A Wrapper for Reweighting Training Instances for Handling Imbalanced Data Sets

M. Karagiannopoulos, D. Anyfantis, S. Kotsiantis and P. Pintelas
Educational Software Development Laboratory
Department of Mathematics, University of Patras, Greece
{ mariosk,dany,sotos,pintelas}@math.upatras.gr,
WWW home page: http://www.math.upatras.gr/~esdlab

Abstract. A classifier induced from an imbalanced data set has a low error rate for the majority class and an undesirable error rate for the minority class. This paper firstly provides a systematic study on the various methodologies that have tried to handle this problem. Finally, it presents an experimental study of these methodologies with a proposed wrapper for reweighting training instances and it concludes that such a framework can be a more valuable solution to the problem.

1 Introduction

Classifiers are often faced with imbalanced data sets for various reasons; the latest can cause the classifier to be biased towards one class. This bias is the outcome of one class being seriously under represented in the training data in favor of other classes. It can be qualified to the way in which classifiers are designed. Inductive classifiers are normally designed to minimize errors over the training examples. Learning algorithms on the other hand ignore classes containing few examples [11]. For a number of application domains, a massive disproportion in the number of cases belonging to each class is common. For example, in detection of fraud in telephone calls and credit card transactions. Moreover, in direct marketing, it is frequent to have a small response rate (about 1%) for most marketing campaigns.

The machine learning community has mostly addressed the issue of class imbalance in two ways. One is to give distinct costs to training instances [6] while the other is to re-sample the original dataset, either by oversampling the minority class and/or under-sampling the majority class [12], [9]. Although many methods for coping with imbalanced data sets have been proposed, there are still several open questions. One open question is whether simply changing the distribution skew can improve predictive performance steadily. To handle the problem, we developed a

wrapper for reweighting training instances. The effectiveness of our approach is evaluated over eight imbalanced datasets using the C4.5 [15], Naive Bayes [5] and 5NN [1] as classifiers and the geometric mean of accuracies as performance measure [12].

In the following section we review the attempts for handling imbalanced data sets, while section 3 presents the details of our approach. Section 4 presents experimental results comparing our approach to other approaches. Finally, section 5 discusses the results and suggests directions for future work.

2 Review of existing techniques for handling imbalanced data sets

A simple method that can be used to imbalanced data sets is to reweigh training examples according to the total cost assigned to each class [4]. The idea is to change the class distributions in the training set towards the most costly class. In [8] the effect of imbalance in a dataset is discussed. Two main strategies are evaluated: Under-sampling and Resampling. Both the two sampling approaches were helpful, and is observed that sophisticated sampling techniques does not give any clear advantage in the domain considered.

Another approach is presented in [13]. They combined over-sampling of the minority class with under-sampling of the majority class. However, the over-sampling and under-sampling combination did not provide significant improvement. In [3] an over-sampling approach is presented according to which the minority class is over-sampled by creating "synthetic" instances rather than by over-sampling with replacement with better results.

Changing the class distribution is not the only technique to improve classifier performance when learning from imbalanced data sets. A different approach to incorporating costs in decision-making is to define fixed and unequal misclassification costs between classes [8].

An alternative to balancing the classes is to develop a learning algorithm that is intrinsically insensitive to class distribution in the training set. An example of this kind of algorithm is the SHRINK algorithm [12] that finds only rules that best summarize the positive instances (of the small class), but makes use of the information from the negative instances. MetaCost [6] is another method for making a classifier cost-sensitive. The procedure begins to learn an internal cost-sensitive model by applying a cost-sensitive procedure, which employs a base learning algorithm. Then, MetaCost procedure estimates class probabilities using bagging and then re-labels the training instances with their minimum expected cost classes, and finally relearns a model using the modified training set.

In [16] different weights for false positives and false negatives are used to apply AdaBoost than bagging in text-filtering. AdaBoost uses a base classifier to induce multiple individual classifiers in sequential trials, and a weight is assigned to each training instance. At the end of each trial, the vector of weights is adjusted to reflect the importance of each training instance for the next induction trial. This adjustment effectively increases the weights of misclassified examples and decreases the weights of the correctly classified examples. A similar technique is proposed in [7].

