Racing for unbalanced methods selection

Andrea DAL POZZOLO[1], Olivier CAELEN[2],
Serge WATERSCHOOT[2], and Gianluca BONTEMPI[1]

[1] Machine Learning Group, Computer Science Department, Faculty of Sciences ULB,
Université Libre de Bruxelles, Brussels, Belgium
[2] Business Analytics Competence Center, Atos Worldline, Belgium

Abstract. State-of-the-art classification algorithms suffer when the data is skewed towards one class. This led to the development of a number of techniques to cope with unbalanced data. However, as confirmed by our experimental comparison, no technique appears to work consistently better in all conditions. We propose to use a racing method to select adaptively the most appropriate strategy for a given unbalanced task. The results show that racing is able to adapt the choice of the strategy to the specific nature of the unbalanced problem and to select rapidly the most appropriate strategy without compromising the accuracy.

Keywords: Unbalanced data; Fraud detection; Racing

1 Introduction

Learning from unbalanced datasets is a difficult task since most learning algorithms are not designed to cope with a large difference between the number of cases belonging to different classes [2]. The unbalanced nature of the data is typical of many applications such as medical diagnosis, text classification and oil spills detection. Credit card fraud detection [16, 6, 17] is another well-known instance of highly unbalanced problem since (fortunately) the number of fraudulent transactions is typically much smaller than legitimate ones. In literature several methods for dealing with unbalanced datasets have been proposed. They essentially belong to the following categories: sampling, ensemble and distance-based.

The ratio between majority and minority class is not the only factor that determines the difficulty of a classification/detection task. Another influential factor is the amount of overlapping of the classes of interest [9]. Other studies [10, 14] showed that some methods are superior to others under certain conditions.

All these supports the idea that under different conditions, such as different datasets and algorithms, the best methods may change. In particular, in credit card fraud detection, the fraudulent behaviour evolves over the time changing the distribution of the frauds and a method that worked well in the past could become inaccurate afterward. Since in real large variate tasks it is hard to know a priori the nature of the unbalanced tasks, the user is recommended to test all techniques with a consequent high computational cost. In this context we propose a racing strategy to accelerate the search of the strategy in an unbalanced problem.

In this paper we first review the most common methods for dealing with unbalanced data in a supervised context. Then we make an exhaustive comparison of these methods on a real credit-card fraud dataset and nine public benchmark datasets. The results show that there is no balancing technique which is consistently the best one and that the best method depends on the algorithm applied as well as the the dataset used. For this reason, we propose the adoption of a racing strategy [15] to automatically select the most adequate technique for a given dataset. The rationale of the racing strategy consists in testing in parallel a set of alternative balancing strategies on a subset of the dataset and to remove progressively the alternatives which are significantly worse. Our results show that by adopting a racing strategy we are able to select in an efficient manner either the best balancing method or a method which is not significantly different from the best one. Moreover racing is able to reduce consistently the computation needed before finding the right methods for the dataset.

2 Strategies for unbalanced classification

Let us consider a binary classification task where the distribution of the target class is highly skewed. Let us call the majority class negative (coded as 0) and the minority class as positive (coded as 1). When the data is unbalanced, standard machine learning algorithms tend to be overwhelmed by the majority class [10]. There are several methods that deal with this problem and we can distinguish them into the following main categories: sampling, ensemble, distance-based and hybrid.

Sampling techniques do not take into consideration any class information in removing or adding observations, yet they are easy to implement and to understand. *Undersampling* [7] consists in down-sizing the majority class by removing observations at random until the dataset is balanced. In an unbalanced problem it is realistic to assume that many observations of the majority class are redundant and that by removing some of them at random the resulting distribution should not change much. *Oversampling* [7] consists in up-sizing the small class at random decreasing the level of class imbalance. By replicating the minority class until the two classes have equal frequency, oversampling increases the risk of overfitting [7] by biasing the model towards the minority class. *SMOTE* [5] over-samples the minority class by generating synthetic minority examples in the neighborhood of observed ones. The idea is to form new minority examples by interpolating between examples of the same class. This has the effect of creating clusters around each minority observation.

Ensemble methods combine an unbalanced method with a classifier to explore the majority and minority class distribution. *BalanceCascade* [13] is a supervised strategy to undersample the majority class. This method iteratively removes the majority class instances that are correctly classified by a boosting algorithm.

