# Wrapper-based Computation and Evaluation of Sampling Methods for Imbalanced Datasets

Nitesh V. Chawla
Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN
nchawla@cse.nd.edu

Lawrence O. Hall
Department of Computer Science and Engineering
University of South Florida
Tampa, FL
hall@cse.usf.edu

Ajay Joshi
Department of Computer Science and Engineering
University of South Florida
Tampa, FL
ajoshi2@cse.usf.edu

## ABSTRACT

Learning from imbalanced datasets presents an interesting problem both from modeling and economy standpoints. When the imbalance is large, classification accuracy on the smaller class(es) tends to be lower. In particular, when a class is of great interest but occurs relatively rarely such as cases of fraud, instances of disease, and regions of interest in large-scale simulations, it is important to accurately identify it. It then becomes more costly to misclassify the interesting class. In this paper, we implement a wrapper approach that computes the amount of under-sampling and synthetic generation of the minority class examples (SMOTE) to improve minority class accuracy. The *f-value* serves as the evaluation function. Experimental results show the wrapper approach is effective in optimization of the composite *f-value*, and reduces the average cost per test example for the datasets considered. We report both average cost per test example and the cost curves in the paper. The true positive rate of the minority class increases significantly without causing a significant change in the *f-value*. We also obtain the lowest cost per test example, compared to any result we are aware of for the KDD Cup-99 intrusion detection data set.

## Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning - Induction; H.2.8 [**Database Management**]: Applications - Data Mining

## General Terms

Algorithms, Performance, Design, Experimentation

## Keywords

cost-sensitive learning and evaluation, imbalanced datasets, wrapper, under-sampling, SMOTE

## 1. INTRODUCTION

In this work, we focus on the problem of learning a classification model from imbalanced data sets. An imbalanced data set is one in which there is a significant difference in the number of examples in a set of classes. The imbalanced datasets pose an economic or utility problem, as there is usually a higher cost in misclassifying the interesting class. A simple consistent guess can become an accurate classifier, by classifying everything as the majority class, but that is not useful for the problem at hand. On the other hand, a simple guess classifying everything as the interesting class will also not work due to the number of false positives. One wants a high number of true positives, while maintaining a low false-positive rate.

There are many examples of imbalanced data sets where the minority class is of interest. For example, cellular-phone fraud or credit card fraud data are typically comprised of a very small proportion of the the fraudlent cases (minority class) [19, 29, 11]. However, it is quite important to predict a fraudulent transaction. It is also important to minimize the false positives (the nonfraudulent transactions that are predicted to be fraudulent) because these cost time to investigate and can potentially upset the customer. Thus, there is a non-zero cost associated with the false positives as well. Typically, the cost with the false negatives will be the cost of the transaction. We don't want a system that will strongly target true positives at the expense of a high false positive rate, thereby increasing the total cost of the operation.

As another example, large-scale simulations can be based on extremely large data sets. Some simulations are replacing or augmenting physical experiments. This requires that they be done in great detail [26, 9]. However, the process of building very large-scale simulations and examining them for correctness when looking for important, subtle details may prevent all areas of interest from being viewed [6]. In any event, the process of validating a simulation can take weeks to months. A similar amount of time is required to actually utilize and explore the simulation. This is indicative of the great opportunity for building intelligent tools which can help the simulation designers/users find regions of interest and/or anomalies quickly. There is a cost involved not only in correctly displaying the regions of interest but also the costs in time. Hence, the intelligent tool should not only be "fast", but also accurately identify the interesting regions, without too many false alarms. Having many false alarms for the user to browse through can inadvertently increase the cost in terms of the time spent. As these two examples highlight, there is a "utility" associated with the usage of


*UBDM'05* August 21, 2005, Chicago, Illinois, USA.

a technique. That utility is comprised of various costs of errors, time spent, etc.

We investigate an enhancement to a particular composite approach, combining over-sampling by creating new examples and under-sampling, for dealing with imbalanced data sets. The Synthetic Minority Oversampling TEchnique (SMOTE) creates synthetic examples from minority classes [14]. We also under-sample the majority class(es) to obtain higher accuracy on the minority class(es) without greatly increasing the number of false positives. However, previous work has not shown how to effectively set the amounts of under-sampling and SMOTE for a given dataset. In this paper, we explore an automated method to do this. We set up the method such that the *f-value* [10] is optimized. We chose the *f-value* as it is a composite measure that incorporates both the false positives and false negatives. Hence, if an approach significantly increases the true positives, but also increases the false positives, then the *f-value* will appropriately reflect that. We evaluate the final performance of the classifiers under a cost-based framework using cost-curves and average cost per test example.

It is important to identify the potentially optimal under-sampling and SMOTE percentages. The amount of sampling performed to mitigate the imbalance in class distribution will have an effect on the performance of the classifier. We want to reduce the costs per test example. The utility of the learning algorithm for a particular domain or task is strongly dependent on the right amount of sampling and the examples distribution in the dataset. Each dataset and the corresponding class distribution will have its own requirements [32]. The computational time and resources spent deploying the wrapper technique should be mitigated by the reduced cost per test example, and a higher detection of the interesting class or regions in the dataset. There is a tradeoff between the time spent in learning or searching for the parameters, and the relative reduction in the costs or improvement in the true positives on the testing set. We will show that minority class accuracy is improved on several data sets with only small increases in false positive predictions. In addition, we will also show that our approach produces much reduced costs per test example. The approach is shown to be both tractable computationally and effective in choosing the parameters.

## 2. LEARNING FROM IMBALANCED DATASETS

Researchers in the machine learning community have dealt with the problem of class imbalance by using various approaches like over-sampling the minority classes, undersampling the majority classes, assigning different costs for different misclassification errors, learning by recognition as opposed to discrimination, etc [3, 25, 27, 32, 4, 1, 24, 2, 34]. There is a significant body of research comparing the various sampling methods [12, 28, 17, 22, 7]. Sampling strategies have almost become the de facto standard for countering the imbalance in datasets [13]. With all this there is still no answer on how to do the sampling required for obtaining good classifier accuracies on minority classes.

