MCS-based Balancing Techniques for Skewed Classes: an Empirical Comparison

Maria Teresa Ricamato, Claudio Marrocco and Francesco Tortorella
DAEIMI Universitá degli Studi di Cassino via G. Di Biasio 43, 03043 Cassino, Italy
{*mt.ricamato, c.marrocco, tortorella*}*@unicas.it*

Abstract

The class imbalance is a critical problem in classification tasks related to many real world applications. A large number of solutions were proposed in literature, both at the algorithmic and data levels. In this paper we analyze the second kind of approach and, in particular, we focus our attention on the use of Multiple Classification Systems where each classifier is trained on a dataset containing the minority class and a subset of the majority class samples. The aim of this approach is to avoid the drawbacks of other methods, commonly used in this context, which force a balanced distribution by oversampling the minority class. We compare the results obtained applying different realizations of the method on the UCI Repository datasets.

1. Introduction

A large number of applications in real world are characterized by the class imbalance problem where one class is under-represented relatively to the others. Learning algorithms that do not consider class imbalance tend to be overwhelmed by the major class and ignore the minor one [5]. This is a critical problem in many applications such as fraud/intrusion detection, biometric verification, risk management, medical diagnosis/monitoring and text categorization.

Some empirical studies have shown that the optimal distribution is shifted to include more minority-class samples than the natural distribution. As an example, experiments in [10] demonstrate that, when learning from a balanced class distribution, a C4.5 classifier generally comes up with fewer but more accurate classification rules for the minority class than for the majority class.

Several solutions to imbalanced problem class were proposed in literature, both at the algorithmic and data levels. The former approach uses modified standard learning algorithms [6], while the latter uses different forms of re-sampling that alter the internal distribution of the datasets. One of the most employed approach to this aim is SMOTE [4] which oversamples minority class by creating synthetic examples "similar" to the minority class samples. The samples introduced by this technique could not be representative of the actual distribution of the minority class, e. g. when the generated samples are not physically consistent. This could lead to an erroneous fitting of the distribution of the minority class.

In this paper we describe and compare different approaches which avoid such drawbacks. The rationale is to build a Multiple Classifier System (MCS), tailored on imbalanced data which exploits a suitable balancing technique. The main idea is to part the majority class N in several subsets N_i, each consisting of as many samples as the minority class $|N_i| = |P|$, where P indicates the minority class. So we build an MCS aggregating several classifiers, each trained on a set $N_i \cup P$. In this way, we use balanced training set without creating artificial data. Finally, we combine the outputs to obtain the final result.

It is worth noting that in applications characterized by imbalance classes, frequently, key parameters of the environment, such as costs, benefits and class priors are not known or changing over time. For this reason the most effective tool for measuring the quality of classifier is by the *Receiver Operating Characteristic curve* (ROC curve) which provides the system performance in terms of *true positive rate* and *false positive rate* at all operating points (i.e., for all the possible values of the decision threshold). A commonly used index which summarizes the overall performance of the classifier under all the possible operating points is the *Area Under the ROC Curve* (AUC). So, we analyze classification systems performance using this parameter.

The paper is organized as follows: in Section 2 we describe the different sampling techniques and how we

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

evaluate their performance, in Section 3 we show our experimental results, while in Section 4 we draw some conclusions.

2. Balancing Techniques and their evaluation

Starting from an imbalanced dataset, there are several ways to part it in an ensemble of balanced training sets for building the MCS. However, all the possible techniques can be related to three basic schemes: BalanceWithReplacement (*BWR*) [8], BalanceCascade (*BC*) [8], BalanceWithoutReplacement (*BWOR*) [9].

BWR (Fig.1) and BWOR (Fig.2) could be considered *parallel* schemes. BWR randomly extracts *T* subsets from *N* (with replacement), while BWOR extracts *M* subsets such that $M = \left\lfloor \frac{|N|}{|P|} \right\rfloor$, where $|N|$ and $|P|$ indicate the number of negative and positive samples, respectively. For each subset N_i ($1 \leq i \leq T$ or $1 \leq i \leq M$), a classifier C_i is trained using N_i and P.

BalanceCascade (Fig 3) is a serial scheme. At each step we build a balanced subset N_i from the original dataset D, we train a classifier C_i with $N_i + P$, and we fix C_i's false positive rate $f_p = \sqrt[T-1]{\frac{|P|}{|N|}}$. Then we eliminate from D all negative samples that are correctly classified by C_i. At next step, we extract the balanced subset N_i from a dataset with a less number of negative samples.

