTEBoost and AdaBoost.M1, for the KDD-Cup'99 data set, the (relative) improvement in *F-value* for the U2R class (~4%) was drastically higher than for the R2L class (0.61%). The U2R class was significantly less represented in the data set than the R2L class (the number of U2R examples was around 15 times smaller than the number of examples from the R2L class). In addition, the (relative) improvements in *F-value* for the mammography (2.2%) and satimage (3.4%) data sets were better than for the phoneme data set (1.4%), which had much less imbalanced classes. For phoneme data, boosting and SMOTE Boost were comparable to each other, while for higher values of the SMOTE parameter N , boosting was even better than SMOTEBoost. In this data set the number of majority class examples is only twice the number of minority class examples, and increasing the SMOTE parameter N to values larger than 200 causes the minority class to become the majority. Hence, the classifiers in the SMOTEBoost ensemble will now tend to over-learn the minority class, causing a higher degradation in *precision* for the minority class and therefore a reduction in *F-value*.

We have also shown that SMOTEBoost gives higher *F-values* than the AdaCost algorithm [7]. The cost-adjustment functions from the AdaCost algorithm were chosen as follows: $\beta_- = 0.5 * c + 0.5$ and $\beta_+ = -0.5 * c + 0.5$, where β_- and β_+ are the functions for mislabeled and correctly labeled examples, respectively. AdaCost causes a greater sampling from the minority class examples due to the β function in the boosting distribution. This implicitly has an effect of over-sampling with replication. SMOTEBoost on the other hand constructs new examples at each round of boosting, thus avoiding overfitting and achieving higher minority class classification performances than AdaCost. Although AdaCost improves the *recall* over AdaBoost, it significantly reduces the precision thus causing a reduction in *F-value*. It is also interesting to note that SMOTEBoost achieves better *F-values* than the procedure "First SMOTE, then Boost" since in every boosting iteration new examples from minority class are generated, and thus, more diverse classifiers are created in the boosting ensemble. Finally, SMOTEBoost particularly focuses on the examples selected in the D_t , which are potentially misclassified or are on the classification boundaries.

5 Conclusions

A novel approach for learning from imbalanced data sets is presented. The proposed SMOTEBoost algorithm is based on the integration of the SMOTE algorithm within

the standard boosting procedure. Experimental results from several imbalanced data sets indicate that the proposed SMOTEBoost algorithm can result in better prediction of minority classes than AdaBoost, AdaCost, “First SMOTE then Boost” procedure and a single classifier. Data sets used in our experiments contained different degrees of imbalance and different sizes, thus providing a diverse test bed.

The SMOTEBoost algorithm successfully utilizes the benefits from both boosting and the SMOTE algorithm. While boosting improves the predictive accuracy of classifiers by focusing on difficult examples that belong to all the classes, the SMOTE algorithm improves the performance of a classifier only on the minority class examples. Therefore, the embedded SMOTE algorithm forces the boosting algorithm to focus more on difficult examples that belong to the minority class than to the majority class. SMOTEBoost implicitly increases the weights of the misclassified minority class instances (false negatives) in the distribution D_i by increasing the number of minority class instances using the SMOTE algorithm. Therefore, in the subsequent boosting iterations SMOTEBoost is able to create broader decision regions for the minority class compared to the standard boosting. We conclude that SMOTEBoost can construct an ensemble of diverse classifiers and reduce the bias of the classifiers. SMOTEBoost combines the power of SMOTE in vastly improving the *recall* with the power of boosting in improving the *precision*. The overall effect is a better *F-value*.

Our experiments have also shown that SMOTEBoost is able to achieve higher F-values than AdaCost, due to SMOTE's ability to improve the coverage of the minority class when compared to the indirect effect of oversampling with replication in AdaCost.

Although the experiments have provided evidence that the proposed method can be successful for learning from imbalanced data sets, future work is needed to address its possible drawbacks. First, automatic determination of the amount of SMOTE will not only be useful when deploying SMOTE as an independent approach, but also for combining SMOTE and boosting. Second, our future work will also focus on investigating the effect of mislabeling noise on the performance of SMOTEBoost, since it is known that boosting does not perform well in the presence of noise.

Acknowledgments

This research was partially supported by the US Department of Energy through the San-dia National Labs ASCI VIEWS Data Discovery Program contract number DE-AC04-76DO00789 and by Army High Performance Computing Research Center contract number DAAD19-01-2-0014. The content of the work does not necessarily reflect the position or policy of the government and no official endorsement should be inferred. Access to computing facilities was provided by AHPCRC and the Minnesota Supercomputing Institute. We also thank Philip Kegelmeyer for his helpful feedback. We would also like to thank anonymous reviewers for their useful comments on the paper.

