Does Cost-Sensitive Learning Beat Sampling for Classifying Rare Classes?

Kate McCarthy, Bibi Zabar and Gary Weiss
Fordham University
441 East Fordham Road
Bronx, NY 10458
creed112@aol.com, zabar@fordham.edu, gweiss@cis.fordham.edu

ABSTRACT

A highly-skewed class distribution usually causes the learned classifier to predict the majority class much more often than the minority class. This is a consequence of the fact that most classifiers are designed to maximize accuracy. In many instances, such as for medical diagnosis, the minority class is the class of primary interest and hence this classification behavior is unacceptable. In this paper, we compare two basic strategies for dealing with data that has a skewed class distribution and non-uniform misclassification costs. One strategy is based on cost-sensitive learning while the other strategy employs sampling to create a more balanced class distribution in the training set. We compare two sampling techniques, up-sampling and down-sampling, to the cost-sensitive learning approach. The purpose of this paper is to determine which technique produces the best overall classifier—and under what circumstances.

Categories and Subject Descriptors

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

General Terms

Algorithms

Keywords

Cost-sensitive learning, sampling, data mining, induction, decision trees, rare classes, class imbalance

1. INTRODUCTION

In many real-world domains, such as fraud detection and medical diagnosis, the class distribution of the data is skewed and the cost of misclassifying the minority class is substantially greater than the cost of misclassifying the majority class. In these cases, it is important to create a classifier that minimizes the overall misclassification cost. This tends to cause the classifiers to perform better on the minority class than if the misclassification costs were equal. For highly skewed class distribution, this also ensures that the classifier does not only predict the majority class.

The most direct method for dealing with highly skewed class distributions with unequal misclassification costs is to use cost-sensitive learning. An alternate strategy for dealing with skewed data with non-uniform misclassification costs is to use sampling to alter the class distribution of the training data so that the resulting training set is more balanced. There are two basic sampling methods for achieving a more balanced class distribution: up-sampling and down-sampling (also referred to as over-sampling and under-sampling). In this context, up-sampling replicates minority class examples and down-sampling discards majority class examples.

This paper compares cost-sensitive learning, up-sampling, and down-sampling to determine which method leads to the best overall classifier performance, where the best overall classifier is the one that minimizes total cost. Since sampling is often used instead of cost-sensitive learning in practice, we compare these methods to see which yields better results. Our conjecture is that cost-sensitive learning will outperform both up-sampling and down-sampling because of well-known problems (described in the next section) with these sampling methods. We evaluate this conjecture using C5.0 [18], a more advanced version of Quinlan's popular C4.5 program. We also evaluate this conjecture for data sets that are not skewed (but have non-uniform misclassification costs) to broaden the scope of our study. We compare cost-sensitive learning only to the basic up-sampling and down-sampling methods because these are the only methods available to most practitioners (some of the variants developed by researchers to address the weaknesses with sampling are discussed in Section 7).

2. BACKGROUND

In this section we provide basic background information on cost-sensitive learning, sampling, and the connection between the two. Some related work is also described.

2.1 Cost-Sensitive Learning

In this paper we focus our attention on two-class learning problems. The behavior of a classifier for such problems can be described by a confusion matrix. Figure 1 provides the terminology for such a confusion matrix. Holding with established practice, the positive class is the minority class and the negative class is the majority class.

UBDM '05, August 21, 2005, Chicago, Illinois, USA.
Copyright 2005 ACM 1-59593-208-9/05/0008...$5.00.

	ACTUAL		
PREDICTED		Positive class	Negative class
	Positive class	True positive (TP)	False positive (FP)
	Negative class	False negative (FN)	True negative (TN)

Figure 1: A Confusion Matrix

Corresponding to a confusion matrix is a cost matrix. The cost matrix will provide the costs associated with the four outcomes shown in the confusion matrix, which we refer to as C_{TP}, C_{FP}, C_{FN}, and C_{TN}. As is often the case in cost-sensitive learning, we assign no costs to correct classifications, so C_{TP} and C_{TN} are set to 0. Since the positive (minority) class is often more interesting than the negative (majority) class, typically $C_{FN} > C_{FP}$ (note that a false negative means that a positive example was misclassified).

A cost-sensitive learner can accept cost information from a user and assign different costs to different types of misclassification errors. Learners can implement cost-sensitive learning in a variety of ways. One common method is to alter the class probability thresholds used to assign the classification value. For example, in a decision tree learner the probability threshold associated with a terminal node is typically set to 0.5, so that the node is labeled with the most probable class. If the ratio of misclassification costs for a two-class problem is set to 2:1, then the class probability threshold would be 0.33 [9, 17]. Note that in this implementation of cost-sensitive learning no data is discarded or replicated.

