Experimental Perspectives on Learning from Imbalanced Data

Jason Van Hulse JVANHULSE@GMAIL.COM
Taghi M. Khoshgoftaar TAGHI@CSE.FAU.EDU
Amri Napolitano ANAPOLI1@FAU.EDU
Florida Atlantic University, Boca Raton, FL 33431 USA

Abstract

We present a comprehensive suite of experimentation on the subject of learning from imbalanced data. When classes are imbalanced, many learning algorithms can suffer from the perspective of reduced performance. Can data sampling be used to improve the performance of learners built from imbalanced data? Is the effectiveness of sampling related to the type of learner? Do the results change if the objective is to optimize different performance metrics? We address these and other issues in this work, showing that sampling in many cases will improve classifier performance.

1. Introduction

In many real-world classification domains, the vast majority of examples are from one of the classes. In binary classification, it is typically the minority (positive) class that the practitioner is interested in. Imbalance in the class distribution often causes machine learning algorithms to perform poorly on the minority class. In addition, the cost of misclassifying the minority class is usually much higher than the cost of other misclassifications. Therefore a natural question in machine learning research is how to improve upon the performance of classifiers when one class is relatively rare?

A common solution is to sample the data, either randomly or intelligently, to obtain an altered class distribution. Numerous techniques have been proposed (Section 3), although it is unclear which techniques work best. Some researchers have experimentally evaluated the use of sampling when learning from imbalanced data (e.g., (Drummond & Holte, 2003), (Kubat & Matwin, 1997), (Maloof, 2003), (Japkowicz, 2000), (Weiss & Provost, 2003), (Chawla et al., 2002), (Han et al., 2005), (Monard & Batista, 2002)). The experimentation performed in this study, however, is more comprehensive than related work, which typically utilize one or two learners and a couple of datasets. In this work, we consider 35 different benchmark datasets (Section 2), seven sampling techniques (Section 3), and 11 commonly-used learning algorithms (Section 4). As explained in Section 5, a total of 1,232,000 classifiers were constructed in these experiments. In addition, we perform statistical analysis using analysis of variance (ANOVA) models to understand the statistical significance of the results. These components make our work very comprehensive, and dramatically increase the reliability of our conclusions. We strongly advocate robust, statistically valid, and reliable empirical work to understand the relative strengths and weaknesses of different techniques in real-world applications.

2. Experimental Datasets

The 35 datasets utilized in our empirical study are listed in Table 1. The percentage of minority examples varies from 1.33% (highly imbalanced) to almost 35% (only slightly imbalanced). The datasets also come from a wide variety of application domains, and 19 are from the UCI repository (Blake & Merz, 1998). The Mammography dataset was generously provided by Dr. Nitesh Chawla (Chawla et al., 2002). Fifteen datasets (some of which are proprietary) are from the domain of software engineering measurements. We have also considered datasets with diversity in the number of attributes, and datasets with both continuous and categorical attributes. The smallest dataset had 214 total examples (Glass-3), while the two largest datasets each contain 20,000 observations. Note that all datasets have, or were transformed to have, a binary class. We only consider binary classification problems in this work.

3. Sampling Techniques

This section provides a brief overview of the seven sampling techniques considered in this work: random

Appearing in *Proceedings of the 24th International Conference on Machine Learning*, Corvallis, OR, 2007.

Table 1. Empirical Datasets

Name	# minority	% minority	# attr.
SP3	47	1.33%	43
SP4	92	2.31%	43
mammography	260	2.32%	7
nursery-3	328	2.53%	9
solar-flare-f	51	3.67%	13
letter-a	789	3.95%	17
car-3	69	3.99%	7
SP2	189	4.75%	43
cccs-12	16	5.67%	9
SP1	229	6.28%	43
pc1	76	6.87%	16
mw1	31	7.69%	16
glass-3	17	7.94%	10
kc3	43	9.39%	16
cm1	48	9.50%	16
cccs-8	27	9.57%	9
pendigits-5	1055	9.60%	17
satimage-4	626	9.73%	37
optdigits-8	554	9.86%	65
e-coli-4	35	10.42%	8
segment-5	330	14.29%	20
kc1	325	15.42%	16
jm1	1687	19.06%	16
letter-vowel	3878	19.39%	17
cccs-4	55	19.50%	9
kc2	106	20.38%	16
Contra-2	333	22.61%	10
SpliceJunc2	768	24.08%	61
vehicle-1	212	25.06%	19
haberman	81	26.47%	4
yeast-2	429	28.91%	9
phoneme	1586	29.35%	6
cccs-2	83	29.43%	9
german-credit	300	30.00%	21
pima-diabetes	268	34.90%	9

undersampling (RUS), random oversampling (ROS), one-sided selection (OSS), cluster-based oversampling (CBOS), Wilson's editing (WE), SMOTE (SM), and borderline-SMOTE (BSM). RUS, ROS, WE, SM, and BSM require a parameter value to be set, so when it is important to specify this value, we often use notation such as ROS300, which denotes random oversampling with the parameter 300 (the meanings of the parameters in the context of each sampling techniques are explained below).