EasyEnsemble [13] learns different aspects of the original majority class in an unsupervised manner. This is done by creating different balanced training sets by *Undersampling*, learning a model for each dataset and then combining all predictions as in bagging.
The following methods make use of distance measures between input points either to undersample or to remove noisy and borderline examples of each class. *Tomek link* [19] removes observations from the negative class that are close to the positive region in order to return a dataset that presents a better separation between the two classes. Let us consider two input examples x_i and x_j belonging to different classes, and let $d(x_i, x_j)$ be their distance. A (x_i, x_j) pair is called a *Tomek link* if there is no example x_k, such that $d(x_i, x_k) < d(x_i, x_j)$ or $d(x_j, x_k) < d(x_i, x_j)$. Negative examples that are Tomek links are then removed reducing the majority class. *Condensed Nearest Neighbor (CNN)* [8] is used to select a subset S from the original unbalanced set T which is consistent with T in the sense that S classifies T correctly with the one-nearest neighbor rule. Since noisy examples are likely to be misclassified, many of them will be added to the S set which means CNN rule is extremely sensitive to noise [21]. *One-sided Selection* (OSS) [11] is an undersampling method resulting from the application of Tomek links followed by the application of CNN. *Edited Nearest Neighbor (ENN)* [20] removes any example whose class label differs from the class of at least two of its three nearest neighbors. In this way majority examples that fall in the minority region and isolated minority examples are removed. *Neighborhood Cleaning Rule (NCL)* [12] modifies the ENN method by increasing the role of data cleaning. Firstly, NCL removes negatives examples which are misclassified by their 3-nearest neighbors. Secondly, the neighbors of each positive examples are found and the ones belonging to the majority class are removed.
A set of hybrid strategies can be easily created by combining sampling, ensemble and distance base techniques. In particular we consider the following hybrids: *SMOTEnsemble* (EasyEnsemble with SMOTE), *EnnSmote* (SMOTE after ENN), *TomekUnder* (Undersampling after Tomek), *TomekEasyEnsemble* (EasyEnsemble after Tomek) and *TomekSMOTE*: (SMOTE after Tomek).

3 Racing for strategy selection

The variety of approaches discussed in Section 2 suggests that in a real situation where we have no prior information about the data distribution, it is difficult to decide which unbalanced strategy to use. In this case testing all alternatives is not an option either because of the associated computational cost.

A possible solution comes form the adoption of the Racing approach which was proposed in [15] to perform efficiently model selection in a learning task. The principle of Racing consists in testing in parallel a set of alternatives and using a statistical test to determine if an alternative is significantly worse than the others. In that case such alternative is discarded from the competition, and the computational effort is devoted to differentiate the remaining ones. Historically the first example of Racing method is called Hoeffding Race since it relies on the Hoeffding theorem to decide when a model is significantly worse than the others. The *F-race* version was proposed in [4] and combines the Friedman test with Hoeffding Races [15] to eliminate inferior candidates as soon as enough statistical evidence arises against them. In F-race, the Friedman test is used to check whether there is evidence that at least one of the candidates is significantly different from others and post-tests are applied to eliminate those candidate that are significantly worse than the best one.

Here we adopt F-Race to search efficiently for the best strategy for unbalanced data. The candidates are assessed on different subsets of data and, each time a new assessment is made, the Friedman multiple test is used to dismiss significantly inferior candidates. We used a 10 fold cross validation to provide the assessment measure to the race. If a candidate is significantly better than all the others than the race is terminated without the need of using the whole dataset. In case there is not evidence of worse/better methods, the race terminates when the entire dataset is explored and the best candidate is the one with the best average result.

4 Experimental results

We tested the 15 strategies for unbalanced data discussed in Section 2 on the datasets of the following table:

Dataset ID	Dataset name	Size	Input	Prop 1	Class 1
1	breastcancer	698	10	34.52%	class =4
2	car	1727	6	3.76%	class = Vgood
3	forest	38501	54	7.13%	class = Cottonwood/Willow
4	letter	19999	16	3.76%	letter = W
5	nursery	12959	8	2.53%	class = very_recom
6	pima	768	8	34.89%	class = 1
7	satimage	6433	36	9.73%	class = 4
8	women	1472	9	22.62%	class = long-term
9	spam	4601	57	42.14%	class =1
10	fraud	527026	51	0.39%	Fraud = 1

The first 9 datasets are from UCI [1] repository. UCI datasets that have originally more than two response classes are transformed into binary by picking one class as the minority and joining all the others to form the majority class.