3 Proposed Technique

The problem of determining which proportion of positive/negative examples is the best for learning is an open problem of learning from imbalanced data sets. The proposed technique is based on the previous referred reweighting technique; however, we do not apply a single cost matrix for reweighting training instances. We did not only examine the relationship between false negative and false positive costs to be the inverse of the assumed prior to compensate for the imbalanced priors. We examine all the cost matrixes:

$$\begin{bmatrix} 0 & 1 \\ x & 0 \end{bmatrix}$$

Where x takes the values from [(Number of Instances of the Majority Class)/ (Number of Instances of the Majority Class)-1] to [(Number of Instances of the Majority Class)/ (Number of Instances of the Majority Class)+1], with step 0.1. The cost-matrix with the best performance using 10-fold cross validation is then applied for the classification of new instances. The proposed technique (WRTI) is presented in Fig. 1. A key feature of our method is that it does not require any modification of the underlying learning algorithm.

In the following section, we empirically evaluate the performance of our approach with the other well known techniques using a decision tree, an instance base learner and a Bayesian model as base classifiers.

```
   RatioA =ClassWithMoreInstances)/ClassWithLessInstances);
   for (s = RatioA - 1.0; s < RatioA + 1.0; s = s + 0.1)
           {
               CostMatrix cm;
   for (int i=0; i<2; i++)
           {
               for (int j=0; j<2; j++)
               {
                   if (i == j)
                       cm.setCell(i, j, 0);
                   if (i == 0 && j == 1)
                       cm.setCell(i, j, 1);
                   if (i == 1 && j == 0)
                       cm.setCell(i, j, s);
               }
           }
             CostSensitiveClassifier csc = new
CostSensitiveClassifier();
             csc.setCostMatrix(cm);
             csc.setClassifier(UsedClassifier);
             eval.crossValidateModel(csc, data, 10);
             result =
eval.truePositiveRate(ClassIndexWithLessInstances) *
```

```
eval.truePositiveRate(ClassIndexWithMoreInstances);
                    if (result > BestResult)
                    {   BestResult = result;
                        BestCM = cm;
                    }
   }
```

Fig. 1. A wrapper for reweighting training instances

4 Experiments

In Table 1, there is a brief description of the data sets that we used for our experiments. Except for the "eap" data set, all were drawn from the UC Irvine Repository [2]. Eap data is from Hellenic Open University and was used in order to determine whether a student is about to drop-out or not [10].

Table 1. Description of the data sets

Data sets	Instances	Categorical features	Numerical features	Instances of minority class	Classes
breast-cancer	286	9	0	85	2
credit-g	1000	13	7	300	2
Diabetes	768	0	8	268	2
Haberman	306	0	3	81	2
Hepatitis	155	13	6	32	2
Ionosphere	351	34	0	126	2
Eap	344	11	0	122	2
Sick	3772	22	7	231	2

A classifier's performance of two class problems can be separately calculated for its performance over the positive instances (denoted as α^+) and over the negative instances (denoted as α^-). The true positive rate (α^+) or sensitivity is the fraction of positive instances predicted correctly by the model. Similarly, the true negative rate (α^-) or specificity is the fraction of negative instances predicted correctly by the classifier. In [12] the authors propose the geometric mean of the accuracies: $g = \sqrt{a^+ \times a^-}$ for imbalanced data sets. Moreover, ROC curves (Receiving Operator Characteristic) provide a visual representation of the trade off between true positives (α^+) and false positives (α^-). These are plots of the percentage of correctly classified positive instances α^+ with respect to the percentage of incorrectly classified negative instances α^- [14]. The method for plotting a ROC curve is closely related to a method for making algorithms cost-sensitive, that we call Threshold method [17]. This method uses a threshold so as to maximize the given performance measure in the curve. Classification ability of the learning methods in our experiments was measured with geometric mean of the accuracies. For the examined cost models, the relationship between false negative and false positive costs was chosen to be the inverse of the assumed prior to compensate for the imbalanced priors. In the following Tables, win (v) indicates that the proposed method along with the learning algorithm performed statistically better than the other classifier according to t-test

with $p<0.05$. Loss (*) indicates that the proposed method along with the learning algorithm performed statistically worse than the other classifier according to t-test with $p<0.05$. In all the other cases, there is no significant statistical difference between the results.