There are a number of different approaches that can be applied to build classifiers on imbalanced data sets. In this work, we examined under sampling and over-sampling by creating synthetic examples of minority classes. Under-sampling the majority class can reduce the bias of the learned classifier towards it and thus improve the accuracy on the minority classes.

Some studies [27, 21] have been done which combined under-sampling of majority classes with over sampling by replication of minority classes. While Japkowicz [21] found this approach very effective, Ling and Li [27] were not able to get significant improvement in their performance measures. Japkowicz experimented with only one-dimensional artificial data of varying complexity whereas Ling and Li used real data from a Direct Marketing problem. This might have been the reason for the discrepancy between their results. On the whole, from the body of literature, it was found that under-sampling of majority classes was better than over-sampling with replication of minority classes [17, 12] and that the combination of the two did not significantly improve the performance over under sampling alone.

Chawla et al. [14] introduced a new over-sampling approach for two class problems that over-sampled the minority class by creating synthetic examples rather than replicating examples. They pointed out the limitation of over-sampling with replication in terms of the decision regions in feature space for decision trees. They showed that as the minority class was over sampled by increasing amounts, for decision trees, the result was to identify similar but more specific regions in the feature space. A preferable approach is to build generalized regions around minority class examples.

The synthetic minority over-sampling technique (SMOTE) was introduced to provide synthetic minority class examples which were not identical but came from the same region in feature space. The over-sampling was done by selecting each minority class example and creating a synthetic example along the line segment joining the selected example and any/all of the k minority class nearest neighbors. In the calculations of the nearest neighbors for the minority class examples a Euclidean distance for continuous features and the value Distance Metric (with the Euclidean assumption) for nominal features was used. For examples with continuous features, the synthetic examples are generated by taking the difference between the feature vectors of selected examples under consideration and their nearest neighbors. The difference between the feature vectors is multiplied by a random number between 0 and 1 and then added to the feature vector of the example under consideration to get a new synthetic example. For nominal valued features, a majority vote for the feature value is taken between the example under consideration and its k nearest neighbors. This approach effectively selects a random point along the line segment between the two feature vectors. This strategy forces the decision regions of the minority class learned by the classifier to become more general and effectively provides better generalization performance on unseen data.

However, an investigation into how to choose the number of examples to be added was not done. In addition, the amount of under-sampling also needs to be determined. Given the various costs of making errors, it is important to identify potentially optimal values for both SMOTE and under-sampling. This is equivalent to discovering the operating point in the ROC space giving the best trade-off between True Positives and False Positives. In this paper, we develop an approach to automatically set the parameters. We discuss a wrapper framework using cross-validation that

performs a step-wise and greedy search for the parameters. Note that while the computational aspects of the automated approach induces certain costs, we do not incorporate that into our framework. We optimize based on the different types of errors made. However, we do try to restrict our search space. We show that this approach works on three highly skewed datasets. We also utilized a cost-matrix to indicate the costs per test example based on the different kinds of errors.

## 3. WRAPPER

In this work, a wrapper [23] approach was utilized to determine the percentage of minority class examples to add to the training set and the percentage to under-sample the majority class examples. The wrapper approach works by doing a guided search of the parameter space. In this case the underlying classifier is used to evaluate the chosen performance function for every considered amount of under-sampling and SMOTE. A particular parameter or set of parameters is chosen and a five-fold cross validation on the train data is done to get the performance average. The parameters are varied in a systematic way such that a set of parameter candidates are generated, training sets are updated, and the classifiers built and evaluated. The candidate associated with the highest performance is chosen to have its parameters systematically modified to create new candidate solutions. This process is a type of best-first search. In order to evaluate the effectiveness of the wrapper approach in selecting the parameters for under-sampling and SMOTE, we need to use a metric other than strict accuracy. With imbalanced data, accuracy can be misleading, because it causes you to favor high prediction accuracy on the majority class which is often uninteresting. Hence, the *f-value* metric was used as the evaluation function [10]. It is made up of two measures: *precision* which gives us the measure of correctness of the classifier in predicting the actual positive or minority class, whereas *recall* gives us the measure of the percentage of positive or minority class examples predicted correctly. The precision, recall and *f-value* were calculated as follows, where $\beta$ corresponds to the relative importance of *precision* vs *recall*.

$$precision = \frac{TP}{TP + FP} \tag{1}$$

$$recall = \frac{TP}{TP + FN} \tag{2}$$

$$f - value = \frac{(1 + \beta^2) \times precision \times recall}{\beta^2 \times recall + precision} \tag{3}$$

We implemented our wrapper approach as follows. We first do a ten-fold stratified split to separate the original dataset into ten training sets and ten disjoint testing sets. Then, for each of the ten training folds, we implement the wrapper approach using five-fold cross-validation to get more robust amount estimates for under-sampling and SMOTE. Note that these performance estimates will hold true only when either the training data is a good representative of the actual data distribution or the wrapper strategy does not over-fit the training data. If the training data is not a good representative of the actual data, no strategy can help. So the only thing which remains is to see whether the wrapper approach finds under-sampling and SMOTE levels which when used to build a classifier, do not over-fit the training data. Once the wrapper selects the particular amount of SMOTE and under-sampling, we apply those amounts five different random times on the training set, since both SMOTE and under-sampling randomly remove or create new instances. The classifiers learned from the updated training sets are evaluated on the same testing set, and those performances are averaged. This is done for each of the 10 folds. Thus, the final ten-fold average reported is essentially over fifty classifiers.