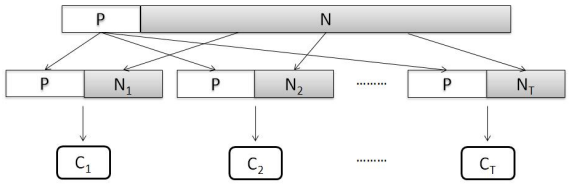

Figure 1. BalanceWithReplacement

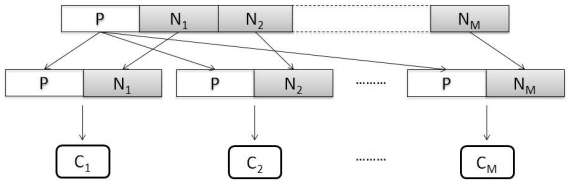

Figure 2. BalanceWithoutReplacement

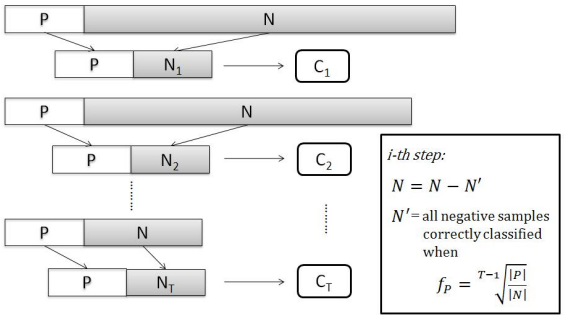

Figure 3. BalanceCascade

After the classifiers are trained, we have to combine the outputs to obtain the final results. Each classifier has a confidence degree, which is a continuous-valued output. We use this output for mean rule, so we compute the average value of the outputs.

The described balancing schemes do not create artificial samples, but use only samples in the dataset, combined in different ways. However, when dealing with imbalanced class, the most used approach is SMOTE [4], an algorithm based on oversampling the minority class. It creates synthetic examples taking each minority class sample and introducing new samples along the line segments joining each of the *k* minority class nearest neighbors, where $k \cong \frac{N}{P} - 1$. 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.

As previously said, we use the AUC to evaluate the classification performance. We prefer AUC to other, more diffused parameters, such as the accuracy, because of the increased sensitivity and discriminating degree of the AUC, the independence from the decision threshold and the invariance to prior class probabilities [3, 7]. The AUC of a classifier f could be easily evaluated by numerically integrating the corresponding ROC. However, there is a useful equivalence between the AUC of a classifier and the *Wilcoxon-Mann-Whitney statistic* (WMW) which is defined as:

$$\frac{\sum_{i=1}^{|P|}\sum_{j=1}^{|N|}[f(\mathbf{p}_i) > f(\mathbf{n}_j)]}{|P|\cdot|N|} \tag{1}$$

where $p_i \in P$, $n_j \in N$ and $[\cdot]$ is the Iverson bracket.

3. Experimental Results

In this section, we show the results obtained on eleven datasets of the UCI Repository of Machine Learning Database [2], with different imbalancing degrees. We consider two class classification problem, so when a dataset has more than two classes, we take the class with smaller number of samples as minority class and we consider the other samples as the majority class.

We have adopted three basic types of individual classifiers: Gentle Adaboost[1], a more robust and stable version of real AdaBoost, Logistic Classifier [1] which maximizes the likelihood criterion using the logistic (sigmoid) function, and Fisher Classifier which finds the linear discriminant function between the classes in a dataset by minimizing the errors in the least square sense. We used the PrTools toolbox[2] for Matlab for both Logistic and Fisher Classifiers.

For every datasets and every method, we used 10-fold cross validation. All the results are reported in terms of AUC averaged over the 10 folds.

It is worth noting that the number of classifiers depends on the method that we consider. For *BWOR* it varies with the imbalancing degree, so it differs for each datasets. For *BWR* and *BC* methods, it is fixed to 4 as suggested in [8]). For SMOTE, the generated samples are added to the original data set, thus obtaining an artificially based training set. Also in this case we used 10-fold cross validation. For the sake of comparison, we have also considered the classifiers trained with the original imbalanced datasets; we denote the relative results with "natural distribution" (*ND*).

The results are organized in the following way: Tables 1- 3 show the mean AUC for each dataset and each base classifier. In order to evaluate the performance for each method overall the datasets we have also used the relative rank values. We have considered 15 possible combinations between method and classifiers. For each dataset, we calculate the rank value (from 1 to 15), in the following way: the highest mean AUC gets rank 1, the second highest the rank 2, and so on. If there are tied AUC, the average of the ranks involved is assigned to all AUC tied for a given rank. Then for each method we calculate the mean of rank values, then are shown in Table 4 in increasing order. The lower the value, the better the related method.

Let us analyze the results obtained for each classifier. Gentle Adaboost has better performance than the other classifiers. It is the most robust and stable classifier among those considered; notwithstanding, its performance improve when we use balanced datasets (*BWR* and *BWOR*). SMOTE performance is comparable to the natural distribution, so in this case, adding artificial samples is not helpful for our problem. The worst method for Gentle Adaboost is *BC*. Similar results are shown by the other classifiers: in both cases, *BWR* and *BWOR* perform better than SMOTE while the latter is quite comparable with natural distribution. In summary, MCS based balancing techniques have much better performance than system based on dataset with natural distribution, while this only partially holds when the classes are balanced by adding artificial samples. It is interesting to note that MCS based balancing techniques are beneficial also for effective classifiers such as Adaboost. Moreover, the same techniques where applied to simple classifiers produce classification system with performance comparable to more powerful classifiers, but with a lower complexity.