When misclassification costs are known or can be assumed the best metric to evaluate overall classifier performance is total cost. Total cost is the only evaluation metric used in this paper and is used to evaluate the results for both cost-sensitive learning and sampling. The formula for total cost is shown below, in equation 1.

$$\text{Total Cost} = (FN \times C_{FN}) + (FP \times C_{FP}) \qquad [1]$$

2.2 Sampling

Sampling can be used to alter the class distribution of the training data. As described earlier, this can be accomplished via up-sampling or down-sampling. Both sampling methods have been used to deal with skewed class distributions [1, 2, 3, 6, 10, 11]. The reason that altering the class distribution of the training data aids learning with highly-skewed data sets is that it effectively imposes non-uniform misclassification costs. For example, if one alters the class distribution of the training set so that the ratio of positive to negative examples goes from 1:1 to 2:1, then one has effectively assigned a misclassification cost ratio of 2:1. This equivalency between altering the class distribution of the training data and altering the misclassification cost ratio is well known and was formally established by Elkan [9].

Previous research on learning with skewed class distributions has altered the class distribution using up-sampling and down-sampling. There are disadvantages to using sampling to implement cost-sensitive learning, however. The disadvantage with down-sampling is that it discards potentially useful data. There are two disadvantages with up-sampling. First, it increases the size of the training set, which will increase the time necessary to learn the classifier. Second, since most up-sampling methods generate exact copies of existing examples, overfitting is likely to occur in that classification rules may be formed to cover a single, replicated example.

2.3 Why Use Sampling?

Given the disadvantages with sampling, it is worth asking why anyone would use sampling to deal with highly-skewed class distributions (with non-uniform misclassification costs) when cost-sensitive learning appears to be a more direct solution. In this section, we discuss several reasons for this. The most obvious reason is that many learning algorithms are not cost-sensitive and therefore a wrapper approach, like the one using sampling, is the only option. This is certainly less true today than in the past, but many of the older non-commercial learners still provide no mechanism for cost sensitive learning.

A second reason is that many highly skewed data sets are enormous and therefore require the size of the training set to be reduced. In this case, down-sampling seems to be a reasonable, and valid, strategy. In this paper, we do not consider the need to reduce the training set size. We would point out, however, that if one needs to discard some training data, it still might be beneficial to discard some of the majority class examples in order to reduce the training set size to the required size, and then also employ cost-sensitive learning, so that the amount of training data is not reduced beyond what is absolutely necessary.

A final reason one might give for using sampling instead of cost-sensitive learning is that the misclassification costs are often not known. This is not a valid reason for using sampling over cost-sensitive learning, however, since the same issue arises with sampling—what is the proper sampling rate? Ideally, the sampling rate should be based on the cost information. If that is not available, one might try various sampling rates and look at the performance of the induced classifier. However, the same strategy can be employed with cost-sensitive learning—various cost ratios can be evaluated and one can select the cost ratio based on the observed performance characteristics of the induced classifier. Alternatively, if misclassification costs are not known one can evaluate the performance of a classifier over a range of costs by using ROC analysis.

Overall, we feel that the only reason to use sampling to handle skewed class distributions is if the amount of available training data cannot be handled by the learning algorithm. Otherwise, our conjecture is that cost-sensitive learning should be used. We evaluate this conjecture in this paper.

3. DATA SETS

We used a total of fourteen data sets in our experiments. Twelve of the data sets were obtained from the UCI Repository and two of the data sets came from AT&T and were used in previously published work done by Weiss and Hirsh [16]. A summary of these data sets is provided in Table 1. The data sets are listed in descending order according to class imbalance (the most imbalanced data sets are listed first). The data sets marked with an asterisk (*) were originally multi-class data sets that were previously mapped into two classes for work done by Weiss and Provost [17]. The letter-a and letter-vowel data sets are derived from the letter recognition data set that is available from the UCI Repository.

The data sets were chosen on the basis of their class distributions and data set sizes. Although the main focus of our research is to compare cost-sensitive learning and sampling for classifying rare classes in imbalanced data sets, we also included a few data sets with more balanced class distributions to see if and how the overall results would differ. The boa1, promoters, and coding data sets each had an evenly balanced 50-50 distribution, so they were used for the sake of comparison. We used data sets of varying sizes to see how this would affect our results. One would expect that cost-sensitive learning would outperform down-sampling for small data sets, since throwing away any data in this situation should be harmful.

Since these data sets do not come with misclassification cost information, we evaluated the cost-sensitive and sampling strategies using a wide variety of costs. This is described in detail in the next section.

Table 1: Data Set Summary

Data Set	% Minority	Total Examples
Letter-a*	4%	20,000
Pendigits*	8%	13,821
Connect-4*	10%	11,258
Bridges1	15%	102
Letter-vowel*	19%	20,000
Hepatitis	21%	155
Contraceptive	23%	1,473
Adult	24%	21,281
Blackjack	36%	15,000
Weather	40%	5,597
Sonar	47%	208
Boa1	50%	11,000
Promoters	50%	106
Coding	50%	20,000

4. EXPERIMENTS

In this section we begin by describing C5.0, the learner used for our experiments. We then describe our experimental methodology for using cost-sensitive learning and sampling.

4.1 C5.0

All of our experiments utilize C5.0 [18], a commercial classifier induction program, which is a more advanced version of Quinlan's popular C4.5 and ID3 learners [14, 15]. Unlike these older programs, C5.0 supports cost-sensitive learning.

Both the cost-sensitive learning and sampling experiments used 75% of the data for training and 25% for testing. Each experiment was run ten times, using random sampling to create these two data sets. All results shown in this paper are the averages of these ten runs. Classifiers are evaluated using total cost, which was defined earlier in equation 1.