The two most common preprocessing techniques are random *minority oversampling* (ROS) and random *majority undersampling* (RUS). In ROS, instances of the minority class are randomly duplicated. In RUS, instances of the majority class are randomly discarded from the dataset.

In one of the earliest attempts to improve upon the performance of random resampling, Kubat and Matwin (Kubat & Matwin, 1997) proposed a technique called *one-sided selection* (OSS). One-sided selection attempts to intelligently undersample the majority class by removing majority class examples that are considered either redundant or 'noisy.'

Wilson's editing (Barandela et al., 2004) (WE) uses the kNN technique with $k = 3$ to classify each example in the training set using all the remaining examples, and removes those majority class examples that are misclassified. Barandela et al. also propose a modified distance calculation, which causes an example to be biased more towards being identified with positive examples than negative ones.

Chawla et al. (Chawla et al., 2002) proposed an intelligent oversampling method called Synthetic Minority Oversampling Technique (SMOTE). SMOTE (SM) adds new, artificial minority examples by extrapolating between preexisting minority instances rather than simply duplicating original examples. The technique first finds the k nearest neighbors of the minority class for each minority example (the paper recommends $k = 5$). The artificial examples are then generated in the direction of some or all of the nearest neighbors, depending on the amount of oversampling desired.

Han et al. presented a modification of Chawla et al.'s SMOTE technique which they call *borderline-SMOTE* (Han et al., 2005) (BSM). BSM selects minority examples which are considered to be on the border of the minority decision region in the feature-space and only performs SMOTE to oversample those instances, rather than oversampling them all or a random subset.

Cluster-based oversampling (Jo & Japkowicz, 2004) (CBOS) attempts to even out the between-class imbalance as well as the within-class imbalance. There may be subsets of the examples of one class that are isolated in the feature-space from other examples of the same class, creating a within-class imbalance. Small subsets of isolated examples are called *small disjuncts*. Small disjuncts often cause degraded classifier performance, and CBOS aims to eliminate them without removing data.

RUS was performed at 5%, 10%, 25%, 50%, 75%, and 90% of the majority class. ROS, SM, and BSM were performed with oversampling rates 50%, 100%, 200%, 300%, 500%, 750%, and 1000%. When performing Wilson's editing, we utilized both the weighted and unweighted (standard Euclidean) versions, and denote them WE-W and WE-E. A total of 31 combinations of sampling technique plus parameters were utilized. In addition, we built a classifier with no sampling, which we denote 'NONE'. All of these sampling techniques were implemented in Java in the framework of the WEKA machine learning tool (Witten & Frank, 2005).

4. Learners

This section provides brief descriptions of the 11 classification algorithms along with an explanation of the parameters used in our experiments. These classifiers were considered since they are commonly-used in the machine learning community and in research on class imbalance.

All learners were built using WEKA, and changes to default parameter values were done only when experimentation showed a general improvement in the classifier performance across all datasets based on preliminary analysis.

Two different versions of the *C4.5* (Quinlan, 1993) decision tree learner (denoted C4.5D and C4.5N) were constructed using J48 in WEKA. C4.5D uses the default WEKA parameter settings, while C4.5N uses no pruning and Laplace smoothing (Weiss & Provost, 2003). Two different *K nearest neighbors* classifiers (denoted IBk in WEKA) were constructed, using $k = 2$ and $k = 5$, and denoted 2NN and 5NN. The 'distanceWeighting' parameter was set to 'Weight by 1/distance' to use inverse distance weighting in determining how to classify an instance. For a *Naive Bayes* (NB) classifier, the parameters were left at the default values.

Two parameters were changed from the default values for the *Multilayer perceptrons* (MLP) learner. The 'hiddenLayers' parameter was changed to '3' to define a network with one hidden layer containing three nodes, and the 'validationSetSize' parameter was changed to '10' to cause the classifier to leave 10% of the training data aside to be used as a validation set to determine when to stop the iterative training process. *Radial basis function networks* (RBF) are another type of artificial neural network. The only parameter change for RBF (called 'RBF Network' in WEKA) was to set 'numClusters' to 10.

RIPPER (Repeated Incremental Pruning to Produce Error Reduction) is a rule-based learner. JRip is the implementation of RIPPER in WEKA, and the default parameters were used in all experiments. The *logistic regression* learner is denoted LR, and no changes to the default parameter values for this learner were made.

The *random forest* (RF) classifier (Breiman, 2001) utilizes bagging and the 'random subspace method' to construct randomized decision trees. The outputs of ensembles of these randomized, unpruned decision trees are combined to produce the ultimate prediction. No changes to the default parameters were made in our experiments. The *support vector machine* learner is called SMO in WEKA and denoted SVM in this study. For our experiments, the complexity constant 'c' was changed from 1.0 to 5.0, and the 'buildLogisticModels' parameter, which allows proper probability estimates to be obtained, was set to 'true' (Witten & Frank, 2005). In particular, the SVM learner used a linear kernel.