The two search parameters for the wrapper are the under-sampling and SMOTE percentages. The search space becomes large if the search is done simultaneously for both the under-sampling percentage and the SMOTE percentage (creation of new synthetic examples). Hence, we chose to first use wrappers to find the best under-sampling percentage. The wrapper starts with no under sampling for all majority classes and obtains baseline results on the training data. Then in a step-by-step greedy fashion it traverses through the search space of under-sampling percentages to seek better performance over the minority classes. The search process continues as long as it does not reduce the *f-value* of the minority classes or reduce the *f-value* over the majority classes more than some specified amount (generally 5%). Note that for under-sampling we look at both the minority and majority class *f-values*. We also looked at the *f-value* for the majority class as we only want to remove the redundant examples through undersampling, and not remove some of the important majority class examples. By looking at both the values simultaneously we are maintaining the decision regions for all the classes. Also, we wanted to identify the amount of under-sampling before introducing any synthetic minority class examples as that could have inadvertently penalized the *f-value* for the majority class. We want to first remove the majority class examples, that add no learning value to the base classifier.

Then with the under-sampling percentage fixed, we used the wrapper approach, to find the SMOTE percentage. Oversampling by creating synthetic examples is done until no minority class *f-value* increase is obtained for 3 candidate expansions. Now, for SMOTE we are only interested in improving the performance of the minority class. The *f-value* takes into account the increase in false positives (lowered precision), if any, by SMOTE increments. Thus, an overwhelming increase in the precision will stop the SMOTE process. This provides significantly improved computation times at the cost of a potential loss in accuracy. Once the best percentages for under-sampling and over-sampling via SMOTE are found, the training folds are updated with the requisite SMOTE and undersampling amounts. A classifier is then learned and evaluated on the unseen test data. We would like to be able to put this in a cost-framework if the time spent in searching for the "optimal" and "best" under-sampling and SMOTE percentages, justifies the performance improvement. We are investigating that line of work, as future work.

## 4. EXPERIMENTS

We report results on three data sets:

- Mammography Dataset,
- Forest Cover Dataset, and

**Table 1: Summary of Datasets. The percentages indicate the proportion of minority class in the complete dataset.**

| Dataset | # of Examples | # classes | # of Majority class examples | # of Minority class examples | # of attributes | # of continuous attributes |
|---|---|---|---|---|---|---|
| Mammography | 11183 | 2 | 10923 | 260 (2.3%) | 6 | 6 |
| Forest cover | 38501 | 2 | 35754 | 2747 (7.13%) | 54 | 54 |
| Modified KDD cup 99 (intrusion data) | 69980 | 5 | Normal: 35000; Dos: 25988; Probe: 4813 | U2R: 267 (0.41%); R2L: 3912 (5.95%) | 41 | 34 |

- KDD-cup 99: Network Intrusion Detection Dataset (two versions).

A brief summary of the datasets is presented in Table 1 and further details are given in later subsections. The Forest Cover dataset is available from the UCI repository [8] and our modifications to it will be described in the proceeding. The network intrusion data set comes from the KDD cup competition in 1999 [20] and the mammography data set is one that we locally extracted [33]. It is clear from Table 1 that there is significant imbalance between the two classes of each of these data sets. Hence, there is an opportunity to improve the minority class recognition accuracy because a typical classifier will be highly accurate but focused on the majority class. We report the *f-value* for all our experiments. The *f-value* assumed a $\beta$ of 1. We introduced a cost-matrix for the mammography dataset, as there can be a large cost associated with misclassification of a potentially malignant calcification (cancer) as non-calcification (non-cancerous). Moreover, there is also a slight cost associated with misclassifying the non-calcifications as calcifications. While there wasn't a natural application of costs to the forest cover dataset, we still constructed a cost-matrix for the sake of analysis. The KDD-cup dataset comes with a cost-matrix for each of the relative type of errors.

However, we did not incorporate the cost-matrix during the validation stage to select the amount of SMOTE and under-sampling. We are going to investigate that as a future line of work. It requires a definite cost matrix to be known for a dataset. It will be interesting to compare the SMOTE and undersampling parameters discovered using cost matrices during validation with the SMOTE and undersampling parameters discovered without using the cost-matrices (assuming the same loss).

## 4.1 Classifiers

Experiments were done with two types of classifiers, decision trees using software (USFC4.5) which emulates C4.5 release 8 [30] and a rule learning technique called RIPPER [15]. USFC4.5 was used with the default settings. By default, RIPPER will build rules first for the smallest class and will not build rules for the largest class. In the case of two class problems with imbalanced classes, such as here, only rules for the minority class are going to be built. Hence, one might expect that RIPPER will be better than a decision tree in accuracy on the minority class.

The wrapper algorithm that uses five fold cross-validation on the training set finds the undersampling and SMOTE percentages for a particular training fold (one of the ten folds for cross-validating the system). Then under-sampling and SMOTE are applied to each fold with wrapper selected percentages, a classifier was built on the updated training data and evaluated on the test data, unseen during the wrapper process. Due to the inherent random nature of under-sampling and SMOTE, the process of training and testing with wrapper selected under-sampling and SMOTE percentages is done five times to get an averaged (more stable) performance measure. To summarize, on each of the 10 folds, training and testing for wrapper selected SMOTE and under-sampling percentages was done five times i.e. SMOTE and under-sampling was done for a total of 50 times for cross-validation to get average stable results. All results reported in the proceeding are averages obtained in this way. In the tables, t-stat indicates the results of a significance test at the 95% level. This was a paired t-test. The x% of under-sampling means that x% of majority class examples were retained; and the y% of SMOTE means that many more examples of the minority class were created. For example, 200% of SMOTE means that twice as many (than the original number) minority class examples were created.