4.2 Cost-Sensitive Learning

In our experiments, we are interested in targeting the cases where the cost of incorrectly classifying a minority (positive) class example will have a higher cost than the cost of incorrectly classifying a majority (negative) class example. Hence we applied a higher misclassification cost to CFN, the cost of a false negative misclassification. For our experiments, a false positive prediction, CFP, was assigned a cost of 1, while CFN was allowed to vary. For the majority of the experiments CFN was evaluated for the values: 1, 2, 3, 4, 6, and 10, although for some experiments the costs were allowed to increase beyond this point.

4.3 Sampling

Up-sampling and down-sampling were used to implement the desired misclassification cost ratios, as described in Section 2.2. Since C5.0 does not provide the necessary support for sampling, the required sampling was done external to C5.0 and the resulting sampled training data was then passed to C5.0. No changes were made to the test data, but none were necessary since the resulting classifiers were evaluated using total cost, based on the cost information associated with each experiment. The misclassification cost ratios used for sampling were the same ones for cost-sensitive learning. Note that the greater cost ratio, the more training examples had to be discarded when down-sampling. The test set size was held fixed for all experiments.

5. RESULTS

Classifiers were generated for each data set using cost-sensitive learning, up-sampling and down-sampling for a variety of misclassification cost ratios. These classifiers were evaluated using total cost. We generated one figure for each of the fourteen data sets, showing how the total cost varies when cost-sensitive learning, up-sampling and down-sampling are used. Some of these figures are included in this section while the remaining figures can be found in the Appendix. After presenting these detailed results for each data set, summary statistics are provided which make it easier to compare and contrast the cost-sensitive learning method with the two sampling methods.

The results for the letter-a data set in Figure 2 show that cost-sensitive learning and up-sampling performed similarly whereas down-sampling performed much worse for all cost ratios (note that all methods will perform the same for 1:1). The letter-vowel data set, shown in Figure A1 in the Appendix, provides nearly identical results except that cost-sensitive learning performed slightly better than up-sampling for most cost ratios (both still outperform down-sampling).

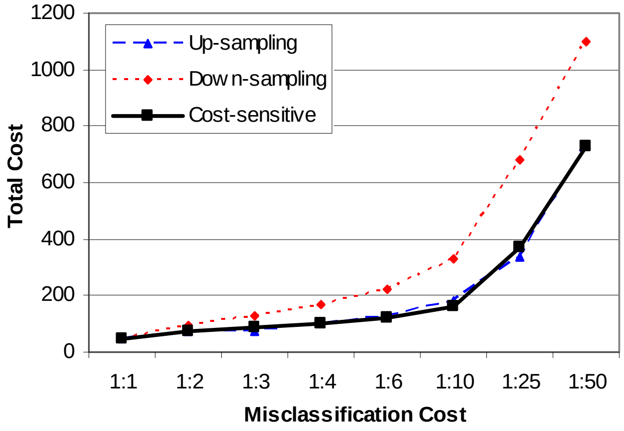

Figure 2: Results for Letter-a

The results for the weather data set, provided in Figure 3, show that up-sampling consistently performed much worse than down-sampling and cost-sensitive learning, both of which performed similarly. This *exact* same pattern also occurs in the results for the adult and boa1 data sets, which are provided in Figures A2 and A3, respectively, in the Appendix.

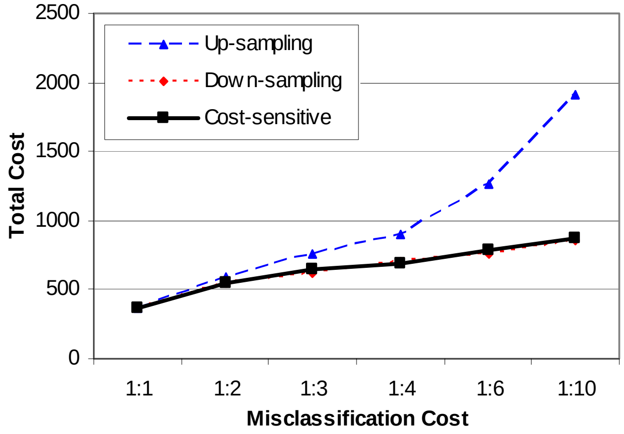

Figure 3: Results for Weather

The results for the coding data set in Figure 4 show that cost-sensitive learning outperformed both sampling methods, although the difference in total cost is much greater when compared to up-sampling. As we shall see shortly in Table 3, however, cost-sensitive learning still outperforms down-sampling by 9%, a substantial amount (it outperforms up-sampling by 20%).

