C4.5 and Imbalanced Data sets: Investigating the effect of sampling method, probabilistic estimate, and decision tree structure

Nitesh V. Chawla NITESH.CHAWLA@CIBC.CA
Customer Behavior Analytics, Business Analytic Solutions, CIBC, BCE Place, 11th Floor, 161 Bay Street, Toronto, ON, CANADA M6S 5A6

Abstract

Imbalanced data sets are becoming ubiquitous, as many applications have very few instances of the "interesting" or "abnormal" class. Traditional machine learning algorithms can be biased towards majority class due to over-prevalence. It is desired that the interesting (minority) class prediction be improved, even if at the cost of additional majority class errors. In this paper, we study three issues, usually considered separately, concerning decision trees and imbalanced data sets — quality of probabilistic estimates, pruning, and effect of preprocessing the imbalanced data set by over or under-sampling methods such that a fairly balanced training set is provided to the decision trees. We consider each issue independently and in conjunction with each other, highlighting the scenarios where one method might be preferred over another for learning decision trees from imbalanced data sets.

1. Introduction

A data set is imbalanced if the classes are not approximately equally represented. There have been attempts to deal with imbalanced data sets in domains such as fraudulent telephone calls (Fawcett & Provost, 1996), telecommunications management (Ezawa et al., 1996), text classification (Lewis & Ringuette, 1994; Dumais et al., 1998; Mladenić & Grobelnik, 1999; Cohen, 1995) and detection of oil spills in satellite images (Kubat et al., 1998).

The compelling question, given the different class distributions, is: *What is the correct distribution for a learning algorithm?* Weiss and Provost (2003) present a detailed analysis of the effect of class distribution on classifier learning. Our observations agree with their work that the natural distribution is often not the best distribution for learning a classifier. Also, the imbalance in the data can be more characteristic of "sparseness" in feature space than the class imbalance. In that scenario, simple over-sampling and under-sampling might not suffice (Chawla et al., 2002). We extend our previous work of sampling strategies under the setting of different levels of decision tree pruning, and different probabilistic estimates at leaves. We consider over-sampling with replication, under-sampling, and synthetically creating minority class examples.

The representation or structure of a decision tree is also important to consider. Pruned or unpruned trees can have varied effects on learning from imbalanced data sets. Pruning can be detrimental to learning from imbalanced data sets as it can potentially collapse (small) leaves belonging to the minority class, thus reducing the coverage. Thus, it brings us to another question: *What is the right structure of the decision tree?* Do we need to use the completely grown tree or pruning is necessary? Can pruning be useful if applied with sampling strategies? Using C4.5 (Quinlan, 1992) as the classifier, we investigate three different levels of pruning: no pruning, default pruning, and pruning at a certainty level of 1.

A decision tree, is typically, evaluated by predictive accuracy that considers all errors equally. However, predictive accuracy might not be appropriate when the data is imbalanced and/or the costs of different errors vary markedly. As an example, consider the classification of pixels in mammogram images as possibly cancerous (Woods et al., 1993; Chawla et al., 2002). A typical mammography data set might contain 98% normal pixels and 2% abnormal pixels. A simple default strategy of guessing the majority class would give a predictive accuracy of 98%. Ideally, a fairly high

Workshop on Learning from Imbalanced Datasets II, ICML, Washington DC, 2003.

rate of correct cancerous predictions is required, while allowing for a small to moderate error rate in the majority class. It is more costly to predict a cancerous case as non-cancerous, than otherwise.