# 5. RESULTS

We did a ten-fold cross-validation, for mammography and forest cover datasets, in which the original dataset is stratified into ten disjoint sets or folds from which ten distinct testing sets and ten training sets are created. For the intrusion dataset, we utilized the training and testing sets as provided. We also used the cost-matrix as provided for the intrusion dataset and report the average cost per test example to compare with other published results [18]. For the mammography and forest cover datasets, we report various perfomance metrics, including *TPrate*, *FPrate*, *f-values*, average cost per test example at different cost ratios, and cost curves. Our main goal is to compare the classifiers in terms of reduction in the expected cost across different cost ratios. Drummond and Holte [16] introduced the cost space representation that allows for comparing different classifiers in terms of the expected cost. Let p(+) be the prior probability of the positive class, and p(-) be the prior probability of the negative class. $C(-|+)$ is cost of misclassifying a positive example as a negative example (false negative); and $C(+|-)$ is cost of misclassifying a negative example as a positive example (false positive). The Normalized Expected Cost (NE[C]) can then be expressed in terms of *TPrate*, *FPrate*, and Probability Cost Function (PCF) as follows:

$$PCF(+) = \frac{p(+)C(-|+)}{p(+)C(-|+) + p(-)C(+|-)} \qquad (4)$$

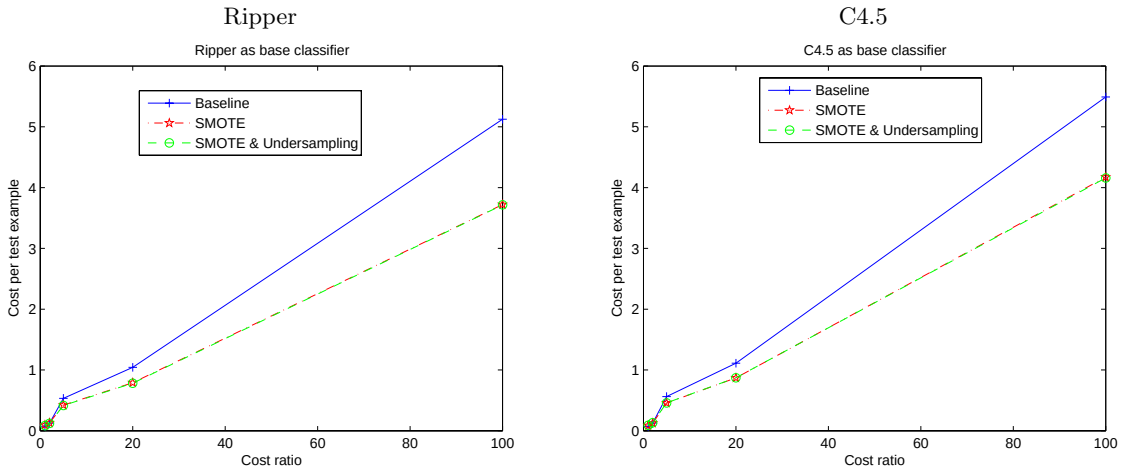


**Figure 1: Average Cost per test example at different cost ratios for the Mammography dataset.**

$$NE[C] = (1 - TPrate - FPrate) \times PCF(+) + FP \quad (5)$$

The performance of classifier using a fixed threshold, as used in this paper, is represented by a pair of (TP, FP). It can thus be represented as a line in the cost space, comprising of the normalized expected cost ($NE[C]$) in the y-axis and $PCF(+)$ in the x-axis. The range of both the measures is between 0 and 1. Given a family of such classifiers, if a classifier is lower in the normalized expected cost across a range of $PCF$, it dominates the other. One can, thus, choose a classifier that has a minimum cost either over a range of $PCF(+)$ or at a particular operating range.

## 5.1 Mammography Data

The Mammography Dataset was used in [33] and consists of 11,183 total samples with six numeric features and two classes representing calcification (cancerous) and non-calcification (non-cancerous). The minority class which represents calcification contained only 260 examples in the dataset i.e. only 2.32% of the total examples. The results obtained are shown in Table 2. The negative sign before the number in the '% increase' row indicates reduction in the associated value. It can be seen from Table 2, that for all four experimental trials, the wrapper algorithm was able to statistically significantly improve TP-rates for the minority class at the expense of a statistically significant reduction in *f-values* for the majority class. However, the wrapper method also produces significantly higher FP-rates than the baseline methods. But, the correspnding decrease in the *f-value* was not significant. Hence, the goal of a higher true positive rate is attainable without a significant reduction in the overall *f-value*.

We then constructed a cost-matrix for the mammography dataset by considering the the following costs of making errors between the positive and negative examples (using the convention $(C(+|-), C(-|+))$: (5, 5); (5, 10); (5, 50); (5, 100); and (5, 500). Figure 1 shows the results using these varied cost ratios with the different methods presented in Table 2. As one would expect, if using the same costs of errors, the baseline method produces the lowest cost per test example, and is indeed the preferred method. However, varying the costs from twice as much for false negative to 100 times, we see that the SMOTE classifier achieves the least cost. Under-sampling in conjunction with SMOTE provides very little reduction in the cost, if any. Both the C4.5 and Ripper classifiers exhibit similar behavior with SMOTE — significant improvement in performance over baseline. Ripper is well-suited for the task, as it is able to produce lower cost estimates per test example. We believe that incorporating a *f-value* in the wrapper framework maintains the relative importance of false positives and false negatives, as the number of true positives increases. However, one might vary the relative importance of precision and recall in the equation based on the specified costs. We assumed uniform costs in the *f-value*.

We also implemented cost-curves over the range of $PCF(+)$ established by varying $C(+|-)$ and $C(-|+)$ [16]. Figure 2 shows the result. Again over the wide range of $PCF(+)$, the Ripper-SMOTE classifier achieves the lowest expected costs. Until a $PCF(+)$ of 0.05 all the classifiers achieve similar performances, but beyond that the Ripper-SMOTE classifier dominates over the others.