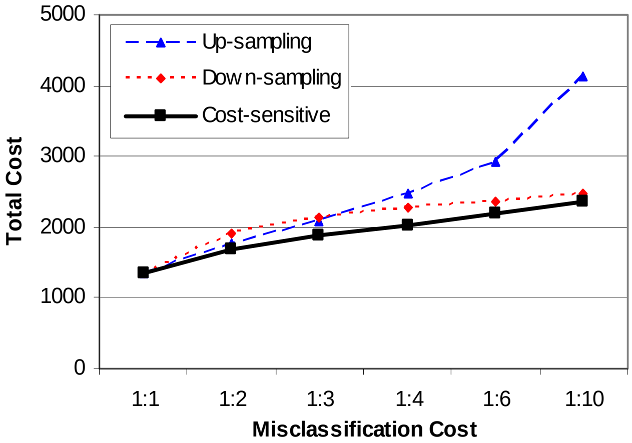

Figure 4: Results for Coding

The blackjack data set, shown in Figure 5, is the only data set for which all three methods yielded nearly identical performance for all cost ratios. The connect-4 data set (Figure A4) yielded nearly identical costs for all three methods as well, except for the highest cost ratio, 1:25, in which case up-sampling performed the worst.

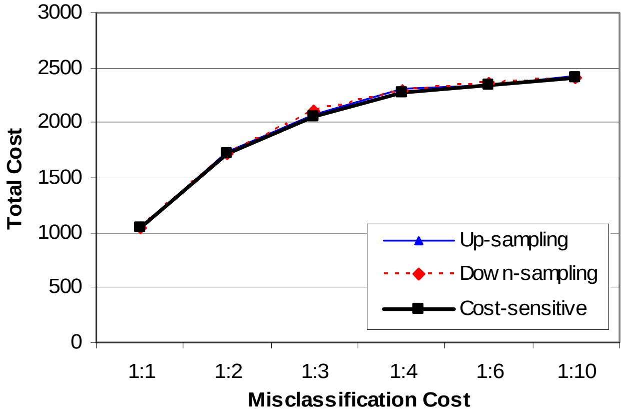

Figure 5: Results for Blackjack

There were three data sets for which the cost-sensitive method underperformed the two sampling methods for most cost ratios. This occurred for the contraceptive, hepatitis, and bridges1 data sets. The results for the contraceptive data set are shown in Figure 6, while the results for the hepatitis data set and bridges1 data set can be found in Figures A5 and A6 in the Appendix.

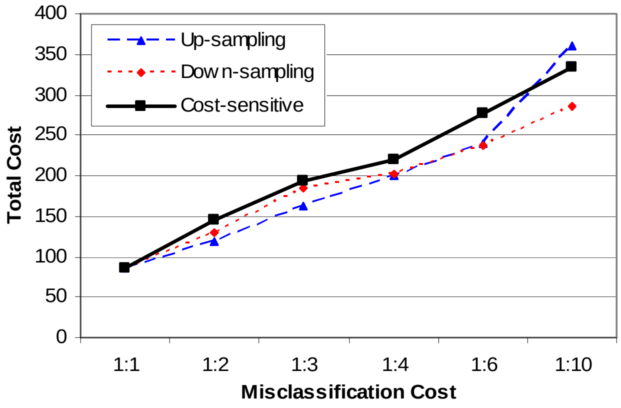

Figure 6: Results for Contraceptive

The sonar data set (Figure A7) is the only data set for which down-sampling consistently beats both the cost-sensitive and up-sampling method. The promoters data set (Figure A8) is the only data set for which up-sampling consistently beat the other two methods. We previously noted that the coding data set (Figure 4) is the only one in which the cost-sensitive method consistently beat the two sampling methods. Thus, we see that it is quite rare for any of the three methods to beat both of the other two methods—although it is common for each to beat one of the other methods. The only data set not yet discussed is the pendigits data set (Figure A9). Overall, the cost-sensitive learning method tends to beat both sampling methods for this data set, although the results vary by cost ratio.

Tables 2 and 3 summarize the performance of up-sampling, down-sampling, and cost-sensitive learning for all fourteen data sets. Table 2 specifies the first/second/third place finishes over the evaluated cost ratios for each data set and method. For example, Table 2 shows that for the letter-a data set up-sampling generates the best results (i.e., lowest total cost) for 4 of the 7 evaluated cost ratios and the second best result for 3 of the 7 cost ratios.

Table 2: First/Second/Third Place Finishes

Data Set	Up-sampling	Down-sampling	Cost-Sensitive
Letter-a	4/3/0	0/0/7	3/4/0
Pendigits	3/1/3	1/2/4	3/4/0
Connect-4	2/0/3	0/3/2	3/2/0
Bridges1	5/0/0	0/5/0	0/3/2
Letter-vowel	4/1/0	0/0/5	1/4/0
Hepatitis	3/2/0	2/3/0	0/5/0
Contraceptive	3/2/0	2/3/0	0/1/4
Adult	2/3/0	3/1/1	0/4/1
Blackjack	1/1/3	2/1/2	3/2/0
Weather	0/0/5	4/1/0	1/4/0
Sonar	2/3/0	3/2/0	0/2/3
Boa1	0/0/5	4/1/0	2/3/0
Promoters	5/0/0	0/2/3	0/3/2
Coding	0/2/3	0/3/2	5/0/0
Total	33/18/22	21/27/26	21/41/12

The problem with Table 2 is that it does not quantify the improvements—the reduction in total cost. It treats all "wins" as equal even if the difference in costs between the methods is quite small. Table 3 remedies this by providing the relative reduction in cost for the strategies. The second and third columns compare cost-sensitive learning (abbreviated "Cost") versus up-sampling and down-sampling, respectively. The last column compares up-sampling to down-sampling. A negative value indicates an increase in cost rather than a reduction in cost. As an example, the results in Table 3 for the letter-a data set indicate that cost-sensitive learning performs slightly worse than up-sampling (-0.9%) but much better than down-sampling (37.9%) and that up-sampling performs much better than down-sampling (38.4%).