Moreover, distribution/cost sensitive applications can require a ranking or a probabilistic estimate of the instances. For instance, revisiting our mammography data example, a probabilistic estimate or ranking of cancerous cases can be decisive for the practitioner. The cost of further tests can be decreased by thresholding the patients at a particular rank. Secondly, probabilistic estimates can allow one to threshold ranking for class membership at values < 0.5. Hence, the classes assigned at the leaves of the decision trees have to be appropriately converted to probabilistic estimates (Provost & Domingos, 2003; Zadrozny & Elkan, 2001). This brings us to another question: *What is the right probabilistic estimate for imbalanced data sets?*

We attempt to answer the questions raised in the preceding discussion using C4.5 release 8 decision tree as our classifier. We used AUC as the performance metric (Swets, 1988; Bradley, 1997; Hand, 1997). We wanted to compare various methods based on the quality of their probabilistic estimates. This can allow us to rank cases based on their class memberships, and can give a general idea of the ranking of the 'positive" class cases. AUC can give a general idea of the quality of the probabilistic estimates produced by the model, without requiring one to threshold at a probability of 0.5 or less for classification accuracy (Hand, 1997). AUC can tell us whether a randomly chosen majority class example has a higher majority class membership than a randomly chosen minority class example.

The paper is structured as follows. In Section 2 we describe the probabilistic version of C4.5 trees, as used in this paper. Section 3 discusses the pruning levels used for the experiments. In Section 4 we describe the sampling strategies. Section 5 includes our experiments, and Section 6 presents the summary and future work.

2. Probabilistic C4.5

Typically, C4.5 assigns the frequency of the correct counts at the leaf as the probabilistic estimate. For notational purposes, TP is the number of true positives at the leaf, FP is the number of false positives, and C is the number of classes in the data set. Thus, the frequency based probabilistic estimate can be written as:

$$P_{leaf} = TP/(TP + FP) \tag{1}$$

However, simply using the frequency of the correct counts (of classes) at a leaf might not give sound probabilistic estimates (Provost & Domingos, 2003; Zadrozny & Elkan, 2001). A (small) leaf can potentially give optimistic estimates for classification purposes. For instance, the frequency based estimate will give the same weights to leaves with the following (TP, FP) distributions: $(5, 0)$ and $(50, 0)$. The relative coverage of the leaves and the original class distribution is not taken into consideration. Given the evidence, a probabilistic estimate of 1 for the $(5, 0)$ leaf is not very sound. Smoothing the frequency-based estimates can mitigate the aforementioned problem (Provost & Domingos, 2003). One way of smoothing those probabilities is using the Laplace estimate, which can be written as follows:

$$P_{Laplace} = (TP + 1)/(TP + FP + C) \tag{2}$$

Again considering the two pathological cases of $TP = 5$ and $TP = 50$, the Laplace estimates are 0.86 and 0.98, respectively, which are more reliable given the evidence.

However, Laplace estimates might not be very appropriate for highly imbalanced data sets (Zadrozny & Elkan, 2001). In that scenario, it could be useful to incorporate the prior of positive class to smooth the probabilities so that the estimates are shifted towards the minority class base rate (b). The m-estimate (Cussents, 1993) can be used as follows (Zadrozny & Elkan, 2001):

$$P_m = (TP + bm)/(TP + FP + m) \tag{3}$$

where b is the base rate or the prior of positive class, and m is the parameter for controlling the shift towards b. Zadrozny and Elkan (2001) suggest using m, given b, such that $bm = 10$.

3. Tree structure

Pruning is useful for decision trees as it improves generalization and accuracy of unseen test instances. However, pruning methods are generally based on a error function, and might not be conducive towards learning from imbalanced data sets. We wanted to empirically investigate the pruning methods over a range of imbalanced data sets, and consider their effect on the probabilistic estimates and the sampling methods. C4.5 uses error-based pruning. We considered three different levels of pruning of the C4.5 decision tree: unpruned, default pruned, and pruned at certainty factor of 1 (Quinlan, 1992). For unpruned trees, we modified

C4.5 code so that the tree growing process does not prune and does not "collapse", as proposed by Provost and Domingos (2003). To evaluate the effect of pruning on the imbalanced data sets, we pruned the trees at the certainty factor of 25% (default pruning), and at the certainty factor of 1%.