5. Experimental Design

The design of our experiments can be summarized as follows. For each of the 35 datasets, 20 different runs of five-fold cross validation (CV) were executed. For each iteration of CV, the training dataset consisted of four folds, and the remaining fold served as a test dataset. Each of the 31 sampling techniques (and also no sampling) were applied to the training data, 11 different learners were constructed from the transformed dataset, and each of the learners was evaluated on the test dataset (based on CV).

Table 2. Sampling, $\pi < 10\%$, SVM

Level	AUC	HSD	Level	G	HSD
ROS1000	0.898	A	RUS5	82.24	A
RUS5	0.897	AB	CBOS	80.36	AB
SM1000	0.890	AB	ROS1000	76.49	CB
BSM1000	0.886	AB	SM1000	75.17	CB
CBOS	0.872	B	BSM1000	71.93	C
WE-W	0.821	C	OSS	51.81	D
OSS	0.818	C	WE-W	45.28	E
NONE	0.809	C	NONE	41.75	E
Level	**AUC**	**HSD**	**Level**	**G**	**HSD**
ROS1000	0.861	A	ROS1000	78.156	A
SM300	0.860	A	SM1000	78.017	A
BSM300	0.856	A	RUS5	76.431	AB
RUS25	0.849	AB	BSM1000	75.851	AB
CBOS	0.830	CB	CBOS	73.173	B
WE-W	0.828	C	WE-W	51.725	C
OSS	0.814	CD	OSS	45.977	D
NONE	0.798	D	NONE	42.505	D

Table 3. Sampling, $\pi < 10\%$, RF

Level	AUC	HSD	Level	G	HSD
RUS5	0.892	A	RUS5	83.08	A
SM1000	0.865	B	SM1000	64.16	B
BSM1000	0.859	BC	BSM1000	60.17	BC
ROS300	0.847	BCD	ROS1000	59.47	BC
WE-W	0.842	BCD	CBOS	57.20	CD
NONE	0.837	CD	OSS	56.90	CD
CBOS	0.825	DE	WE-E	51.69	DE
OSS	0.810	E	NONE	49.08	E
Level	**AUC**	**HSD**	**Level**	**G**	**HSD**
RUS10	0.862	A	RUS10	79.16	A
SM750	0.857	AB	SM1000	70.54	B
BSM1000	0.852	AB	BSM1000	68.80	BC
WE-W	0.846	AB	WE-W	65.29	CD
ROS200	0.844	ABC	ROS300	64.33	DE
OSS	0.839	BC	OSS	61.65	DEF
NONE	0.836	BC	CBOS	61.24	EF
CBOS	0.825	C	NONE	59.76	F

In total, 20 five-fold CV runs times 35 datasets is 3500 different training datasets. 31 sampling techniques, plus no sampling, were applied to each of the 3500 training datasets, resulting in $32 \times 3500 = 112,000$ tranformed datasets, each of which is used for learner construction. Since there are 11 learners, a total of $11 \times 112,000 = 1,232,000$ classifiers were constructed and evaluated in our experiments.

To measure the performance of the classifiation algorithms, the area under the ROC curve (AUC), Kolmogorov-Smirnov statistic (K/S) (Hand, 2005), geometric mean (G), F-measure (F), accuracy (Acc), and true positive rate (TPR) were calculated. The last four performance measures utilize the implicit classification threshold of 0.5 (i.e., if the posterior probability of posi-

Table 4. Sampling, $\pi < 10\%$, NB

Level	AUC	HSD	Level	G	HSD
ROS750	0.896	A	RUS5	81.78	A
RUS25	0.896	A	SM1000	81.37	A
BSM50	0.895	A	ROS1000	80.96	A
WE-W	0.895	A	BSM1000	76.79	B
NONE	0.895	A	CBOS	76.46	B
SM50	0.895	A	OSS	70.06	C
OSS	0.894	A	WE-W	61.21	D
CBOS	0.887	A	NONE	60.72	D
Level	**AUC**	**HSD**	**Level**	**G**	**HSD**
SM200	0.842	A	RUS5	70.98	A
BSM50	0.841	A	WE-W	70.17	A
WE-E	0.841	A	SM1000	70.12	A
RUS90	0.840	A	BSM1000	69.99	A
ROS750	0.840	A	ROS1000	69.47	AB
NONE	0.840	A	NONE	69.23	AB
OSS	0.831	A	OSS	67.28	B
CBOS	0.805	B	CBOS	57.70	C