Table 3: Comparison of Relative Improvements

Data Set	Cost vs. Up-Sampling	Cost vs. Down-Sampling	Up- vs. Down-Sampling
Letter-a	-0.9%	37.9%	38.4%
Pendigits	3.5%	5.4%	0.9%
Connect-4	3.2%	-0.1%	-3.9%
Bridges1	-38.4%	-8.6%	21.2%
Letter-vowel	-7.7%	18.0%	23.7%
Hepatitis	-11.4%	-8.2%	2.3%
Contraceptive	-11.9%	-11.6%	-0.9%
Adult	8.7%	-0.8%	-12.0%
Blackjack	0.5%	0.5%	0.0%
Weather	27.9%	-1.3%	-50.0%
Sonar	-0.9%	-23.8%	-33.78%
Boa1	17.6%	-0.6%	-30.0%
Promoters	-40.6%	-1.2%	28.2%
Coding	20.0%	9.1%	-18.0%
Ave Savings	**-2.2%**	**1.1%**	**-2.4%**
Total Wins	**7**	**6**	**6**

The results from Table 2 and Table 3 show that cost-sensitive learning, as implemented in C5.0, does not consistently beat both or either of the sampling methods. Furthermore, none of three methods is a clear winner over all, or either, of the other methods. Overall, up-sampling seems to perform the best, by a relatively small margin, followed by cost-sensitive learning, with down-sampling doing the worst (based on total average savings). However, the results vary widely for each of the data sets. The best way to characterize the overall performance of the cost-sensitive approach based on Table 2 is that it rarely performs the worst. Even up-sampling, which performs the best overall, comes in last many more times (22 versus 12). Thus, one conclusion is that performance of cost-sensitive learning does not fluctuate quite as much as the sampling methods, over the different data sets.

6. DISCUSSION

Based on the results from all of the data sets, there was no definitive winner between cost-sensitive learning, up-sampling and down-sampling. Given that there is no clear and consistent winner, the logical question to ask is whether we can characterize under what circumstances each method performs best. We begin by analyzing the impact of data set size. Our study included four data sets (bridges1, hepatitis, sonar, and promoters) that are substantially smaller than the rest. If we compute the first/second/third place records for these four data sets from Table 2, we get the following results: up-sampling *15/5/0*, down-sampling *5/12/3* and cost-sensitive learning *0/13/7*. Based on this data, up-sampling clearly does much better than down-sampling and cost-sensitive learning. The data in Table 2 also supports this conclusion. The one exception is the sonar data set, where down-sampling beats up-sampling.

With the exception of the sonar results, the sampling results make sense. That is, we expect down-sampling, which throws away data, to perform more poorly than up-sampling for small data sets. The data also implies that up-sampling also outperforms cost-sensitive learning in these cases, however. One possible explanation for the failure of cost-sensitive learning in this situation is that when there is very little training data, it will be difficult to accurately estimate the class-membership probabilities—something that is required in order to get good results from cost-sensitive learning.

If we look at the eight data sets with over 10,000 examples each (letter-a, pendigits, connect-4, letter-vowel, adult, blackjack, boa, and coding), our results are as follows for first/second/third place finishes: up-sampling *16/11/17*, down-sampling *10/11/2*, and cost-sensitive *20/23/1*. The results from Table 3 show that over these eight data sets the average improvement between cost-sensitive learning and up-sampling is 5.5% and between cost-sensitive learning and down-sampling is 5.7%. Thus, for the large data sets, cost-sensitive learning does often yield the best results. Perhaps cost-sensitive learning does well in these cases because the larger amount of training data makes it easier to more accurately estimate the class-membership probabilities.

Another factor worth considering is the degree to which the class distribution of the data set is unbalanced. This will impact the extent to which sampling must be used to get the desired distribution. The results in Tables 2 and 3, which are ordered by decreasing class imbalance, show no obvious pattern, however.

Our results do not generally support our conjecture that cost-sensitive learning should outperform sampling for obtaining the best classifier performance. However, the results tend to indicate that the conjecture may hold for larger data sets. This suggests that perhaps cost-sensitive learning performs well only when there are sufficient data to generate accurate probability estimates (for c5.0 this translates to having many examples at each leaf node). We have found some supporting evidence to suggest why cost-sensitive learning is not a clear winner in all cases. Recent research [7] has shown that cost-sensitive learning, including C5.0's implementation of cost-sensitive learning, does not always produce the desired, and expected, results. Specifically, this research showed that one can achieve lower total cost by using a cost ratio for learning that is different from the actual cost information. This tends to indicate that there may be a problem with the cost-sensitive learning process.