Unpruned and uncollapsed trees can potentially lead to a problem of small disjuncts (Weiss, 1995) as the trees are grown to their full complete size on the imbalanced training sets. Overfitting can occur, and pruning can be used to improve generalization of the decision trees.

4. Sampling strategies

A popular way to deal with imbalanced data sets is to either over-sample the minority class or under-sample the majority class. We present two versions of over-sampling, one by replicating each minority class example and the other by creating new synthetic examples (SMOTE) (Chawla et al., 2002), and underampling.

4.1. Over-sampling

Over-sampling with replication does not always improve minority class prediction. We interpret the underlying effect in terms of decision regions in feature space. Essentially, as the minority class is over-sampled by increasing amounts, the effect is to identify similar but more specific regions in the feature space as the decision region for the minority class.

If we replicate the minority class, the decision region for the minority class becomes very specific and will cause new splits in the decision tree. This will lead to overfitting. Replication of the minority class does not cause its decision boundary to spread into the majority class region.

4.2. SMOTE: Synthetic Minority over-sampling TEchnique

We generate synthetic examples by operating in the "feature space" rather than the "data space" (Chawla et al., 2002). The synthetic examples cause the classifier to create larger and less specific decision regions, rather than smaller and more specific regions. The minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. Synthetic samples are generated in the following way: Take the difference between the feature vector (sample) under consideration and its nearest neighbor. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the minority class to become more general.

The nominal values are treated differently. We use Cost and Salzberg (1993) modification of Value Distance Metric (Stanfill & Waltz, 1986) to compute the nearest neighbors for the nominal valued features. VDM looks at the overlap of feature values over all feature vectors. A matrix defining the distance between corresponding feature values for all feature vectors is created. The distance δ between two corresponding feature values is defined as follows.

$$\delta(V_1, V_2) = \sum_{i=1}^{n} |\frac{C_{1i}}{C_1} - \frac{C_{2i}}{C_2}|^k \tag{4}$$

In the above equation, V_1 and V_2 are the two corresponding feature values. C_1 is the total number of occurrences of feature value V_1, and C_{1i} is the number of occurrences of feature value V_1 for class i. A similar convention is also applied to C_{2i} and C_2. k is a constant, usually set to 1. The distance Δ between two feature vectors is given by:

$$\Delta(X, Y) = w_x w_y \sum_{i=1}^{N} \delta(x_i, y_i)^r \tag{5}$$

$r = 1$ yields the Manhattan distance, and $r = 2$ yields the Euclidean distance (Cost & Salzberg, 1993). w_x and w_y are the exemplar weights in the modified VDM. Since SMOTE is not used for classification purposes, we set the weights to 1 equation 5. We create new set of feature values (for the synthetic minority class example) by taking the majority vote of the feature vector in consideration and its k nearest neighbors. In the absence of a majority, we select the feature value at random.

4.3. Under-sampling

We under-sample the majority class by randomly removing samples from the majority class population until the minority class becomes some specified percentage of the majority class (Chawla et al., 2002). This forces the learner to experience varying degrees of under-sampling and at higher degrees of under-sampling the minority class has a larger presence in the training set.

Table 1. Data set details.

Data set	Size	Features	Distribution
Pima	768	8	0.65; 0.35
Phoneme	5484	5	0.71; 0.29
Satimage	6435	36	0.9; 0.1
Mammography	11183	6	0.98; 0.02
Krkopt	28056	6	0.99; 0.01

5. Data sets

We used five data sets with very different class distributions. Four of our data sets come from the UCI repository (Blake & Merz, 1998). For the krkopt data set we sampled two classes to make it a 2-class and highly skewed data set. Similarly, we converted satimage into a 2-class data set by converting all but one small class into a single class (Chawla et al., 2002). Table 1 summarizes our data sets. The mammography data set is available from the Intelligent Systems Lab, University of South Florida. We divided the data sets into 2/3rd training and 1/3rd testing stratified sets for our experiments.