In Table 2, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data set using Naive Bayes (NB) as base classifier. Five well-known algorithms were used for the comparison: Threshold method [17], Reweighting and Cost Sensitive method [4], Adaboost cost sensitive method [16], and Metacost algorithm [6]. We also present the accuracy of the simple Bayes algorithm as borderline. It must be mentioned that we used the free available source code for these methods [17] for our experiments. In the Table 2 except for geometric mean we also present the true-positive rate, and true-negative rate. The positive class for our experiments is the majority class. In the last row of the Table 2, the average value of the geometric means is also calculated in all data sets.

Table 2. Accuracy on majority class (α+), accuracy on minority class (α-) and geometric mean (g) with NB as base classifier

Data sets		WRTINB	ReWNB	ThresNB	CostNB	AdabcosNB	MetacostNB	NB
breast-cancer	g	0.67	0.66	0.63*	0.66	0.63*	0.65	0.6*
	α+	0.66	0.74 v	0.62 *	0.74 v	0.72 v	0.79 v	0.85v
	α-	0.68	0.58 *	0.65 *	0.58 *	0.56 *	0.54 *	0.43*
credit-g	g	0.73	0.72	0.71	0.72	0.71	0.66 *	0.65*
	α+	0.71	0.75 v	0.69	0.75 v	0.75 v	0.77 v	0.86v
	α-	0.75	0.69 *	0.74	0.69 *	0.67 *	0.57 *	0.49*
diabetes	g	0.74	0.73	0.72	0.73	0.73	0.70 *	0.71*
	α+	0.75	0.78 v	0.65 *	0.78 v	0.77	0.75	0.84v
	α-	0.73	0.68 *	0.8 v	0.68 *	0.69 *	0.66 *	0.6 *
haberman	g	0.6	0.56 *	0.59	0.56*	0.56*	0.57*	0.44*
	α+	0.88	0.89	0.64 *	0.89	0.88	0.87	0.94v
	α-	0.41	0.35 *	0.55 v	0.35 *	0.36 *	0.38 *	0.21*
hepatitis	g	0.8	0.8	0.76 *	0.8	0.78	0.81	0.78
	α+	0.86	0.83 *	0.87	0.83 *	0.86	0.79 *	0.87*
	α-	0.75	0.78 v	0.67 *	0.78 v	0.71 *	0.84 v	0.7*
ionosphere	g	0.84	0.82	0.88 v	0.82	0.91 v	0.77*	0.83
	α+	0.87	0.78 *	0.93 v	0.78 *	0.93 v	0.68 *	0.8*
	α-	0.81	0.87	0.81	0.87 v	0.9 v	0.88 v	0.86v
eap	g	0.85	0.85	0.83	0.85	0.83	0.85	0.84
	α+	0.87	0.87	0.86	0.87	0.85	0.88	0.9 v
	α-	0.83	0.83	0.81	0.83	0.82	0.83	0.78*
sick	g	0.86	0.86	0.76*	0.86	0.87	0.8*	0.86
	α+	0.82	0.82	0.98 v	0.82	0.88 v	0.73 *	0.94v
	α-	0.9	0.9	0.59 *	0.9	0.86 *	0.87 *	0.78*
Average	g	0.76	0.75	0.74	0.75	0.75	0.73	0.71

In general, all the tested techniques give better results than the single Naive Bayes. The most remarkable improvement is from our technique, even though the

Threshold method gives, on average, the best accuracy in the minority class. The Metacost cannot improve the results of the NB as his author suspects. It must be noted that for NB classifier, modifying the decision boundary (Cost Sensitive method) is equivalent to reweighting training instances so as the relationship between false negative and false positive costs to be the inverse of the imbalanced priors. Moreover, Adaboost cost sensitive method cannot give better results than Cost Sensitive, even though it is a more time consuming technique.

In Table 3, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data sets using C4.5 as base classifier.

Table 3. Accuracy on majority class (α+), accuracy on minority class (α-) and geometric mean (g) with NB as base classifier