7. RELATED WORK

Previous research has compared cost-sensitive learning and sampling. The experiments that we performed are similar to the work that was done by Chen, Liaw, and Breiman [6], who proposed two methods of dealing with highly-skewed class distributions based on the Random Forest algorithm. Balanced Random Forest (BRF) uses down-sampling of the majority class to create a training set with a more equal distribution between the two classes, whereas Weighted Random Forest (WRF) uses the idea of cost-sensitive learning. By assigning a higher misclassification cost to the minority class, WRF improves classification performance of the minority class and also reduces the total cost. However, although both BRF and WRF outperform existing methods, the authors found that neither one is consistently superior to the other. Thus, the cost-sensitive version of the Random Forest does not outperform the version than employs down-sampling.

Drummond and Holte [8] found that down-sampling outperforms up-sampling for skewed class distributions and non-uniform cost ratios. Their results indicate that this is because up-sampling shows little sensitivity to changes in misclassification cost, while down-sampling shows reasonable sensitivity to these changes. Breiman et al. [2] analyzed classifiers produced by sampling and by varying the cost matrix and found that these classifiers were indeed similar. Japkowicz and Stephen [10] found that cost-sensitive learning outperforms under-sampling and over-sampling, but only on artificially generated data sets. Maloof [12] also compared cost-sensitive learning to sampling but found that cost-sensitive learning, up-sampling and down-sampling performed nearly identically. However, because only a single data set was analyzed, one really could not draw any general conclusions from that data. Since we analyzed fourteen real-world data sets, we believe our research extends this earlier work and provides the most conclusive evidence that cost-sensitive learning does not clearly, or consistently, outperform up-sampling or down-sampling.

8. CONCLUSION

The results from our study indicate that between cost-sensitive learning, up-sampling, and down-sampling, there is no clear or consistent winner for maximizing classifier performance when cost information is known. If we focus exclusively on large data sets with more than 10,000 total examples, however, it appears that cost-sensitive learning often outperforms the sampling methods—although it still does not happen in every case. Note that in this study our focus was on using the cost information to improve the performance of the minority class, but in fact our results are much more general; they can be used to assess the relative performance of the three methods for implementing cost-sensitive learning. Our results also allow us to compare up-sampling to down-sampling. We found that up-sampling performed better than down-sampling overall, although the behavior varies widely for each data set.

There are a variety of enhancements that people have made to improve the effectiveness of sampling. While these techniques have been compared to up-sampling and down-sampling, they generally have not been compared to cost-sensitive learning. This would be worth studying in the future. Some of these enhancements include introducing new "synthetic" examples when up-sampling [5], deleting less useful majority-class examples when down-sampling [11] and using multiple sub-samples when down-sampling such than each example is used in at least one sub-sample [3].

In our research, we plotted classifier performance for different cost ratios and then summarized the results by recording the number of first/second/third place finishes for each method and also by averaging the results. We did this based on the assumption that the actual cost information will be known or can be estimated. This is not always the case and the reporting of our results could benefit by using other methods, such as ROC analysis or cost curves.

The implications of this research are significant. The fact that sampling, a wrapper approach, performs competitively—if not better—than a commercial tool that implements cost-sensitivity raises several important questions. These questions are: 1) why doesn't the cost-sensitive learner perform better given the known drawbacks with sampling, 2) are there ways we can improve cost-sensitive learners and 3) are we better off not using the cost-sensitivity features of a learner and using sampling instead. We hope to address these questions in future research.

9. REFERENCES

[1] Abe, N., Zadrozny, B., and Langford, J. An iterative method for multi-class cost-sensitive learning. *KDD '04*, August 22-25, 2004, Seattle, Washington, USA, 2004.

[2] Breiman, E., J Friedman, R. Olshen and C. Stone. *Classification and Regression Trees*. Belmont, CA: Wadsworth International Group, 1984.

[3] Chan, P., and Stolfo, S. Toward scalable learning with non-uniform cost and class distributions: a case study in credit card fraud detection. *American Association for Artificial Intelligence*, 1998.

[4] Chawla, N. C4.5 and imbalanced datasets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. *ICML 2003 Workshop on Imbalanced Datasets*.

[5] Chawla, N.V., Bowyer, K.W., Hall, L.O., Kegelmeyer, W.P. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, Volume 16, 321-357, 2002.

[6] Chen C., Liaw, A., and Breiman, L. Using random forest to learn unbalanced data. Technical Report 666, Statistics Department, University of California at Berkeley, 2004. <http://www.stat.berkeley.edu/users/chenchao/666.pdf>

[7] Ciraco, M., Rogalewski, M., and Weiss, G. Improving classifier utility by altering the misclassification cost ratio. *Proceedings of the KDD-2005 Workshop on Utility-Based Data Mining*.

[8] Drummond, C., and Holte, R. C4.5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. *Workshop on Learning from Imbalanced Data sets II, ICML,* Washington DC, 2003.

[9] Elkan, C. The foundations of cost-sensitive learning. *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence*, 2001.

[10] Japkowicz N, and Stephen, S. The class imbalance problem: a systematic study. *Intelligent Data Analysis Journal*, 6(5), 2002.

[11] Kubat, M and Matwin, S. Addressing the curse of imbalanced training sets: one-sided selection. *Proceedings of the Fourteenth International Conference on Machine Learning*, 179-186, 1997.