References

1. N. V. Chawla, K.W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic Minority Over-Sampling Technique, *Journal of Artificial Intelligence Research*, vol. 16, 321-357, 2002.
2. F. Provost, T. Fawcett, Robust Classification for Imprecise Environments, *Machine Learning*, vol. 42/3, pp. 203-231, 2001.
3. M. Buckland, F. Gey, The Relationship Between Recall and Precision, *Journal of the American Society for Information Science*, 45(1):12--19, 1994.
4. M. Joshi, V. Kumar, R. Agarwal, Evaluating Boosting Algorithms to Classify Rare Classes: Comparison and Improvements, *First IEEE International Conference on Data Mining*, San Jose, CA, 2001.
5. Y. Freund, R. Schapire, Experiments with a New Boosting Algorithm, *Proceedings of the 13th International Conference on Machine Learning*, 325-332, 1996.
6. K. Ting, A Comparative Study of Cost-Sensitive Boosting Algorithms, *Proceedings of 17th International Conference on Machine Learning*, 983-990, Stanford, CA, 2000.
7. W. Fan, S. Stolfo, J. Zhang, P. Chan, AdaCost: Misclassification Cost-Sensitive Boosting, *Proc. of 16th International Conference on Machine Learning*, Slovenia, 1999.
8. G. Karakoulas, J. Shawe-Taylor, Optimizing Classifiers for Imbalanced Training Sets. In Kearns, M., Solla, S., and Cohn, D., editors. *Advances in Neural Information Processing Systems 11*, MIT Press, 1999.
9. M.Joshi, R. Agarwal, V. Kumar, Predicting Rare Classes: Can Boosting Make Any Weak Learner Strong?, *Proceedings of Eighth ACM Conference ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Edmonton, Canada, 2002.
10. M.Joshi, R. Agarwal, PNrule: A New Framework for Learning Classifier Models in Data Mining (A Case-study in Network Intrusion Detection), *First SIAM Conference on Data Mining*, Chicago, IL, 2001.
11. P. Chan, S. Stolfo, Towards Scalable Learning with Non-uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection, *Proceedings of Fourth ACM Conference ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 164-168, New York, NY, 1998.
12. M. Kubat, R. Holte, and S. Matwin, Machine Learning for the Detection of Oil Spills in Satellite Radar Images, *Machine Learning*, vol. 30, pp. 195-215, 1998.
13. N. Japkowicz, The Class Imbalance Problem: Significance and Strategies, *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning*, Las Vegas, Nevada, 2000.
14. D. Lewis and J. Catlett, Heterogeneous Uncertainty Sampling for Supervised Learning, *Proceedings of the Eleventh International Conference of Machine Learning*, San Francisco, CA, 148-156, 1994.
15. C. Ling and C. Li, Data Mining for Direct Marketing Problems and Solutions, *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, New York, NY, 1998.
16. J. Quinlan, *C4.5: Programs for Machine Learning*. San Mateo, CA: Morgan Kaufman, 1992.
17. W. Cohen, Fast Effective Rule Induction, *Proceedings of the 12th International Conference on Machine Learning*, Lake Tahoe, CA, 115-123, 1995.

18. C. Stanfill, D. Waltz, Toward Memory-based Reasoning, *Communications of the ACM*, vol. 29, no. 12, pp. 1213-1228, 1986.
19. S. Cost, S. Salzberg, A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features, *Machine Learning*, vol. 10, no. 1, pp. 57-78, 1993.
20. KDD-Cup 1999 Task Description, <http://kdd.ics.uci.edu/databases/kddcup99/task.html>
21. R. Lippmann, D. Fried, I. Graf, J. Haines, K. Kendall, D. McClung, D. Weber, S. Webster, D. Wyschogrod, R. Cunningham, M. Zissman, Evaluating Intrusion Detection Systems: The 1998 DARPA Off-line Intrusion Detection Evaluation, *Proceedings DARPA Information Survivability Conference and Exposition (DISCEX) 2000*, Vol 2, pp. 12-26, IEEE Computer Society Press, Los Alamitos, CA, 2000.
22. C. Blake and C. Merz, UCI Repository of Machine Learning Databases <http://www.ics.uci.edu/~mllearn/~MLRepository.html>, Department of Information and Computer Sciences, University of California, Irvine, 1998.
23. F. Provost, T. Fawcett, R. Kohavi, The Case Against Accuracy Estimation for Comparing Induction Algorithms, *Proceedings of 15th International Conference on Machine Learning*, 445-453, Madison, WI, 1998.
24. ELENA project, <ftp.dice.ucl.ac.be> in directory pub/neural-nets/ELENA/databases