Table 5. Sampling, $\pi < 10\%$, C4.5N

Level	AUC	HSD	Level	G	HSD
SM100	0.886	A	RUS5	81.51	A
BSM1000	0.884	A	SM750	66.87	B
WE-E	0.882	A	ROS500	64.98	BC
ROS50	0.881	A	CBOS	64.16	BCD
RUS25	0.881	A	BSM750	63.52	BCD
NONE	0.881	A	OSS	61.97	BCD
OSS	0.856	B	WE-W	60.34	CD
CBOS	0.846	B	NONE	59.39	D
Level	**AUC**	**HSD**	**Level**	**G**	**HSD**
SM300	0.853	A	RUS10	76.34	A
ROS300	0.853	A	SM1000	69.74	B
BSM1000	0.844	AB	BSM1000	67.97	BC
WE-W	0.833	BC	ROS1000	64.87	CD
RUS25	0.829	BC	WE-W	62.89	CD
OSS	0.824	C	CBOS	61.58	DE
NONE	0.820	C	OSS	60.98	DE
CBOS	0.814	C	NONE	57.66	E

tive class membership is > 0.5, then the example is classified as belonging to the positive class). The first two, the AUC and K/S, measure the general ability of the classifier to separate the positive and negative classes.

6. Results

6.1. Experimental Data

The first set of results we present are for some of the individual learners separately. Due to space limitations we can only provide a small sampling of the results, however. First, the datasets were grouped into four categories based on severity of imbalance: those with $\pi < 5\%$, $5\% < \pi < 10\%$, $10\% < \pi < 20\%$, and finally $20\% < \pi$ (π is the percentage of examples belonging to the minority class). The reason for this categorization scheme is to capture differences in the performance of sampling techniques given different levels of imbalance. We focus primarily on the results from $\pi < 10\%$ for the learners SVM, RF, NB, C4.5N, and LR (Tables 2 to 6). Each of these tables shows the ordering of the sampling techniques, as measured by AUC and G,

Table 6. Sampling, $\pi < 10\%$, LR

Level	AUC	HSD	Level	G	HSD
ROS300	0.892	A	RUS5	81.14	A
WE-W	0.890	A	CBOS	79.08	AB
NONE	0.889	A	ROS1000	77.31	AB
RUS75	0.889	A	SM1000	75.27	BC
OSS	0.888	A	BSM1000	71.45	BC
BSM50	0.887	A	OSS	56.94	D
SM50	0.886	A	WE-W	49.24	E
CBOS	0.860	B	NONE	47.54	E
Level	**AUC**	**HSD**	**Level**	**G**	**HSD**
ROS500	0.847	A	ROS1000	77.09	A
WE-W	0.846	A	SM1000	76.34	A
RUS75	0.843	A	RUS10	76.03	AB
SM300	0.841	A	BSM1000	75.04	AB
BSM500	0.840	A	CBOS	72.47	B
NONE	0.839	A	WE-W	52.89	C
OSS	0.839	A	OSS	49.35	CD
CBOS	0.809	B	NONE	46.28	D

along with a test of statistical significance. In Tables 2 to 6, the first nine rows are the results for datasets with $\pi < 5\%$, while the second nine rows are for the datasets with $5\% < \pi < 10\%$. The values for the performance measure (either AUC or G) in Tables 2 to 6 are averaged over all of the datasets with either $\pi < 5\%$ at the top of the table or $5\% < \pi < 10\%$ at the bottom of the table. For example, from Table 2, SVM with ROS1000 obtained an average AUC of 0.898 over the 20 CV runs of the eight datasets with $\pi < 5\%$, and SVM with ROS1000 obtained an average AUC of 0.861 over the 20 CV runs of the 11 datasets with $5\% < \pi < 10\%$.

For each learner and group of datasets, a one-factor analysis of variance (ANOVA) (Berenson et al., 1983) was constructed, where the factor was the sampling technique. Tukey's Honestly Significant Difference (HSD) test (SAS Institute, 2004) is a statistical test comparing the mean value of the performance measure for the different sampling techniques. Two sampling techniques with the same block letter are not significantly different with 95% statistical confidence (all of the statistical tests in this work use 95% confidence level). Finally note that these tables show the parameter value for each of the seven types of sampling that achieved the optimal value. For example, from Table 2, ROS at 1000% obtained the highest average AUC (across all of the datasets with $\pi < 5\%$) of 0.898, followed by RUS at 5%. Note that based on the average AUC over all datasets with $\pi < 5\%$, ROS1000, RUS5, SM1000, and BSM1000 are not significantly different from one another (they all have the block letter 'A' in the HSD column) when used with the SVM classifier. Further, RUS5, SM1000, BSM1000, and CBOS are not significantly different from one another, since they have the block letter 'B' in the HSD column. We present the results only for these five learners and only these two performance measures due to space limitations. AUC and G were included because they represent one measure that is threshold dependent

Table 7. Best Sampling Technique By Learner, AUC

AUC	< 5%	5% - 10%	10% - 20%	> 20%
C4.5D	RUS5	RUS10	RUS25	RUS50
C4.5N	SM100	SM300	WE-W	WE-W
LR	ROS300	ROS500	ROS500	NONE
MLP	RUS10	ROS300	ROS300	ROS200
NB	ROS750	SM200	SM750	NONE
RBF	BSM500	RUS10	RUS90	WE-W
RF	RUS5	RUS10	WE-W	WE-W
RIPPER	RUS5	RUS10	SM750	SM200
SVM	ROS1000	ROS1000	SM200	ROS100
2NN	WE-W	SM200	WE-W	WE-W
5NN	BSM300	SM1000	WE-W	WE-W