6. Results

We used the different variants of C4.5 in conjunction with the three different probabilistic estimates and sampling methods. The goal was to empirically investigate the effect of the structure, probabilistic estimate, and sampling method on AUC. Not knowing the "right" class distribution, we simply over-sampled (both SMOTE and replication) or under-sampled such that the class ratio is one, in addition to using the original class distribution. We believe there might be other appropriate class distributions that could give us better results in terms of AUC, and that is a part of our future work. One can potentially find out the right distribution by exploring the possible ratios between the minority class and majority class. One can also approximate the different distributions by thresholding at different probabilities coming from leaf mixtures. By doing so one can increase or decrease the effect of mixture distribution at a leaf. For all our experiments, the distribution of examples in the testing set was the same as originally occuring in the data set.

We compare various approaches using box-plots to show AUC improvements (or deterioration) provided by one method over another as shown in the Figures 1 to 9. Each method (or box) in the box-plots represents all the data sets. The whiskers at the end of the box plots show the minimum and maximum values (outliers), while the bar shows the median. If the median bar is above 0, than the approach, represented by the box plot, is doing better on average than the approach compared to. And if the complete box, including the whiskers, is above 0 then that approach is consistently better than the other approach. The convention in the figures is as follows: *original* implies that the decision tree is learned from the original distribution; *smote* implies that the decision tree is learned from the balanced distribution constructed by SMOTE; *over* implies that the decision tree is learned from the balanced distribution constructed from over-sampling with replication; and *under* implies that the decision tree from the balanced distribution constructed from under-sampling. Each of *original*, *smote*, *over*, and *under* is suffixed with following: *laplace* or m to signify the probabilistic estimate used; U, P, or PC (U is unpruned, P is default pruning, and PC is pruning at certainty factor of 1) to show the pruning method used.

Figures 1 to 3 summarize the effect of the probabilistic estimate on learning C4.5 decision trees from imbalanced data sets. The box-plots represent improvement in AUC obtained by $P_{laplace}$ over P_{leaf} and P_m over P_{leaf}. The X-axis represents each of the methods corresponding to the box plots. Figure 1 is for unpruned decision trees, Figure 2 is for pruned decision trees, and Figure 3 is for decision trees pruned with certainty factor of 1. Figures show that both P_m and $P_{laplace}$ estimates provide a consistent advantage over P_{leaf} for the original distribution. This is what one would have expected as the fully grown tree can have small leaves, giving optimistic P_{leaf} estimate, as shown in the example considered earlier. The gain provided by P_m and $P_{laplace}$ is diminished at higher levels of pruning, as pruning effectively eliminates the smaller minority class leaves, reducing the coverage. Thus, pruning can have a detrimental effect on learning from imbalanced data sets. Sampling generally helped in learning, and was not very sensitive to the amount of pruning, as the trees were learned from balanced training sets. Also, P_m and $P_{laplace}$ give better AUC's than P_{leaf} for the sampling methods. Thus, even if the model is learned from a balanced training set (and tested on the skewed testing set), smoothing produces more sound estimates than just the frequency based method.

Figures 4 to 6, for P_{leaf}, $P_{laplace}$ and P_m respectively, compare the effect of pruning on learning from the original and sampled data sets. As we noted from the previous set of Figures, pruning is detrimental to learning from imbalanced data sets. We note pruned trees usually give worse AUC's than unpruned trees. Among the sampling strategies, over-sampling is particularly helped by pruning. This is not surprising