Table 1. Gentle Adaboost Results

Dataset	BWOR	BWR	BC	SMOTE	ND
Abalone	0.854	0.847	0.829	0.833	0.837
Balance	0.738	0.728	0.748	0.807	0.811
Cmc	0.701	0.705	0.681	0.703	0.699
Haberman	0.648	0.655	0.655	0.649	0.637
Housing	0.825	0.819	0.800	0.782	0.806
Mf-morph	0.903	0.913	0.895	0.897	0.893
PageBlock	0.996	0.995	0.979	0.981	0.996
Pima	0.765	0.792	0.779	0.775	0.772
SatImage	0.963	0.960	0.959	0.956	0.958
Vehicle	0.872	0.873	0.881	0.882	0.873
Wpbc	0.755	0.747	0.733	0.763	0.751

Table 2. Fisher Results

Dataset	BWOR	BWR	BC	SMOTE	ND
Abalone	0.824	0.824	0.796	0.822	0.815
Balance	0.613	0.576	0.654	0.628	0.602
Cmc	0.723	0.722	0.712	0.720	0.720
Haberman	0.691	0.708	0.705	0.685	0.697
Housing	0.751	0.747	0.749	0.742	0.739
Mf-morph	0.796	0.797	0.733	0.790	0.784
PageBlock	0.988	0.988	0.984	0.989	0.984
Pima	0.828	0.827	0.827	0.828	0.828
SatImage	0.755	0.754	0.764	0.755	0.750
Vehicle	0.858	0.855	0.863	0.851	0.855
Wpbc	0.860	0.834	0.845	0.834	0.828

[1] available on http://research.graphicon.ru/machine-learning/gml-adaboost-matlab-toolbox.html.

[2] available on http://www.prtools.org

Table 3. Logistic Classifier Results

Dataset	BWoR	BWR	BC	SMOTE	ND
Abalone	0.846	0.845	0.812	0.839	0.848
Balance	0.612	0.576	0.628	0.628	0.603
Cmc	0.722	0.722	0.703	0.720	0.721
Haberman	0.701	0.707	0.718	0.696	0.695
Housing	0.754	0.755	0.722	0.743	0.745
Mf-morph	0.798	0.799	0.714	0.794	0.765
PageBlock	0.989	0.879	0.974	0.749	0.770
Pima	0.827	0.825	0.828	0.827	0.827
SatImage	0.765	0.766	0.737	0.761	0.769
Vehicle	0.861	0.857	0.864	0.858	0.862
Wpbc	0.770	0.757	0.796	0.815	0.837

Table 4. Rank Value

Classifier/Method	Rank Value
Ada/BWR	6.00
Ada/BWOR	6.32
Log/BWOR	6.77
Fis/BWOR	7.23
Ada/ND	7.36
Ada/SMOTE	7.41
Log/ND	8.09
Ada/BC	8.14
Fis/BC	8.23
Log/BWR	8.23
Fis/BWR	8.41
Fis/SMOTE	9.00
Log/SMOTE	9.05
Log/BC	9.82
Fis/ND	9.95

4. Conclusions

In this paper we have analyzed different approaches dealing with imbalanced class problem. The experimental results demonstrate that MCS-based balancing techniques have better performance than classifiers trained on natural distribution. Moreover, they obtain better results than balancing techniques such as SMOTE, based on artificially oversampling the minority class. Therefore, we can conclude that MCS-based balancing techniques allow us to exploit all the information contained in the original dataset (without generating new, frequently unreliable samples) and to effectively compensate the skew between classes.

References

[1] J. A. Anderson. *Classification, Pattern Recognition and Reduction of Dimensionality*, chapter Logistic Discrimination, pages 169–191. Handbook of Statistics, Amsterdam: North-Holland, 1982.

[2] A. Asuncion and D. J. Newman. UCI machine learning repository. 2007.

[3] A. P. Bradley. The use of the area under the ROC curve in the evaluation of machine learning algorithms. *Pattern Recognition*, 30:1145–1159, 1997.

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

[5] N. V. Chawla, N. Japkowicz, and K. Aleksander. Editorial: Special Issue on Learning from Imbalanced Data Sets. *Sigkdd Explorations*, 6(1):1–6, June 2004.

[6] N. V. Chawla, A. Lazarevic, H. L., and K. Bowyer. SMOTEBoost: improving prediction of the minority class in boosting. *7th European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 107–119, 2003.

[7] J. Huang and C. X. Ling. Using AUC and Accuracy in Evaluating Learning Algorithms. *IEEE Transactions on Knowledge and data engineering*, 17:299–310, March 2005.

[8] X.-Y. Liu, J. Wu, and Z.-H. Zhou. Exploratory Under-Sampling for Class-Imbalance Learning. *Proceedings of the Sixth International Conference on Data Mining*, 2006.

[9] M. Molinara, M. T. Ricamato, and F. Tortorella. Facing Imbalanced Classes through Aggregation of Classifiers. *Proceedings of the 14th International Conference on Image Analysis and Processing*, pages 43–48, 2007.

[10] G. M. Weiss and F. Provost. The Effect of Class Distribution on Classifier Learning. Technical report, Department of Computer Science, Rutgers University, 2001.