Data sets		WRTIC4.5	ReWC4.5	ThresC4.5	CostC4.5	Adabcos C4.5	Metacost C4.5	C4.5
breast-cancer	g	0.59	0.57	0.45*	0.5 *	0.56 *	0.55 *	0.5 *
	α+	0.66	0.72 v	0.8 v	0.85 v	0.77 v	0.84 v	0.95 v
	α-	0.52	0.45 *	0.25 *	0.3 *	0.41 *	0.36 *	0.26 *
credit-g	g	0.67	0.66	0.64*	0.61*	0.62*	0.64*	0.58 *
	α+	0.75	0.67 *	0.7*	0.82 v	0.81 v	0.76	0.85 v
	α-	0.6	0.65 v	0.58	0.46 *	0.47 *	0.54	0.4 *
diabetes	g	0.73	0.72	0.7*	0.72	0.67*	0.73	0.7*
	α+	0.71	0.72	0.69	0.78 v	0.79 v	0.78 v	0.82 v
	α-	0.75	0.73	0.71 *	0.67 *	0.57 *	0.67 *	0.6 *
haberman	g	0.65	0.63	0.56 *	0.58 *	0.57 *	0.62 *	0.52 *
	α+	0.65	0.68 v	0.61 *	0.66	0.76 v	0.76 v	0.85 v
	α-	0.65	0.58 *	0.51 *	0.51 *	0.43 *	0.52 *	0.32 *
hepatitis	g	0.72	0.73	0.62 *	0.64 *	0.7	0.68 *	0.58 *
	α+	0.83	0.62 *	0.78 *	0.86 v	0.9 v	0.83	0.9 v
	α-	0.63	0.85 v	0.49 *	0.48 *	0.55 *	0.56 *	0.37 *
ionosphere	g	0.91	0.89	0.88 *	0.88 *	0.9	0.9	0.88 *
	α+	0.96	0.94	0.95	0.94	0.94	0.98	0.94
	α-	0.87	0.85	0.81*	0.82 *	0.86	0.82 *	0.82 *
eap	g	0.84	0.81 *	0.69 *	0.83	0.79 *	0.82	0.83
	α+	0.95	0.86 *	0.91 *	0.94	0.85 *	0.89 *	0.94
	α-	0.74	0.77 v	0.53 *	0.74	0.74	0.76	0.74
sick	g	0.97	0.97	0.92 *	0.96	0.95	0.96	0.93 *
	α+	0.99	0.99	0.99	0.99	1 v	0.98	0.99
	α-	0.95	0.95	0.85 *	0.92 *	0.9 *	0.95	0.87 *
Average	g	0.76	0.75	0.68	0.72	0.72	0.74	0.69

The same five well-known techniques for handling imbalanced data sets were also used for this comparison. Likewise with the previous experiment, our method has better performance than the other techniques. However, Metacost has really better performance with C4.5 than NB. It must also be mentioned that Threshold method gives worst performance than single C4.5. Adaboost cost sensitive method, as in the previous experiment, cannot give better results than reweighting method even though it uses more time for training.

In Table 4, one can see the comparisons of the proposed technique with other attempts that have tried to obtain the best performance of a given imbalance data sets using 5NN as base classifier. The same five well-known techniques for handling imbalanced data sets were also used for this comparison. Likewise with the previous experiment, our method has better performance than the other techniques. It must be mentioned that Adaboost cost sensitive method and Metacost algorithm are extremely time consuming techniques if they are combined with lazy algorithm 5NN without offering spectacular improvement in the performance. Threshold method gives, on average, the least improvement in the performance of 5NN.

Table 4. Accuracy on majority class (α+), accuracy on minority class (α-) and geometric mean (g) with 5NN as base classifier