## 5.2 Forest Cover Data

Originally, the Forest Cover dataset [8] consisted of 581,012 examples with 54 numeric features related to cartographic variables and seven classes representing the type of the forest cover. For our study, the data samples from two classes were extracted while the rest were ignored as done in [14]. The two classes we considered are Ponderosa Pine with 35,754 samples and Cottonwood/Willow with 2,747 samples. The results obtained on this dataset are tabulated below in Table 3.

For the Forest cover dataset, the results for the minority class were as expected, with the wrapper TP rate increasing with statistical significance. But the interesting thing about these results was that, the wrapper *f-values* obtained on the majority class using RIPPER in both scenarios actually increased slightly instead of decreasing which was the general trend. For the 'SMOTE only' scenario using RIPPER, the wrapper *f-values* were better than baseline *f-values* with statistical significance. For C4.5, the drop in the wrapper *f-values* over the majority class though statistically significant was extremely small. These were almost perfect results which one might always hope for, where the minority ex-

**Table 2: Results for the Mammography Data. $>$ indicates Wrapper is statistically significantly greater than Baseline; $<$ indicates Wrapper is statistically significantly lower than Baseline; and $\equiv$ indicates there is no statistically significant difference between the Wrapper and Baseline methods.**

| | | C4.5 | | Ripper | |
|---|---|---|---|---|---|
| | | SMOTE only | Undersampling and SMOTE | SMOTE only | Undersampling and SMOTE |
| | Average SMOTE % | 210% | 180% | 300% | 180% |
| | Average Under- sampling % | 100% | 87% | 100% | 94% |
| Average Minority Class TP-rate | Baseline | 0.546 | 0.546 | 0.577 | 0.577 |
| | Wrapper | 0.658 | 0.659 | 0.696 | 0.665 |
| | % increase | 16.96% | 17.15% | 17.13% | 13.29% |
| | t-stat | -4.7 | -3.913 | -4.276 | -3.322 |
| | significance | $>$ | $>$ | $>$ | $>$ |
| Average Minority Class FP-rate | Baseline | 0.0031 | 0.0031 | 0.00439 | 0.00439 |
| | Wrapper | 0.0088 | 0.0099 | 0.0114 | 0.0099 |
| | % increase | 64.58% | 68.63% | 61.47% | 55.96% |
| | t-stat | -9.626 | -6.387 | -8.952 | -5.461 |
| | significance | $>$ | $>$ | $>$ | $>$ |
| Average Minority Class *f-value* | Baseline | 0.644 | 0.644 | 0.652 | 0.652 |
| | Wrapper | 0.647 | 0.634 | 0.643 | 0.639 |
| | % increase | 0.49% | -1.61% | -1.38% | -1.90% |
| | t-stat | -0.128 | 0.398 | 0.484 | 0.727 |
| | significance | $\equiv$ | $\equiv$ | $\equiv$ | $\equiv$ |
| Average Majority class *f-value* | Baseline | 0.993 | 0.993 | 0.993 | 0.993 |
| | Wrapper | 0.992 | 0.991 | 0.991 | 0.991 |
| | % decrease | 0.16% | 0.21% | 0.21% | 0.18% |
| | t-stat | 3.678 | 3.923 | 5.674 | 4.224 |
| | significance | $<$ | $<$ | $<$ | $<$ |

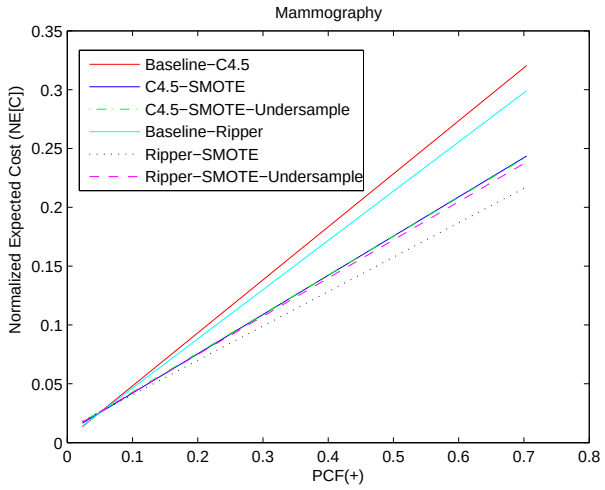


**Figure 2: Cost Curve for Mammography dataset.**

**Table 3: Results for the Forest Cover Data. $>$ indicates Wrapper is statistically significantly greater than Baseline; $<$ indicates Wrapper is statistically significantly lower than Baseline; and $\equiv$ indicates there is no statistically significant difference between the Wrapper and Baseline methods.**

| | | C4.5 | | Ripper | |
|---|---|---|---|---|---|
| | | SMOTE only | Undersampling and SMOTE | SMOTE only | Undersampling and SMOTE |
| | Average SMOTE % | 600% | 430% | 560% | 580% |
| | Average Under- sampling % | 100% | 99% | 100% | 93% |
| Average Minority Class TP-rate | Baseline | 0.873 | 0.873 | 0.834 | 0.834 |
| | Wrapper | 0.905 | 0.903 | 0.900 | 0.905 |
| | % increase | 3.59% | 3.28% | 7.34% | 7.87% |
| | t-stat | -4.889 | -4.223 | -7.544 | -7.532 |
| | significance | $>$ | $>$ | $>$ | $>$ |
| Average Minority Class FP-rate | Baseline | 0.0072 | 0.0072 | 0.009 | 0.009 |
| | Wrapper | 0.0109 | 0.0105 | 0.0133 | 0.014 |
| | % increase | 33.72% | 30.82% | 28.42% | 32.58% |
| | t-stat | -10.996 | -8.626 | -9.507 | -5.719 |
| | significance | $>$ | $>$ | $>$ | $>$ |
| Average Minority Class *f-value* | Baseline | 0.887 | 0.887 | 0.852 | 0.852 |
| | Wrapper | 0.884 | 0.885 | 0.868 | 0.866 |
| | % increase | -0.35% | -0.24% | 1.88% | 1.69% |
| | t-stat | 0.771 | 0.549 | -3.963 | -3.178 |
| | significance | $\equiv$ | $\equiv$ | $>$ | $>$ |
| Average Majority class *f-value* | Baseline | 0.992 | 0.992 | 0.989 | 0.989 |
| | Wrapper | 0.991 | 0.991 | 0.989 | 0.989 |
| | % decrease | 0.06% | 0.05% | -0.06% | -0.04% |
| | t-stat | 2.225 | 1.884 | -2.19 | -1.087 |
| | significance | $<$ | $<$ | $>$ | $\equiv$ |