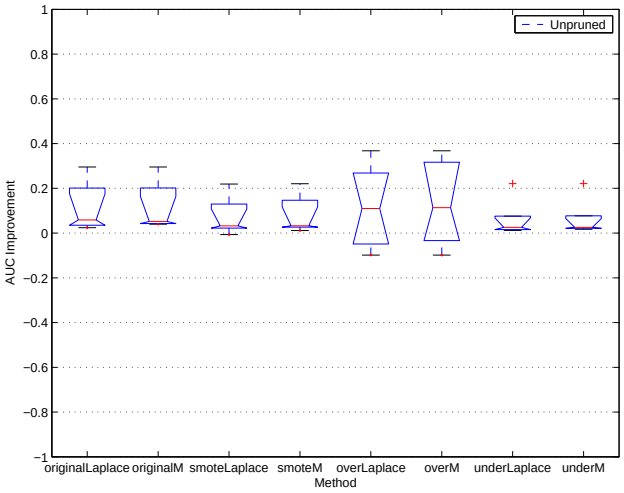

Figure 1. Improvement or deterioration in AUC by $P_{laplace}$ and P_m over P_{leaf} using unpruned trees for original and sampled data sets.

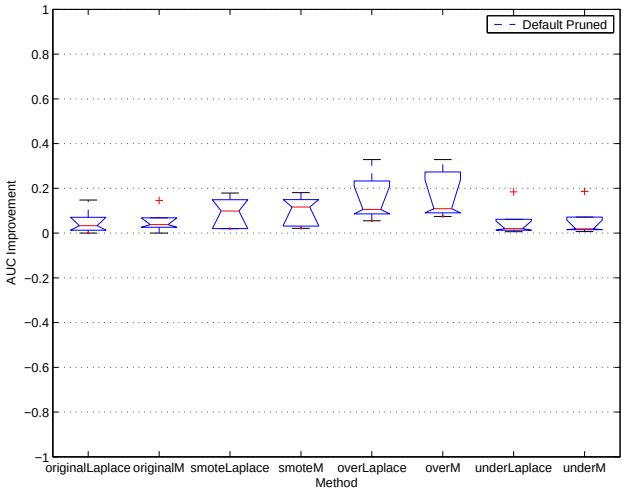

Figure 2. Improvement or deterioration in AUC by $P_{laplace}$ and P_m over P_{leaf} using default pruned trees for original and sampled data sets.

as over-sampling usually leads to small, very specific decision regions (Chawla et al., 2002), and pruning improves their generalization. Thus, pruning is helpful with imbalanced data sets, if one is deploying some sampling routine to balance the class distribution. Otherwise, pruning can reduce the minority class coverage in the decision trees.

Figures 7 to 9 compare the different sampling strategies. Each Figure shows the improvement achieved by over-sampling and SMOTE over under-sampling for P_{leaf}, $P_{laplace}$ and P_m, respectively. We observe that SMOTE on an average is better than under-sampling. We also observe that over-sampling on an average is worse than under-sampling. Based on that evidence, we can also infer that SMOTE on an average is better than over-sampling.

7. Summary and Future Work

In this paper, we presented an empirical analyses of various components of learning C4.5 decision trees from imbalanced data sets. We juxtaposed three issues of learning decision trees from imbalanced data sets, usually considered separately, as part of one study.

Our main conclusions can be summarized as follows:

1. P_{leaf} gives worse probabilistic estimates than $P_{laplace}$ and P_m. The gain provided by P_m and $P_{laplace}$ is diminished at higher levels of pruning. $P_{laplace}$ and P_m are comparable to each other.

2. Pruning is usually detrimental to learning from imbalanced data sets. However, if a sampling routine is used, pruning can help as it improves the generalization of the decision tree classifier. Given that the testing set can come from a different distribution, not having specific trees can help.

3. SMOTE on an average improves the AUC's over the other sampling schemes. We believe this is due to SMOTE working in the "feature space" and constructing new examples. SMOTE helps in broadening the decision region for a learner, thus improving generalization. We also observe that under-sampling is usually better than over-sampling with replication.

As a part of future work we propose another sampling strategy for comparing with SMOTE: under-sampling using neighborhood information. That is, instead of under-sampling at random, only under-sample if the minority class is in the k nearest neighbors. This approach can potentially have scalability issues due to a much higher prevalence of majority class, but this will help us in establishing another benchmark for sampling in "feature space". We would also like to investigate ways to construct appropriate class distributions for a particular domain, and evaluate the different settings considered in this paper by varying testing set distribution.