Table 8. Best Sampling Technique By Learner, G

G	< 5%	5% - 10%	10% - 20%	> 20%
C4.5D	RUS5	RUS10	RUS25	RUS50
C4.5N	RUS5	RUS10	RUS25	RUS50
LR	RUS5	ROS1000	SM500	ROS200
MLP	RUS5	ROS1000	ROS500	ROS200
NB	RUS5	RUS5	BSM1000	BSM200
RBF	RUS5	RUS10	RUS25	ROS200
RF	RUS5	RUS10	RUS25	SM1000
RIPPER	RUS5	RUS10	SM750	SM300
SVM	RUS5	ROS1000	ROS500	ROS200
2NN	RUS5	RUS10	ROS200	ROS200
5NN	RUS5	ROS500	BSM1000	SM200

Table 9. Best Sampling Technique By Learner, K/S

K/S	< 5%	5% - 10%	10% - 20%	> 20%
C4.5D	RUS5	RUS10	RUS25	BSM50
C4.5N	SM500	SM300	BSM50	WE-W
LR	ROS500	ROS1000	ROS1000	OSS
MLP	RUS10	ROS1000	ROS300	ROS200
NB	WE-W	WE-E	BSM50	WE-W
RBF	RUS5	RUS10	RUS90	WE-W
RF	RUS10	SM1000	WE-W	WE-W
RIPPER	RUS5	RUS10	SM750	SM300
SVM	ROS1000	ROS1000	ROS300	ROS100
2NN	WE-W	SM1000	WE-W	WE-W
5NN	ROS750	SM1000	WE-E	WE-W

Table 10. Best Sampling Technique By Learner, F

F	< 5%	5% - 10%	10% - 20%	> 20%
C4.5D	SM300	SM300	SM100	RUS50
C4.5N	SM200	SM300	SM100	WE-W
LR	ROS300	ROS500	SM300	ROS200
MLP	ROS300	ROS300	SM200	ROS200
NB	RUS25	NONE	WE-W	BSM200
RBF	RUS25	RUS25	SM200	ROS300
RF	SM1000	SM750	WE-E	WE-E
RIPPER	CBOS	SM1000	SM500	SM300
SVM	ROS300	SM500	SM300	ROS200
2NN	ROS200	ROS750	WE-W	WE-W
5NN	ROS200	ROS200	BSM100	SM200

(G) and one that is not (AUC). Note for example that sampling does not significantly improve the AUC obtained by NB (Table 4), however applying either RUS, SM, or BSM does significantly improve G.

Tables 7 to 10 present the sampling technique which results in the best AUC, G, K/S, and F measures for each learner and group of imbalance. If applying the sampling technique resulted in performance that was significantly better (with 95% statistical confidence) than that of using no sampling, then the technique is underlined.

Table 11 presents, over all 35 datasets, 11 learners, and six performance measures (AUC, K/S, G, F, Acc, and TPR), the number of times the rank of the sampling technique was $1, 2, \ldots, 8$. A rank of one means that the sampling technique, for a given dataset, learner, and performance measure, resulted in the highest value for the performance measure[1]. RUS resulted in the best performance 748 times (or 32.0% = 748/2340), followed by ROS (408 times). OSS and CBOS were rarely the best technique (66 or 2.8% for OSS and 86 or 3.7% for CBOS). Further CBOS resulted in the worst performance (rank 8, last column) 965 or 42.0% of the time, followed by no sampling, which was the worst 862 or 37.5% of the time.

[1]Note that it is possible for ties to occur. Suppose, for example, that two sampling techniques obtained the best AUC. Both of these techniques would be given a rank of one, while the next best technique would be given a rank of three. Therefore the sum of the columns is not exactly equal for each rank.

Tables 12 and 13 display the ranking of each sampling technique separately for the four groups of imbalance ($\pi < 5\%$ at the top of Table 12 and $5\% < \pi < 10\%$ at the bottom, with $10\% < \pi < 20\%$ at the top of Table 13 and $\pi > 20\%$ at the bottom). Note that adding the individual cells of Tables 12 and 13 produces Table 11. Finally, Tables 14 to 16 show the rankings of the sampling techniques only for datasets with $\pi < 5\%$ and separately for each of the six performance measures, AUC, K/S, G, F, Acc, and TPR (adding the individual cells of Tables 14 to 16 produces the top half of Table 12).