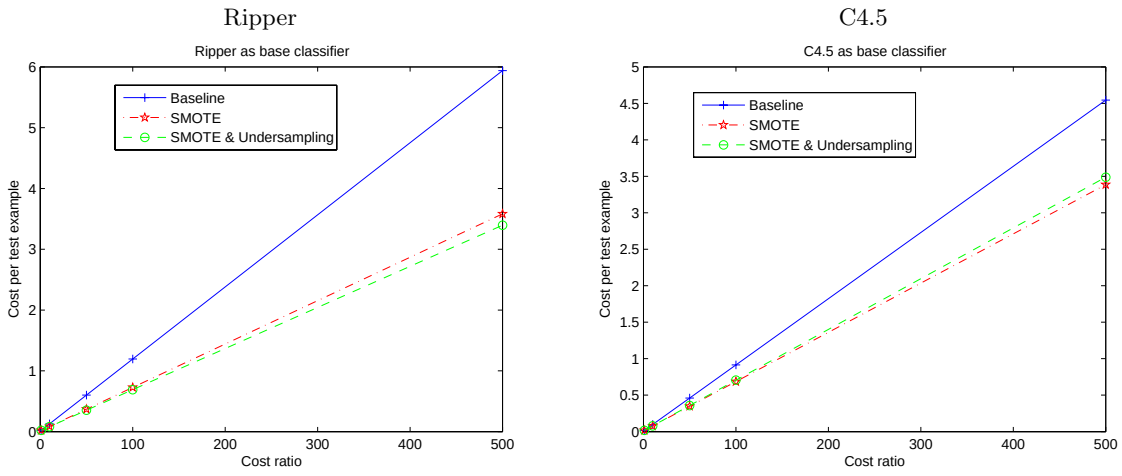


**Figure 3: Average Cost per test example at different cost ratios for the Forest Cover dataset.**

amples which were previously misclassified were correctly classified without increasing the number of majority class examples being classified as belonging to the minority class. The reason for these good results might be due to the similar distribution of the minority class examples in training and test data when cross-validation is performed. For example, in the forest cover dataset which contains 2,747 total minority class examples, the training data will contain approximately 2,472 examples while test data will contain 275 examples. Since there were a fair number of examples in the minority class SMOTE may have been more effective. It is unlike the mammography dataset where the number of minority class examples in the testing set is only 27.

We also looked at the Forest cover dataset under a cost framework. While, there weren't any obvious cost matrices that could be constructed, we simply utilized the following relative costs of $(C(+|-), C(-|+))$: (1, 1); (1, 10); (1, 50); (1,100); and (1,500). As evident in Figure 3, Ripper and C4.5 provide different performances as the cost matrices change for this dataset. Ripper is helped by undersampling, while C4.5 is not. This further justifies the use of wrapper techniques for different classifiers when considering sampling as a strategy for imbalanced datasets. Moreover, Ripper at (1,1) also benefited by SMOTE. Figure 4 shows the cost-curves across the range of $PCF(+)$. The wrapper based SMOTE and SMOTE-Undersampling for C4.5 and Ripper, respectively, produce the lowest expected for the broad range of $PCF(+)$. The choice of the classifier with the sampling methods doesn't seem to make a significant difference in the expected costs, while the individual classifiers are significantly apart in the cost space.

An interesting addition to our work will be analysis of the behavior of SMOTE and the rules thus constructed with both Ripper and C4.5. We would also investigate combination of the outputs of both the classifiers if they are making different kinds of errors to reduce the overall costs.

### 5.3 KDD-99 Cup Intrusion Dataset

This data set we treat differently. We look at it in a way that allows for comparisons with previous published work. A particular interesting example for comparison is to look at the results from of the KDD-99 cup data. A cost matrix was used in the scoring of the competition as shown in Table 4 [18]. It was used to produce the results in Table 7 below. There were many duplicate examples in the original 5 million example training set. All duplicate examples were removed. We also under-sampled both the normal and neptune (dos) class by removing examples which occurred only once. For Training Data 1 as in the Table 4, we under-sampled the normal class, and for the Training Data 2 we under-sampled both the normal and neptune classes. Note that for both these set of experiments, the test set remained unchanged. Our assumption was that some of them could be mislabeled or they were not very representative. These changes resulted in the training data set used here. Only SMOTE was applied to the modified data with the percentages for each minority class shown in Table 5.

It can be seen that our approach with RIPPER as the classifier produced the lowest cost per example after under-sampling both the normal and neptune classes (Training Data 2), and applying 100% SMOTE to the u2r class, while keeping the r2l class unchanged. This was better than the winner of the contest and better than the succeeding results from the literature. Even C4.5 as the base classifier with SMOTE (200% u2r and 300% for r2l) performed better than the other published techniques.

## 6. CONCLUSIONS

In this work, a wrapper [23] approach was utilized to determine the percentage of minority examples to add to the training set and the percentage to under-sample the majority class examples. The wrapper approach works by doing a guided search of the parameter space. The evaluation function was applied with a five fold cross validation done on the training set. Once the best percentages for under sampling and SMOTE are found it can be used to build a classifier on the updated training set and applied on the unseen testing set.