The credit card fraud dataset was provided by a payment service provider in Belgium. The fraud dataset is not available to the public because of the confidentiality of the data. In fraud detection it is important to have a high Precision and Recall, therefore we used F-measure as performance metric since it is able to combines Precision and Recall into a single metric.

We started by testing on the fraud dataset different supervised algorithms such as Random Forest, Neural Network, Support Vector Machine and Naive Bayes. For the sake of reproducibility we used the implementation provided in the R software [18] with default parameters.

Each algorithm was first tested on the entire fraud dataset using a 10 fold cross validation for all the strategies. Naive Bayes is the only algorithm whose performance is not sensitive to the adopted strategy and comparable to the unbalanced strategy (i.e. the strategy leaving the data in the original status).

In Figure 1 we can notice that for Support Vector Machine and Random Forest the group of techniques that include oversampling (i.e. Oversampling and SMOTE-related strategies) are performing better than the others. For most of the algorithms, distance-based strategy (ENN, NCL, CNN, OSS and Tomek) perform as bad as the unbalanced case. The highest F-measure is reached using Random Forest algorithm with SMOTEnsemble.

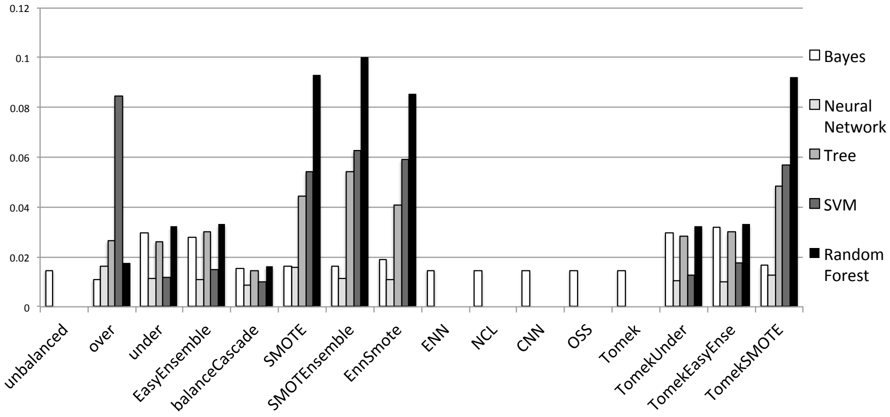


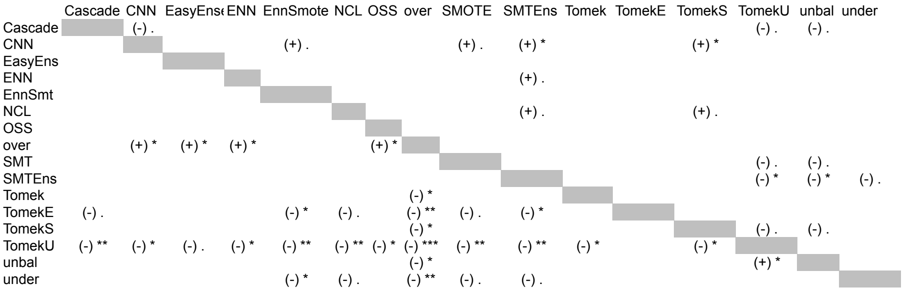
Fig. 1. Comparison of strategies for unbalanced data for the Fraud dataset in terms of F-measure (the higher the better).

At this point we tested our strategies for unbalanced data on the public UCI datasets. On some of them (e.g. Breast Cancer) no significant differences between strategies was detected.

We used the Friedman test to detect differences in the methods across all datasets. A post-hoc test based upon paired t-test of the ranks was used to decide which methods are significantly different from each other (Figure 2). From Figure 2 we can notice that with Random Forest, SMOTEnsemble was statistically better than many of unbalanced strategies, while oversampling was the best for SVM.

What emerges from this study is that there is no single strategy which is coherently superior to all others in all conditions (i.e. algorithm and dataset). Even if sometimes it is possible to find a strategy that is statistically better than others it is computationally demanding testing all strategies in several dataset and algorithms.