[12] Maloof, M. Learning when data sets are imbalanced and when costs are unequal and unknown. *ICML 2003 Workshop on Imbalanced Datasets*.

[13] Pednault, E., Rosen, B., and Apte, C. The importance of estimation errors in cost-sensitive learning. *IBM Research Report RC-21757*, May 30, 2000.

[14] Quinlan, J.R. C4.5: programs for machine learning. Morgan Kaufmann Publishers Inc., San Francisco, CA, 1993.

[15] Quinlan, J.R. Induction of decision trees. Machine Learning 1: 81-106, 1986.

[16] Weiss, G., and Hirsh, H. A quantitative study of small disjuncts. *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, 2000.

[17] Weiss, G., and Provost, F. Learning when training data are costly: the effect of class distribution on tree induction. *Journal of Artificial Intelligence Research*, 2003.

[18] "Data Mining Tools See5 and C5.0." RuleQuest, Nov. 2004. RuleQuest Research. May 13, 2005. <http://www.rulequest.com/see5-info.html>

APPENDIX

The results for the letter-vowel data set in Figure A1 show that up-sampling performed better than cost-sensitive learning for some cost ratios. Furthermore, both up-sampling and cost-sensitive learning perform better than down-sampling.

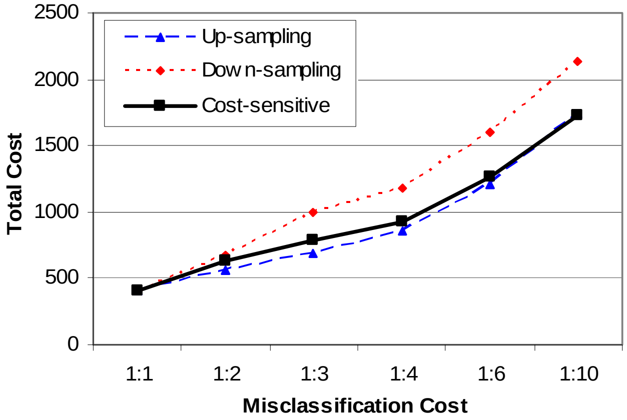

Figure A1: Results for Letter-vowel

The results for the adult data set in Figure A2 and the boa1 data set in Figure A3 both have up-sampling performing much worse than down-sampling and cost-sensitive learning, both of which perform similarly. These results mimic those of the weather data set in Figure 3 in the main body of this paper.

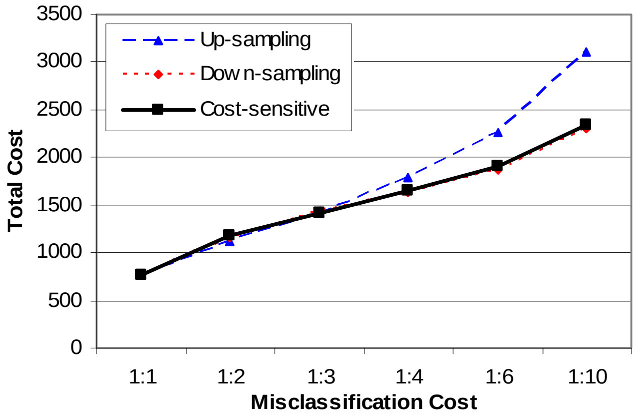

Figure A2: Results for Adult

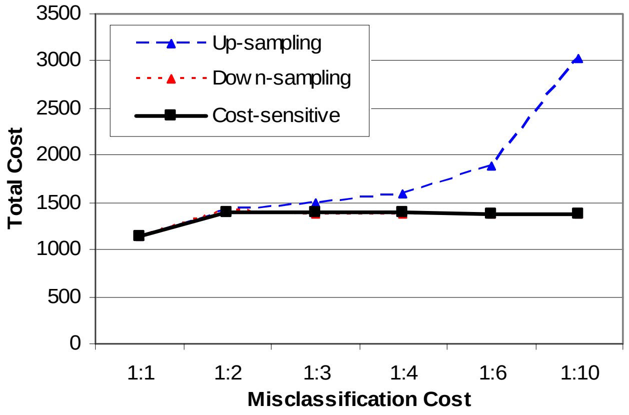

Figure A3: Results for Boa1

The connect-4 data set yields nearly identical performance for all three methods (like the blackjack data set in Figure 5), except for the 1:25 cost ratio.

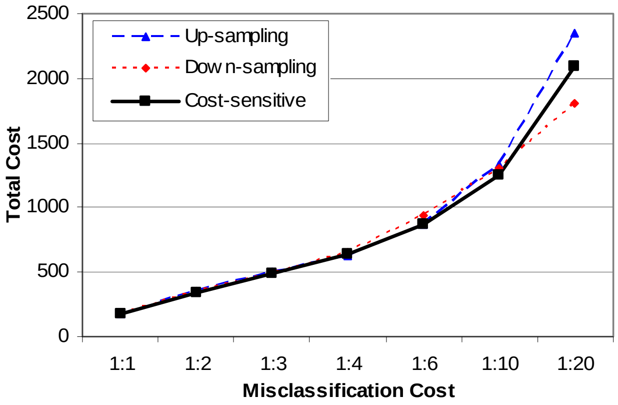

Figure A4: Results for Connect-4

The results for the hepatitis and bridges1 data sets in Figures A5 and A6 have the cost-sensitive method underperforming the two sampling methods for most cost ratios. The contraceptive data set in Figure 6 exhibited similar behavior.