6.2. Discussion of Results

Based on the experiments conducted in this work, a number of conclusions can be drawn. The utility of sampling depends on numerous factors. First, different types of sampling work best with different learners. RUS worked very well for C4.5D (not shown) and RF, while ROS works well with LR. Second, the value of sampling is heavily dependent on the performance measure being used. AUC and K/S, which are classification-threshold independent, generate different results than G, F, TPR, and Acc, which utilize the standard 0.5 threshold on the posterior probability. For numerous learners, such as NB, LR, 2NN, and 5NN (and to a slighly lesser extent RBF and MLP), none of the sampling techniques significantly improved the performance of the learner as measured by the AUC or K/S. However, when the performance is measured using the threshold-dependent measures, significant improvements for all learners are

Table 11. Rank of Sampling Techniques, All Datasets

	Number of times Ranked							
Method	1	2	3	4	5	6	7	8
BSM	274	352	470	451	250	246	209	58
CBOS	86	112	115	170	406	180	276	965
NONE	236	130	147	115	165	248	407	862
OSS	66	135	167	128	234	482	809	289
ROS	408	442	365	410	325	209	145	6
RUS	748	354	367	369	270	118	67	17
SM	302	586	488	362	249	208	97	18
WE	220	195	184	303	410	610	306	82

Table 12. Rank of Sampling Techniques, Datasets $\pi < 10\%$

	Number of times Ranked							
Method	1	2	3	4	5	6	7	8
BSM	48	68	90	72	81	93	54	22
CBOS	45	59	38	30	101	26	58	171
NONE	44	40	30	25	40	82	94	173
OSS	22	42	35	29	56	84	142	118
ROS	107	91	94	120	63	31	19	3
RUS	212	89	93	61	42	12	17	2
SM	37	104	99	118	76	57	31	6
WE	18	39	44	70	73	140	117	27
Method	1	2	3	4	5	6	7	8
BSM	86	112	177	151	69	59	62	10
CBOS	22	23	35	73	140	74	66	293
NONE	84	27	37	37	50	82	123	286
OSS	26	37	45	36	57	128	294	103
ROS	107	133	123	143	105	72	40	3
RUS	273	99	97	103	94	46	10	4
SM	113	227	138	90	75	50	29	4
WE	39	61	65	99	131	211	104	16

obtained. For NB, for example, none of the sampling techniques improved the performance on datasets with $\pi < 5\%$ as measured by the AUC, however, relative to G, RUS, SM, and ROS significantly improved the performance (RUS, SM, and ROS achieved G > 80, while no sampling resulted in G $= 60.72$).

Further, consider Tables 7 to 10. Using the AUC, sampling significantly improved upon the performance of the classifier constructed with the unaltered data in only 15 of 44 scenarios (12 of the 15 occurrences were for datasets with $\pi < 10\%$). For K/S, sampling improved the performance in 12 of the 44 scenarios. For G and F, however, in 42 and 34 of 44 scenarios, respectively, sampling significantly outperformed not using sampling.

RUS performed very well in our experiments, although for individual learners or datasets, other methods were sometimes better. Overall, however, RUS resulted in very good performance, being the best sampling technique 748 of 2340 times. ROS performed the second best overall, followed by SM and BSM. OSS and CBOS in particular performed very poorly, with the latter obtaining the worst overall ranking of the sampling techniques 965 of 2297 times. For datasets with more severe imbalance, RUS does even better, as can be seen from Tables 12 and 13, where RUS was the best technique 39.8% and 36.4% of the time for datasets with $\pi < 5\%$ and $5\% < \pi < 10\%$.

Table 13. Rank of Sampling Techniques, Datasets $\pi > 10\%$

	Number of times Ranked							
Method	1	2	3	4	5	6	7	8
BSM	72	56	75	93	33	22	27	18
CBOS	14	17	13	33	56	25	66	172
NONE	33	17	32	19	35	37	77	146
OSS	6	20	26	25	40	99	136	44
ROS	70	82	55	52	55	53	29	0
RUS	104	48	61	86	56	28	12	1
SM	55	111	105	49	35	25	15	1
WE	43	44	33	37	86	105	34	14
Method	1	2	3	4	5	6	7	8
BSM	68	116	128	135	67	72	66	8
CBOS	5	13	29	34	109	55	86	329
NONE	75	46	48	34	40	47	113	257
OSS	12	36	61	38	81	171	237	24
ROS	124	136	93	95	102	53	57	0
RUS	159	118	116	119	78	32	28	10
SM	97	144	146	105	63	76	22	7
WE	120	51	42	97	120	154	51	25

Table 14. Rank of Sampling Techniques, Datasets $\pi < 5\%$, AUC and K/S

	Number of times Ranked							
AUC	1	2	3	4	5	6	7	8
BSM	13	15	12	12	7	14	8	7
CBOS	2	8	3	2	11	6	15	41
NONE	1	8	10	8	19	20	8	14
OSS	3	1	4	8	14	14	22	22
ROS	22	12	16	19	10	6	3	0
RUS	35	12	16	10	7	3	5	0
SM	7	23	12	11	8	14	13	0
WE	5	11	13	18	13	10	14	4
K/S	1	2	3	4	5	6	7	8
BSM	16	18	15	5	9	13	6	6
CBOS	9	4	6	4	12	1	14	38
NONE	1	8	10	7	8	24	16	14
OSS	4	3	5	4	10	16	22	24
ROS	22	13	18	15	14	4	2	0
RUS	26	15	12	15	14	0	6	0
SM	8	21	12	18	5	13	9	2
WE	3	8	10	17	20	14	12	4