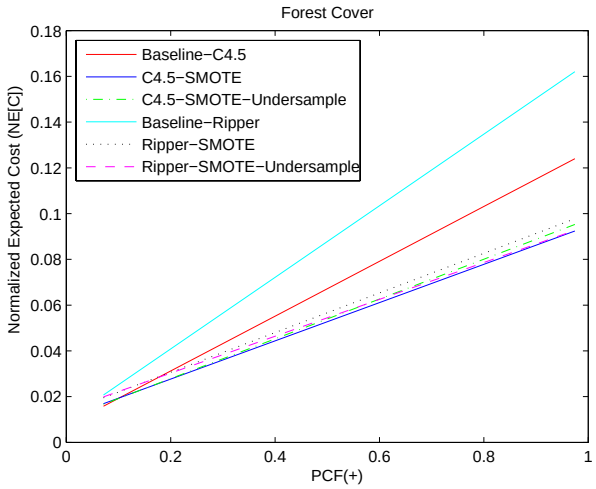


**Figure 4: Cost Curve for Forest Cover dataset.**

The *f-value* metric was used as the evaluation function. By using such a composite measure, we are able to control the relative increases in precision and recall, as both are essentially dependent on the different types of errors — false positives and false negatives. To statistically validate the results, we applied a 10-fold cross-validation framework to all but one of the datasets. Within each 10-folds, the wrapper utilized 5-fold cross-validation to identify the potentially optimal amounts of under-sampling and SMOTE.

We show results from applying this approach to the mammography dataset, the forest cover dataset, and the KDD-cup 99: Network Intrusion Detection dataset. Two learning algorithms were used, RIPPER a rule learning algorithm and C4.5 a decision tree learning algorithm. For the experiments, it was shown that it was possible to significantly increase the accuracy on the minority class, and reduce the overall expected costs. Our approach for imbalanced datasets significantly outperformed the baselines both in the true positive rate and the average cost per test example. Note that the *f-values* did not differ significantly because of the reduction in precision at the expense of the increase in recall. However, the relative increase in false positives does not impact the costs computation, because it is more costly to err as a false negative than a false positive. We achieved the lowest cost per test example of any approach

**Table 4: Cost matrix used for scoring entries in KDD CUP 99 competition.**

| Actual/predicted | dos | u2r | r2l | probe | normal |
|---|---|---|---|---|---|
| dos | 0 | 2 | 2 | 1 | 2 |
| u2r | 2 | 0 | 2 | 2 | 3 |
| r2l | 2 | 2 | 0 | 2 | 4 |
| probe | 2 | 2 | 2 | 0 | 1 |
| normal | 2 | 2 | 2 | 1 | 0 |

**Table 5: Comparison of results obtained on the original KDD CUP 99 test data. The numbers beside u2r and r2l indicate the SMOTE percentage utilized for the experiments.**

| | dos | u2r | r2l | probe | normal | Cost per test example |
|---|---|---|---|---|---|---|
| Winning Strategy [18] | 97.10% | 13.20% | 8.40% | 83.30% | 99.50% | 0.2331 |
| Decision Tree [5] | 96.57% | 13.60% | 0.45% | 77.92% | 99.43% | 0.2371 |
| Nave Bayes [5] | 96.65% | 10.96% | 8.66% | 88.33% | 97.68% | 0.2485 |
| Multi-classifier [31] | 97.30% | 29.80% | 9.60% | 88.70% | - | 0.2285 |
| Using C4.5 on Training Data 1 u2r (200) - r2l (0) | 97.08% | 14.47% | 1.21% | 93.52% | 97.87% | 0.2478 |
| Using RIPPER on Training Data 1 u2r (100) - r2l (0) | 97.45% | 22.37% | 6.96% | 81.64% | 96.18% | 0.2444 |
| Using C4.5 on Training Data 2 u2r (200) - r2l (300) | 99.41% | 14.47% | 7.39% | 93.61% | 97.34% | 0.2051 |
| Using RIPPER on Training Data 2 u2r (100) - r2l (0) | 97.33% | 19.74% | 13.73% | 91.98% | 95.62% | **0.2049** |

we know of for the intrusion detection data. We also introduced artificial costs for both the mammography and forest cover data. Our approach again produces lowest cost per test example, when compared to the baseline approach. It is very compelling that for the Forest cover dataset, our approach produces lower cost per test example even for (1, 1). Hence, the wrapper approach for automatically selecting the amounts of SMOTE with under-sampling is very promising. The proposed framework should be applicable to any sampling technique and evaluation measure.

In this paper, we did not include the costs in the *f-value* by varying the $\beta$ parameter to reflect the relative ratios. We believe that will be an interesting addition to our work. If the costs are known then they can expressed within validation framework for selecting the amounts of SMOTE and under-sampling quantities. We believe incorporating costs should again reduce the overall costs of the errors. The sampling quantities are also discovered using the same costs for both the classes as they will be used during evaluation. Thus, a stronger utilitarian framework can be implemented.

## 7. ACKNOWLEDGMENTS

Larry Hall was partially supported by the Department of Energy through the Advanced Strategic Computing Initiative (ASCI) Visual Interactive Environment for Weapons Simulation (VIEWS) Data Discovery Program Contract number: DEAC04-76DO00789. Nitesh Chawla was partially supported by National Science Foundation grant CNS-0130839, by the Central Intelligence Agency, and Department of Justice grant 2004-DD-BX-1224. We would like to thank Chris Drummond for his helpful input on the cost curves. We would also like to thank the anonymous reviewers for their useful comments.

## 8. REFERENCES

[1] In T. Dietterich, D. Margineantu, F. Provost, and P. Turney, editors, *Proceedings of the ICML'2000 Workshop on COST-SENSITIVE LEARNING*. 2003.