We decided to adopt the F-race algorithm implemented in [3] (with default parameters) to automatise the way to select the best strategy for unbalanced data. In Table 1 we used the F-race method to automatically select the unbalanced strategy.



	Cascade	CNN	EasyEns	ENN	EnnSmote	NCL	OSS	over	SMOTE	SMTEns	Tomek	TomekE	TomekS	TomekU	unbal	under
Cascade		(-) .												(-) .	(-) .	
CNN					(+) .				(+) .	(+) *			(+) *			
EasyEns																
ENN										(+) .						
EnnSmt																
NCL										(+) .			(+) .			
OSS																
over		(+) *	(+) *	(+) *			(+) *									
SMT														(-) .	(-) .	
SMTEns														(-) *	(-) *	(-) .
Tomek								(-) *								
TomekE	(-) .				(-) *	(-) .		(-) **	(-) .	(-) *						
TomekS								(-) *						(-) .	(-) .	
TomekU	(-) **	(-) *	(-) .	(-) *	(-) **	(-) **	(-) *	(-) ***	(-) **	(-) **	(-) *		(-) *			
unbal								(-) *						(+) *		
under					(-) *	(-) .		(-) **	(-) .	(-) .						

Fig. 2. Comparison of strategies using a post-hoc Friedman test in terms of F-measure over multiple datasets. RF results are above the diagonal, SVM below. A cell above the diagonal is marked as (+) if the rank difference between the method in the column and the one in the row is significantively positive, (-) otherwise. Below, (+) denotes that the method in the row is significantively better than the one in the column, (-) otherwise. The matrix contains the level of significance of the t-test annotated as follows: ***, **, * and . for α = 0.001, 0.01, 0.05, 0.1. A cell is left empty if the test is not significant.

Let us remark that for almost all datasets F-race is able to return the best method according to the cross validation (CV) assessment. In the case of Pima and Spam datasets, F-race returns a sub-optimal strategy whose accuracy is however not significantly different from the best one (Pvalue greater than 0.05).

The main advantage of Racing is that bad methods are not tested on the whole dataset reducing the computation needed. Taking into consideration the 15 methods and the unbalanced case, in a 10 fold cross validation we have 160 tests to make (10 folds x 16 methods). In the case of F-race the number of total tests depends upon how many folds are needed before F-race finds the best method. The *Gain* column of Table 1 shows the computational gain (in percentage of the the CV tests) obtained by using F-race. Apart from the Breast Cancer dataset in all the other cases F-race allows a significant computational saving with no loss in performance.

5 Conclusion

Recent literature in data mining and machine learning is plenty of research works on strategies to deal with unbalanced data. However a definitive answer on the best strategy to adopt is yet to come. Our experimental results support the idea that the final performance is extremely dependent on the data nature and distribution.

This consideration has lead us to adopt the F-race strategy where different candidates (unbalanced methods) are tested simultaneously. We have showed that this algorithm is able to select few candidates that perform better than other without exploring the whole dataset. F-race was able to get results similar to the cross validation for most of the dataset.

Dataset	Algo	Exploration	Method	N test	Gain	Mean	Sd	Pval
Fraud	RF	best CV	SMOTEnsemble	160	-	0.100	0.016	-
		F-race	SMOTEnsemble	44	73%			
	SVM	best CV	over	160	-	0.084	0.017	-
		F-race	over	46	71%			
Breast Cancer	RF	best CV	balanceCascade	160	-	0.963	0.035	-
		F-race	balanceCascade	160	0%			
	SVM	best CV	under	160	-	0.957	0.038	-
		F-race	under	160	0%			
Car	RF	best CV	OSS	160	-	0.970	0.039	-
		F-race	OSS	108	33%			
	SVM	best CV	over	160	-	0.944	0.052	-
		F-race	over	93	42%			
Forest	RF	best CV	balanceCascade	160	-	0.911	0.012	-
		F-race	balanceCascade	60	63%			
	SVM	best CV	ENN	160	-	0.809	0.011	-
		F-race	ENN	64	60%			
Letter	RF	best CV	balanceCascade	160	-	0.981	0.010	-
		F-race	balanceCascade	73	54%			
	SVM	best CV	over	160	-	0.953	0.022	-
		F-race	over	44	73%			
Nursery	RF	best CV	SMOTE	160	-	0.809	0.047	-
		F-race	SMOTE	76	53%			
	SVM	best CV	over	160	-	0.875	0.052	-
		F-race	over	58	64%			
Pima	RF	best CV	under	160	-	0.691	0.045	-
		F-race	under	136	15%			
	SVM	best CV	EasyEnsemble	160	-	0.675	0.071	0.107063
		F-race	TomekUnder	110	31%	0.672	0.067	
Satimage	RF	best CV	balanceCascade	160	-	0.719	0.033	-
		F-race	balanceCascade	132	18%			
	SVM	best CV	balanceCascade	160	-	0.662	0.044	-
		F-race	balanceCascade	90	44%			
Spam	RF	best CV	SMOTE	160	-	0.942	0.015	-
		F-race	SMOTE	122	24%			
	SVM	best CV	SMOTEnsemble	160	-	0.917	0.018	0.266028
		F-race	SMOTE	135	16%	0.9178762	0.02	
Women	RF	best CV	TomekUnder	160	-	0.488	0.051	-
		F-race	TomekUnder	150	6%			
	SVM	best CV	EnnSmote	160	-	0.492	0.073	-
		F-race	EnnSmote	102	36%			