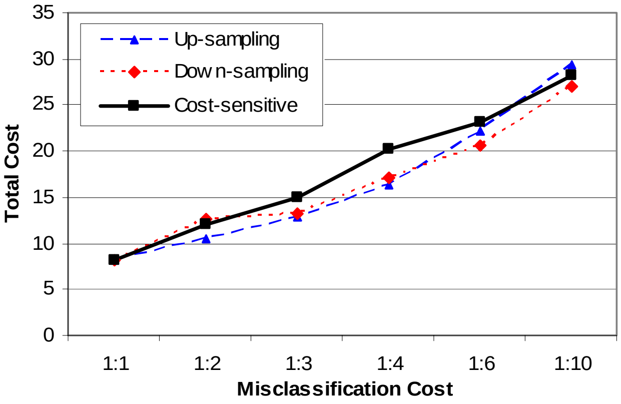

Figure A5: Results for Hepatitis

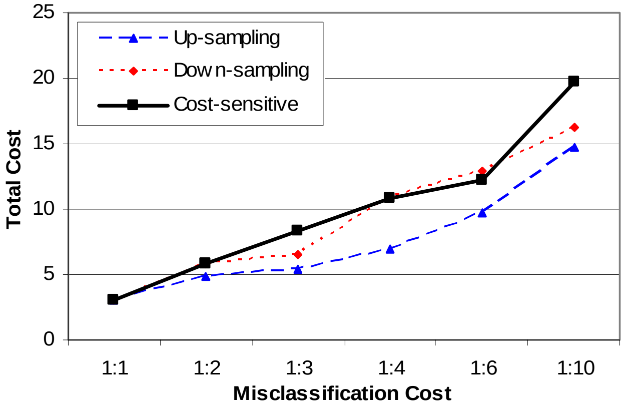

Figure A6: Results for Bridges1

The sonar data set is the only data set in which down-sampling substantially beat both cost-sensitive learning and up-sampling. This is unexpected since the sonar data set is quite small and one would expect down-sampling to perform worst in this situation (for other small data sets, down-sampling did in fact tend to perform poorly).

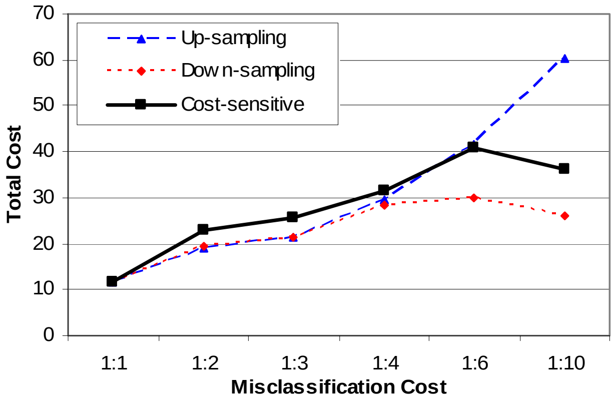

Figure A7: Results for Sonar

The promoters data set is the only data set for which up-sampling substantially beat both down-sampling and up-sampling.

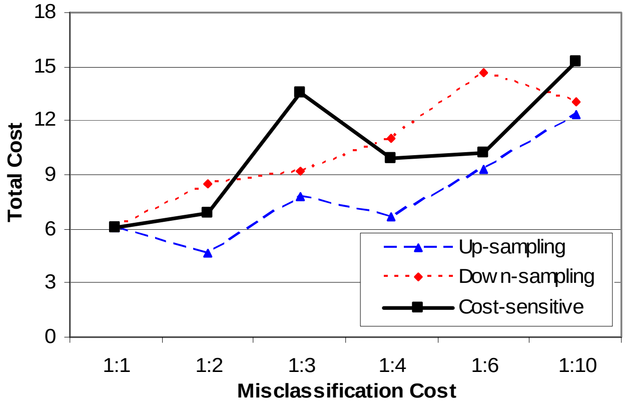

Figure A8: Results for Promoters

The results for the pendigits data set in Figure A9 vary for the different cost ratios, although the cost-sensitive learning method performs best overall.

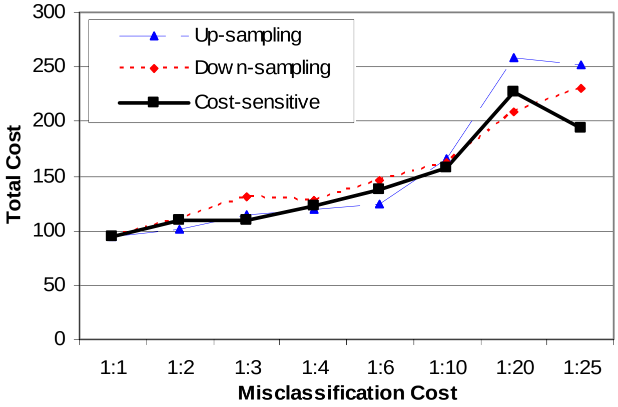

Figure A9: Results for Pendigits