[2] In N. V. Chawla, N. Japkowicz, and A. Kołcz, editors, *Proceedings of the ICML'2003 Workshop on Learning from Imbalanced Data Sets*. 2003.

[3] In N. V. Chawla, N. Japkowicz, and A. Kolcz, editors, *SIGKDD Explorations Special Issue on Learning from Imbalanced Datasets*. SIGKDD, 2004.

[4] In C. Ferri, P. Flach, J. Orallo, and N. Lachice, editors, *ECAI' 2004 First Workshop on ROC Analysis in AI*. ECAI, 2004.

[5] N. B. Amor, S. Benferhat, and Z. Elouedi. Naive Bayes vs. Decision Trees in Intrusion Detection Systems. In *Proceedings of the ACM Symposium on Applied Computing*, pages 420–424, 2004.

[6] R. E. Banfield, L. O. Hall, K. W. Bowyer, and W. P. Kegelmeyer. Ensembles of Classifiers from Spatially Disjoint Data. In *Proceedings of the Sixth International Conference on Multiple Classifier Systems*, 2005.

[7] G. E. A. P. A. Batista, R. C. Prati, and M. C. Monard. A study of the behavior of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1), 2004.

[8] C. Blake and C. Merz. UCI Repository of Machine Learning Databases. Department of Information and Computer Sciences, University of California, Irvine, 1998.

[9] K. W. Bowyer, L. O. Hall, N. V. Chawla, and T. E. Moore. A parallel Decision Tree Builder for Mining Very Large Visualization Datasets. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, 2000.

[10] M. Buckland and F. Gey. The Relationship Between Recall and Precision. *Journal of the American Society for Information Science*, 45:12–19, 1994.

[11] P. Chan, W. Fan, P. Prodromodis, and S. Stolfo. Distributed Data Mining in Credit Card Fraud Detection. *IEEE Intelligent Systems*, 14(6):67–74, 1999.

[12] N. V. Chawla. C4.5 and imbalanced datasets: Investigating the effect of ampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the ICML'03 Workshop on Class Imbalances*, 2003.

[13] N. V. Chawla. Editorial: Learning from Imbalanced Datasets. *SIGKDD Explorations*, 6(1), 2004.

[14] N. V. Chawla, L. O. Hall, B. K. W., and W. P. Kegelmeyer. SMOTE: Synthetic Minority Oversampling TEchnique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[15] W. W. Cohen. Fast Effective Rule Induction. In *Proc. 12th International Conference on Machine Learning*, pages 115–123, Lake Tahoe, CA, 1995. Morgan Kaufmann.

[16] C. Drummond and R. Holte. Explicitly representing expected cost: An alternative to ROC representation. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 198–207, 2001.

[17] C. Drummond and R. Holte. C4.5, class imbalance, and cost sensitivity: Why under-sampling beats over-sampling. In *Proceedings of the ICML'03 Workshop on Learning from Imbalanced Data Sets*, 2003.

[18] C. Elkan. Results of the kdd'99 Classifier Learning Contest. http://www.cse.ucsd.edu/∼elkan/clresults.html, 1999.

[19] J. Ezawa, K., M. Singh, and W. Norton, S. Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management. In *Proceedings of the International Conference on Machine Learning, ICML-96*, pages 139–147, Bari, Italy, 1996. Morgan Kauffman.

[20] S. Hettich and S. D. Bay. The UCI KDD Archive [http://kdd.ics.uci.edu]. Department of Information and Computer Sciences, University of California, Irvine, 1998.

[21] N. Japkowicz. The Class Imbalance Problem: Significance and Strategies. In *Proceedings of the 2000 International Conference on Artificial Intelligence (IC-AI'2000): Special Track on Inductive Learning*, Las Vegas, Nevada, 2000.

[22] N. Japkowicz and S. Stephen. The Class Imbalance Problem: A Systematic Study. *Intelligent Data Analysis*, 6(5):203–231, 2002.

[23] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[24] M. Kubat, R. Holte, and S. Matwin. Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, 30:195–215, 1998.

[25] M. Kubat and S. Matwin. Addressing the Curse of Imbalanced Training Sets: One Sided Selection. In *Proceedings of the Fourteenth International Conference on Machine Learning*, pages 179–186, Nashville, Tennesse, 1997. Morgan Kaufmann.

[26] B. S. Lee, R. R. Snapp, and R. Musick. Toward a Query Language on Simulation Mesh Data: An Object Oriented Approach. In *Proceedings of the International Conference on Database Systems for Advanced Applications*, 2001.

[27] C. Ling and C. Li. Data Mining for Direct Marketing Problems and Solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, New York, NY, 1998. AAAI Press.

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

[29] F. Provost and T. Fawcett. Analysis and Visualization of Classifier Performance: Comparison under Imprecise Class and Cost Distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining*, pages 43–48, New Port Beach, CA, 1997. AAAI Press.

[30] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann, San Mateo, CA, 1992.

[31] M. R. Sabhnani and G. Serpen. Application of Machine Learning Algorithms to KDD Intrusion Detection Dataset with Misuse Detection Context. In *Proceedings of the International Conference on Machine Learning: Models, Technologies, and Applications*, pages 209–215, 2003.

[32] G. Weiss and F. Provost. Learning when Training Data are Costly: The Effect of Class Distribution on Tree Induction. *Journal of Artificial Intelligence Research*, 19:315–354, 2003.

[33] K. Woods, C. Doss, K. Bowyer, J. Solka, C. Priebe, and P. Kegelmeyer. Comparative Evaluation of Pattern Recognition Techniques for Detection of Microcalcifications in Mammography. *International Journal of Pattern Recognition and Artificial Intelligence*, 7(6):1417–1436, 1993.

[34] B. Zadrozny and C. Elkan. Learning and Making Decisions When Costs and Probabilities are Both Unknown. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 204–213, 2001.