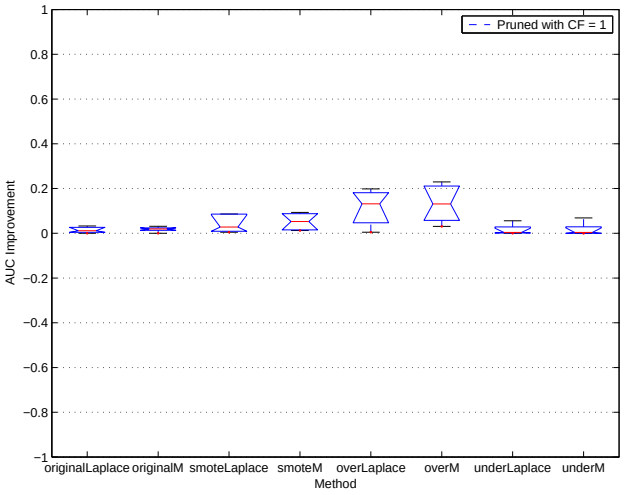

Figure 3. Improvement or deterioration in AUC by $P_{laplace}$ and P_m over P_{leaf} using trees pruned at cf = 1 for original and sampled data sets.

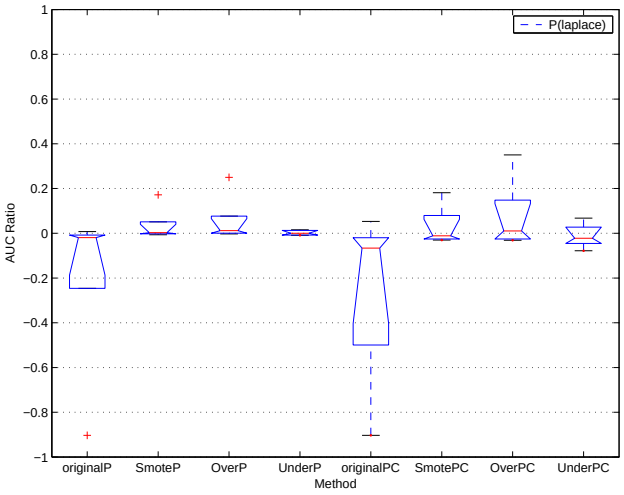

Figure 5. Improvement or deterioration in AUC by pruning using $P_{laplace}$ for decision trees learned from the original and sampled data sets.

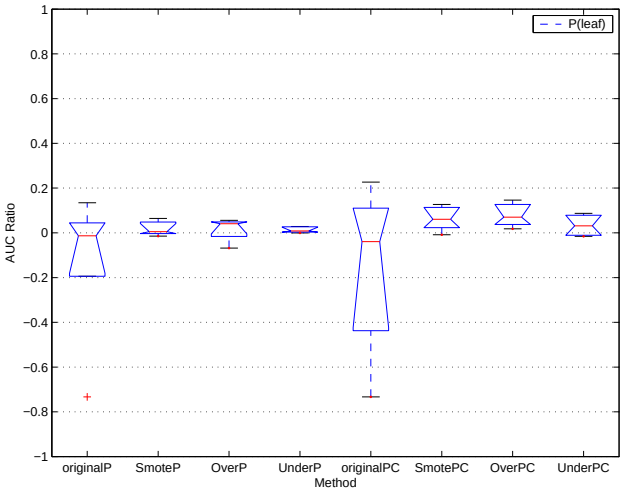

Figure 4. Improvement or deterioration in AUC by pruning using P_{leaf} for decision trees learned from the original and sampled data sets.