Table 1. Comparison of Cross Validation and F-race results with Random Forest and Support Vector Machines in terms of F-measure.

As far as the fraud dataset is concerned, we found SMOTEnsemble together with RandomForest to be the best strategy. In this dataset the unbalanced strategy chosen had a big impact on the accuracy of the results. However, as the frauds evolve over the time the same method could become sub-optimal in the future. In this context the F-race contribution to the selection of the best strategy is crucial in order to have a detection system that adapts quickly to the new data distribution. Within the UCI datasets we noticed that some tasks are much easier (high accuracy) than the others and they may not have an unbalanced method that performs significantly better than the others.

Acknowledgment
The work of Andrea Dal Pozzolo was supported by the Doctiris programe of Innoviris (Brussels Institute for Research and Innovation), Belgium.

References

1. D. N. A. Asuncion. UCI machine learning repository, 2007.

2. G. Batista, A. Carvalho, and M. Monard. Applying one-sided selection to unbalanced datasets. *MICAI 2000: Advances in Artificial Intelligence*, pages 315–325, 2000.
3. M. Birattari. *race: Racing methods for the selection of the best*, 2012. R package version 0.1.59.
4. M. Birattari, T. Stützle, L. Paquete, and K. Varrentrapp. A racing algorithm for configuring metaheuristics. In *Proceedings of the genetic and evolutionary computation conference*, pages 11–18, 2002.
5. N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Arxiv preprint arXiv:1106.1813*, 2011.
6. P. Clark and T. Niblett. The cn2 induction algorithm. *Machine learning*, 3(4):261–283, 1989.
7. C. Drummond, R. Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on Learning from Imbalanced Datasets II*. Citeseer, 2003.
8. P. E. Hart. The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, 1968.
9. R. C. Holte, L. E. Acker, B. W. Porter, et al. Concept learning and the problem of small disjuncts. In *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, volume 1. Citeseer, 1989.
10. N. Japkowicz and S. Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.
11. M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In *MACHINE LEARNING-INTERNATIONAL WORKSHOP THEN CONFERENCE-*, pages 179–186. MORGAN KAUFMANN PUBLISHERS, INC., 1997.
12. J. Laurikkala. Improving identification of difficult small classes by balancing class distribution. *Artificial Intelligence in Medicine*, pages 63–66, 2001.
13. X. Liu, J. Wu, and Z. Zhou. Exploratory undersampling for class-imbalance learning. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 39(2):539–550, 2009.
14. K. T. M. Lin and X. Yao. A dynamic sampling approach to training neural networks for multi-class imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems*, 24:647–660, 2013.
15. O. Maron and A. Moore. Hoeffding races: Accelerating model selection search for classification and function approximation. *Robotics Institute*, page 263, 1993.
16. L. Olshen and C. Stone. Classification and regression trees. *Wadsworth International Group*, 1984.
17. J. Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.
18. R Development Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2011. ISBN 3-900051-07-0.
19. I. Tomek. Two modifications of cnn. *IEEE Trans. Syst. Man Cybern.*, 6:769–772, 1976.
20. D. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, (3):408–421, 1972.
21. D. Wilson and T. Martinez. Reduction techniques for instance-based learning algorithms. *Machine learning*, 38(3):257–286, 2000.