Data sets		WRTI5NN	ReW5NN	Thres5NN	Cost5NN	Adabcos 5NN	Metacost 5NN	5NN
breast-cancer	g	0.64	0.62	0.6 *	0.61*	0.61*	0.51 *	0.45*
	α+	0.73	0.73	0.57 *	0.72	0.7 *	0.86 v	0.96v
	α-	0.56	0.52 *	0.63 v	0.52 *	0.53*	0.3 *	0.21*
credit-g	g	0.66	0.66	0.59 *	0.66	0.63 *	0.63 *	0.57 *
	α+	0.68	0.69	0.84 v	0.69	0.7	0.73 v	0.89 v
	α-	0.64	0.63	0.42 *	0.63	0.56 *	0.55 *	0.37 *
diabetes	g	0.70	0.71	0.69	0.71	0.66 *	0.71	0.68
	α+	0.76	0.69 *	0.79 v	0.69 *	0.71 *	0.75	0.83v
	α-	0.71	0.74 v	0.61 *	0.74 v	0.62 *	0.68 *	0.56*
haberman	g	0.57	0.57	0.58	0.57	0.53	0.59	0.39*
	α+	0.61	0.68 v	0.52 *	0.68 v	0.68 v	0.66 v	0.9v
	α-	0.48	0.47	0.65 v	0.47	0.41 *	0.52 v	0.17*
hepatitis	g	0.74	0.69 *	0.68 *	0.73	0.58 *	0.8 v	0.66 *
	α+	0.77	0.79	0.91 v	0.85 v	0.8 v	0.84 v	0.94 v
	α-	0.7	0.6 *	0.51 *	0.62 *	0.42 *	0.76 v	0.46 *
ionosphere	g	0.83	0.83	0.82	0.83	0.83	0.79 *	0.78*
	α+	0.97	0.97	0.97	0.97	0.95	0.98	0.98
	α-	0.71	0.71	0.7	0.71	0.72	0.63 *	0.62 *
eap	g	0.81	0.8	0.79	0.8	0.78 *	0.77 *	0.78 *
	α+	0.81	0.84 v	0.83	0.84 v	0.79	0.87 v	0.9 v
	α-	0.82	0.76 *	0.75 *	0.76 *	0.77 *	0.69 *	0.68 *
sick	g	0.89	0.84 *	0.62 *	0.84 *	0.87	0.79 *	0.61 *
	α+	0.93	0.89 *	0.99 v	0.89 *	0.98 v	0.9 *	0.99 v
	α-	0.85	0.79 *	0.39 *	0.79 *	0.77 *	0.7 *	0.37*
Average	g	0.73	0.72	0.67	0.72	0.69	0.7	0.62

Conclusion

The problem of imbalanced data sets arises frequently. In this work, we survey some methods proposed by the ML community to solve the problem, we discuss

some limitations of these methods and we propose a wrapper for weighting training instances technique as a more effective solution to problem. Our method allows improved identification of difficult small classes in predictive analysis, while keeping the classification ability of the other classes in an acceptable level. In a following study, we will examine the proposed technique in multi-class datasets.

References

1. Aha, D. (1997). Lazy Learning. Dordrecht: Kluwer Academic Publishers.
2. Blake, C., Keogh, E. & Merz, C.J. (1998). UCI Repository of machine learning databases [http:// www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California.
3. Chawla N., Bowyer K., Hall L., Kegelmeyer W. (2002), SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research 16, 321 - 357.
4. Domingos P. (1998), How to get a free lunch: A simple cost model for machine learning applications. Proc. AAAI-98/ICML98, Workshop on the Methodology of Applying Machine Learning, pp1-7.
5. Domingos P. & Pazzani M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. Machine Learning, 29, 103-130.
6. Domingos, P. (1999). MetaCost: A General Method for Making Classifiers Cost-Sensitive. Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining, 155-164. ACM Press.
7. Fan, W., Stolfo, S.J., Zhang, J. & Chan, P.K. (1999). AdaCost: Misclassification costsensitive boosting. Proceedings of the Sixteenth International Conference on Machine Learning, 97-105. San Francisco: Morgan Kaufmann.
8. Japkowicz N. (2000), The class imbalance problem: Significance and strategies. In Proceedings of the International Conference on Artificial Intelligence, Las Vegas.
9. Japkowicz N. and Stephen, S. (2002), The Class Imbalance Problem: A Systematic Study Intelligent Data Analysis, Volume 6, Number 5.
10. Kotsiantis, S., Pierrakeas, C., and Pintelas, P., Preventing student dropout in distance learning systems using machine learning techniques, LNAI, Vol 2774, pp 267-274, 2003
11. Kotsiantis S., Kanellopoulos, D. Pintelas, P. (2006), Handling imbalanced datasets: A review, GESTS International Transactions on Computer Science and Engineering, Vol.30 (1), pp. 25-36.
12. Kubat, M., Holte, R. and Matwin, S. (1998), 'Machine Learning for the Detection of Oil Spills in Radar Images', Machine Learning, 30:195-215.
13. Ling, C., & Li, C. (1998). 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. AAAI Press.
14. Provost, F. and Fawcett, T. (2001). Robust Classification for Imprecise Environments", Machine Learning, 42, 203–231.
15. Quinlan J.R. (1993), C4.5: Programs for machine learning. Morgan Kaufmann, San Francisco.
16. Schapire R., Singer Y. and Singhal A. (1998). Boosting and Rochhio applied to text filtering. In SIGIR'98.
17. Witten Ian H. and Frank Eibe (2005) "Data Mining: Practical machine learning tools and techniques", 2nd Edition, Morgan Kaufmann, San Francisco, 2005.