Finally, considering in more detail those datasets with $\pi < 5\%$ in Tables 14 to 16, RUS maintains a slight edge over ROS as the best sampling technique relative to the AUC, K/S, and F. Relative to G, RUS is clearly the best sampling technique. As would be expected, not using sampling typically results in the highest overall accuracy (Table 16), however since we are interested in detecting examples of the positive class, this measure is very misleading. We believe overall accuracy is not an appropriate measure, especially given imbalanced data, however it is presented because it is often used in related work. When considering the TPR, RUS is clearly the most successful.

One of the most important conclusions that can be drawn from these experiments is the inferior performance of the 'intelligent' sampling techniques, SM, BSM, WE, OSS, and CBOS (especially the last two). While these techniques seem to be promising solutions to

Table 15. Rank of Sampling Techniques, Datasets $\pi < 5\%$, G and F

	Number of times Ranked							
G	1	2	3	4	5	6	7	8
BSM	6	8	19	14	19	17	2	3
CBOS	6	14	15	11	28	6	1	7
NONE	0	0	1	0	4	10	22	51
OSS	3	7	4	3	6	17	30	18
ROS	16	19	21	28	4	0	0	0
RUS	53	20	9	2	4	0	0	0
SM	5	19	19	27	16	0	2	0
WE	0	0	0	3	7	38	33	7
F	1	2	3	4	5	6	7	8
BSM	8	17	18	16	11	10	8	0
CBOS	9	5	1	2	27	10	12	22
NONE	1	5	2	5	4	17	20	34
OSS	4	8	4	4	8	16	20	24
ROS	26	20	11	21	5	4	1	0
RUS	28	11	15	17	10	3	3	1
SM	10	18	32	16	6	4	1	1
WE	3	3	5	7	18	23	24	5

Table 16. Rank of Sampling Techniques, Datasets $\pi < 5\%$, Acc and TPR

	Number of times Ranked							
Acc	1	2	3	4	5	6	7	8
BSM	2	5	3	12	16	19	28	3
CBOS	7	7	1	1	0	2	12	58
NONE	41	19	7	4	1	5	11	0
OSS	7	15	14	7	9	3	13	20
ROS	17	5	3	9	22	16	13	3
RUS	6	15	34	16	7	6	3	1
SM	3	4	10	15	23	25	5	3
WE	7	17	16	24	8	14	2	0
TPR	1	2	3	4	5	6	7	8
BSM	3	5	23	13	19	20	2	3
CBOS	12	21	12	10	23	1	4	5
NONE	0	0	0	1	4	6	17	60
OSS	1	8	4	3	9	18	35	10
ROS	4	22	25	28	8	1	0	0
RUS	64	16	7	1	0	0	0	0
SM	4	19	14	31	18	1	1	0
WE	0	0	0	1	7	41	32	7

the problem of class imbalance, simpler techniques such as RUS or ROS often performed much better. CBOS and OSS especially performed very poorly in our experiments, very rarely being the best sampling technique and often being among the worst.

6.3. Threats to Validity

Two types of threats to validity are commonly discussed in empirical work (Wohlin et al., 2000). Threats to internal validity are unaccounted influences that may impact the results. Threats to external validity consider the generalization of the results outside the experimental setting, and what limits, if any, should be applied.

All experiments were conducted using WEKA (Witten & Frank, 2005), which is commonly used in machine learning research. Some enhancements were required to implement some of the sampling techniques, and all developed software was thoroughly tested. ANOVA analysis was performed using the SAS GLM procedure (SAS Institute, 2004), and all assumptions for valid statistical inference were verified. Extensive care was taken to ensure the validity of our results.

External validity questions the reliability and generalizability of the experimental results. The comprehensive scope of our experiments greatly enhances the reliability of our conclusions, which is why we utilized 35 different real-world datasets. Performing numerous repetitions of cross validation greatly reduces the likelihood of anomalous results due to selecting a lucky or unlucky partition of the data. Building over one million learners in these experiments allows us to be confident in the reliability of our experimental conclusions.