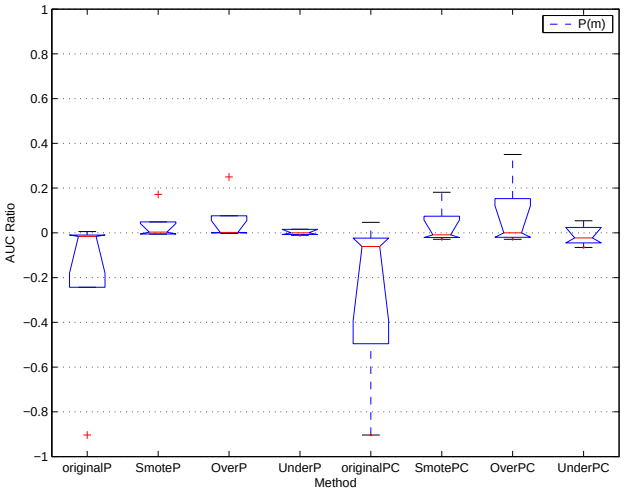

Figure 6. Improvement or deterioration in AUC by pruning using P_m for decision trees learned from the original and sampled data sets.

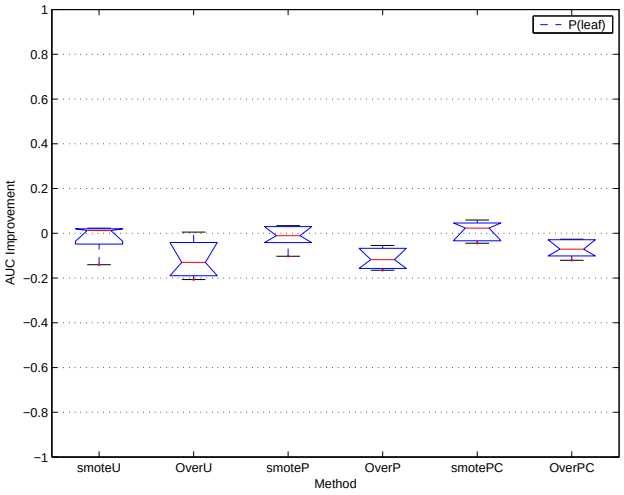

Figure 7. Improvement or deterioration in AUC by over-sampling and SMOTE over under-sampling using P_{leaf} at different levels of pruning.

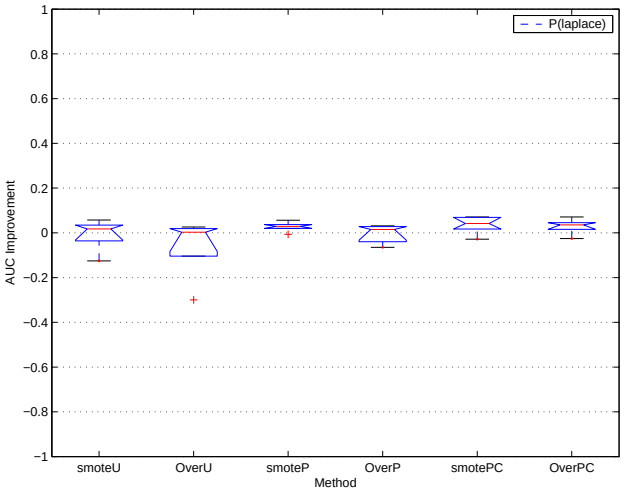

Figure 8. Improvement or deterioration in AUC by over-sampling and SMOTE over under-sampling using $P_{laplace}$ at different levels of pruning.

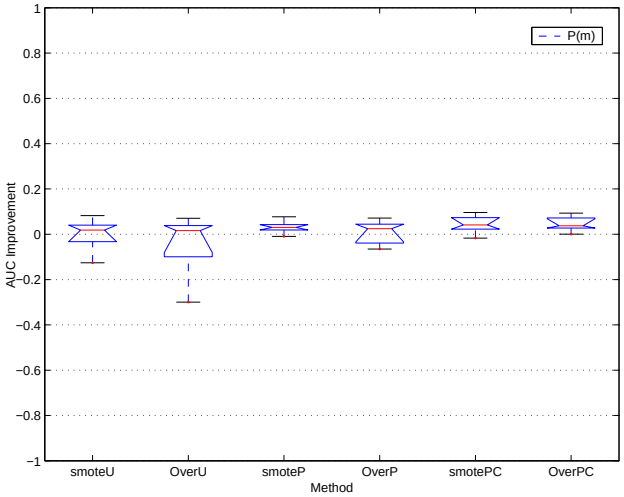

Figure 9. Improvement or deterioration in AUC by over-sampling and SMOTE over under-sampling using P_m at different levels of pruning.