One important consideration is the 'free' parameter associated with the four sampling techniques RUS, ROS, SM, and BSM. Prior work has suggested that no universal prior is optimal for tree construction (Weiss & Provost, 2003), so instead of only using one selected parameter (e.g., balanced classes), we tried numerous possibilities, but only from a limited selection - in other words, no attempt was made to optimize over all possible sampling percentages. Further, when utilizing sampling in practice, the user does have the ability to choose a value which produces good results, for example using cross validation. Our work has shown that in many cases, a sampling percentage which is more towards balanced is better than other choices, and future work should explore this further. In addition, as the sampling percentage varied near the one we deemed 'best', the results did not change dramatically. For example, RUS5 was the best technique for C4.5D for the datasets with $\pi < 5\%$ with respect to AUC (Table 7), but the AUC of RUS10 was very similar. Further, with OSS and CBOS, the techniques explicitly describe how to add/remove instances, so there was no ability to directly alter the level of sampling and we reported the single level of performance achieved. Therefore, we do not believe that the comparison of sampling techniques was unfairly biased towards those with a free parameter.

7. Conclusions

We have presented a comprehensive and systematic experimental analysis of learning from imbalanced data, using 11 learning algorithms with 35 real-world benchmark datasets from a variety of application domains. The objective of this research is to provide practical guidance to machine learning practitioners when building classifiers from imbalanced data, and to present to researchers some possible directions for future study. To our knowledge, no related work has attempted to empirically analyze class imbalance from such a wide scope, comparing learners, sampling techniques, and performance measures using many different datasets. Unfor-

tunately due to space limitations, we can only present a small fraction of our experimental results, however the data clearly demonstrate that sampling is often critical to improving classifier performance, especially optimizing threshold-dependent measures such as the geometric mean or TPR. Further, individual learners respond differently to the application of sampling. Much of the related work on class imbalance has focused on decision tree learners, however these results show that the observations made for decision trees will not carry over to neural networks, regression, or nearest neighbor classification algorithms. Future work may consider additional learners, e.g., different variations of neural network or SVM learners. Sampling can also be compared to cost-sensitive learning in future work. Alternative measures of classifier performance can also be analyzed. Future work should also consider sampling in the context of multi-class learning.

Acknowledgments

We are grateful to the current and former members of the Empirical Software Engineering and Data Mining and Machine Learning Laboratories at Florida Atlantic University for their reviews of this work. We also sincerely thank the anonymous reviewers for their helpful comments.

References

Barandela, R., Valdovinos, R. M., Sanchez, J. S., & Ferri, F. J. (2004). The imbalanced training sample problem: Under or over sampling? *In Joint IAPR International Workshops on Structural, Syntactic, and Statistical Pattern Recognition (SSPR/SPR'04), Lecture Notes in Computer Science 3138*, 806–814.

Berenson, M. L., Levine, D. M., & Goldstein, M. (1983). *Intermediate statistical methods and applications: A computer package approach*. Prentice-Hall, Inc.

Blake, C., & Merz, C. (1998). UCI repository of machine learning databases. *http://www.ics.uci.edu/ mlearn/ MLRepository.html*. Department of Information and Computer Sciences, University of California, Irvine.

Breiman, L. (2001). Random forests. *Machine Learning, 45*, 5–32.

Chawla, N. V., Hall, L. O., Bowyer, K. W., & Kegelmeyer, W. P. (2002). Smote: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 321–357.

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

Han, H., Wang, W. Y., & Mao, B. H. (2005). Borderline-smote: A new over-sampling method in imbalanced data sets learning. *In International Conference on Intelligent Computing (ICIC'05). Lecture Notes in Computer Science 3644* (pp. 878–887). Springer-Verlag.

Hand, D. J. (2005). Good practice in retail credit scorecard assessment. *Journal of the Operational Research Society, 56*, 1109–1117.

Japkowicz, N. (2000). Learning from imbalanced data sets: a comparison of various strategies. *AAAI Workshop on Learning from Imbalanced Data Sets (AAAI'00)* (pp. 10–15).

Jo, T., & Japkowicz, N. (2004). Class imbalances versus small disjuncts. *SIGKDD Explorations, 6*, 40–49.

Kubat, M., & Matwin, S. (1997). Addressing the curse of imbalanced training sets: One sided selection. *Proceedings of the Fourteenth International Conference on Machine Learning* (pp. 179–186). Morgan Kaufmann.

Maloof, M. (2003). Learning when data sets are imbalanced and when costs are unequal and unknown. *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets*.

Monard, M. C., & Batista, G. E. A. P. A. (2002). Learning with skewed class distributions. *Advances in Logic, Artificial Intelligence and Robotics (LAPTEC'02)* (pp. 173–180).

Quinlan, J. R. (1993). *C4.5: Programs for machine learning*. San Mateo, California: Morgan Kaufmann.

SAS Institute (2004). *SAS/STAT user's guide*. SAS Institute Inc.

Weiss, G. M., & Provost, F. (2003). Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 315–354.

Witten, I. H., & Frank, E. (2005). *Data mining: Practical machine learning tools and techniques*. San Francisco, California: Morgan Kaufmann. 2nd edition.

Wohlin, C., Runeson, P., Host, M., Ohlsson, M. C., Regnell, B., & Wesslen, A. (2000). *Experimentation in software engineering: An introduction*. Kluwer International Series in Software Engineering. Boston, MA: Kluwer Academic Publishers.