Acknowledgements

We would like to thank the reviewers for their useful comments on the paper.

References

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.

Bradley, A. P. (1997). The Use of the Area Under the ROC Curve in the Evaluation of Machine Learning Algorithms. *Pattern Recognition*, *30(6)*, 1145–1159.

Chawla, N., Hall, L., K.W., B., & Kegelmeyer, W. (2002). SMOTE: Synthetic Minority Oversampling TEchnique. *Journal of Artificial Intelligence Research*, *16*, 321–357.

Cohen, W. (1995). Learning to Classify English Text with ILP Methods. *Proceedings of the 5th International Workshop on Inductive Logic Programming* (pp. 3–24). Department of Computer Science, Katholieke Universiteit Leuven.

Cost, S., & Salzberg, S. (1993). A Weighted Nearest Neighbor Algorithm for Learning with Symbolic Features. *Machine Learning*, *10*, 57–78.

Cussents, J. (1993). Bayes and pseudo-bayes estimates of conditional probabilities and their reliabil-

ities. *Proceedings of European Conference on Machine Learning.*

Dumais, S., Platt, J., Heckerman, D., & Sahami, M. (1998). Inductive Learning Algorithms and Representations for Text Categorization. *Proceedings of the Seventh International Conference on Information and Knowledge Management.* (pp. 148–155).

Ezawa, K., J., Singh, M., & Norton, S., W. (1996). Learning Goal Oriented Bayesian Networks for Telecommunications Risk Management. *Proceedings of the International Conference on Machine Learning, ICML-96* (pp. 139–147). Bari, Italy: Morgan Kauffman.

Fawcett, T., & Provost, F. (1996). Combining Data Mining and Machine Learning for Effective User Profile. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining* (pp. 8–13). Portland, OR: AAAI.

Hand, D. (1997). *Construction and assessment of classification rules.* Chichester: John Wiley and Sons.

Japkowicz, N., & Stephen, S. (2002). The class imbalance problem: A systematic study. *Intelligent Data Analysis*, *6*.

Kubat, M., Holte, R., & Matwin, S. (1998). Machine Learning for the Detection of Oil Spills in Satellite Radar Images. *Machine Learning*, *30*, 195–215.

Lewis, D., & Ringuette, M. (1994). A Comparison of Two Learning Algorithms for Text Categorization. *Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval* (pp. 81–93).

Mladenić, D., & Grobelnik, M. (1999). Feature Selection for Unbalanced Class Distribution and Naive Bayes. *Proceedings of the 16th International Conference on Machine Learning.* (pp. 258–267). Morgan Kaufmann.

Provost, F., & Domingos, P. (2003). Tree induction for probability-based rankings. *Machine Learning*, *52(3)*.

Quinlan, J. (1992). *C4.5: Programs for Machine Learning.* San Mateo, CA: Morgan Kaufmann.

Stanfill, C., & Waltz, D. (1986). Toward Memory-based Reasoning. *Communications of the ACM*, *29*, 1213–1228.

Swets, J. (1988). Measuring the Accuracy of Diagnostic Systems. *Science*, *240*, 1285–1293.

Weiss, G. (1995). Learning with rare cases and small disjuncts. *Proceedings of the Twelfth International Conference on Machine Learning.*

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

Zadrozny, B., & Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. *Proceedings of the Seventh International Conference on Knowledge Discovery and Data Mining.*