COST SENSITIVE AND PREPROCESSING FOR CLASSIFICATION WITH IMBALANCED DATA-SETS: SIMILAR BEHAVIOUR AND POTENTIAL HYBRIDIZATIONS

Victoria López[1], Alberto Fernández[2], María José del Jesus[2] and Francisco Herrera[1]
[1]*Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, 18071 Granada, Spain*
[2]*Dept. of Computer Science, University of Jaén, Jaén, Spain*

Keywords: Classification, Imbalanced data-sets, Preprocessing, Sampling, Cost-sensitive learning, Hybridizations.

Abstract: The scenario of classification with imbalanced data-sets has supposed a serious challenge for researchers along the last years. The main handicap is related to the large number of real applications in which one of the classes of the problem has a few number of examples in comparison with the other class, making it harder to be correctly learnt and, what is most important, this minority class is usually the one with the highest interest.
In order to address this problem, two main methodologies have been proposed for stressing the significance of the minority class and for achieving a good discrimination for both classes, namely preprocessing of instances and cost-sensitive learning. The former rebalances the instances of both classes by replicating or creating new instances of the minority class (oversampling) or by removing some instances of the majority class (undersampling); whereas the latter assumes higher misclassification costs with samples in the minority class and seek to minimize the high cost errors. Both solutions have shown to be valid for dealing with the class imbalance problem but, to the best of our knowledge, no comparison between both approaches have ever been performed.
In this work, we carry out a full exhaustive analysis on this two methodologies, also including a hybrid procedure that tries to combine the best of these models. We will show, by means of a statistical comparative analysis developed with a large collection of more than 60 imbalanced data-sets, that we cannot highlight an unique approach among the rest, and we will discuss as a potential research line the use of hybridizations for achieving better solutions to the imbalanced data-set problem.

1 INTRODUCTION

In many supervised learning applications, there is a significant difference between the class prior rates, that is the probability a particular example belongs to a particular class. This situation is known as the class imbalance problem (Chawla et al., 2004; Sun et al., 2009; He and Garcia, 2009) and it is dominant in a high number of real problems including, but not limited to, telecommunications, WWW, finances, ecology, biology, medicine and so on; for which it is considered as one of the top problems in data mining (Yang and Wu, 2006). Furthermore, it is worth to point out that the positive or minority class is usually the one that has the highest interest from the learning point of view and it also implies a great cost when it is not well classified (Elkan, 2001).

The hitch with imbalanced data-sets is that standard classification learning algorithms are often biased towards the majority classes and therefore there is a higher misclassification rate in the minority class instances. Therefore, throughout the last years, many solutions have been proposed to deal with this problem, which can be categorized into two major groups:

1. **Data Sampling**: in which the training instances are modified in such a way as to produce a more balanced class distribution that allow classifiers to perform in a similar manner to standard classification (Batista et al., 2004; Chawla et al., 2002).

2. **Algorithmic Modification**: this procedure is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues (Zadrozny and Elkan, 2001). We must also stress in this case the use of cost-sensitive learning solutions, which basically assume higher misclassification costs with samples in the rare class and seek to minimize the high cost errors (Domingos,

López V., Fernández A., José del Jesus M. and Herrera F. (2012).
COST SENSITIVE AND PREPROCESSING FOR CLASSIFICATION WITH IMBALANCED DATA-SETS: SIMILAR BEHAVIOUR AND POTENTIAL HYBRIDIZATIONS.
In *Proceedings of the 1st International Conference on Pattern Recognition Applications and Methods*, pages 98-107
DOI: 10.5220/0003751600980107

Copyright © SciTePress

1999; Zadrozny et al., 2003).

Works in imbalanced classification usually focus on the development of new algorithms along one of the categories previously mentioned. However, there is not a study that exhaustively compares solutions from one category to another making difficult the selection of one kind of algorithm when classifying. The aim of this contribution is to develop a thorough experimental study to analyze the possible differences between preprocessing techniques and cost-sensitive learning for addressing classification with imbalanced data. In addition, we also present in the comparison a hybrid procedure that combines those two approaches to check whether there is a synergy between them.

As baseline classifier, we will use the C4.5 decision tree generating algorithm (Quinlan, 1993); firstly because it has been widely used to deal with imbalanced data-sets (Su and Hsiao, 2007; Drown et al., 2009; García et al., 2009), and secondly since it has been included as one of the top-ten data-mining algorithms (Wu and Kumar, 2009).

In order to analyze the oversampling and undersampling methodologies, we will focus on two of the most robust approaches such as the “Synthetic Minority Over-sampling TEchnique” (SMOTE) (Chawla et al., 2002) and its variant with the Wilson’s Edited Nearest Neighbour (ENN) rule (Wilson, 1972), as suggested by their performance among many different situations (Batista et al., 2004; Fernández et al., 2008). Regarding cost-sensitive methods, we have selected the C4.5-CS algorithm (Ting, 2002), which modifies the computation of the split criteria for C4.5 (normalized information gain) to take into account the a priori probabilities according to the number of samples for each class.

In this work, we focus on imbalanced binary classification problems, having selected a benchmark of 66 problems from KEEL data-set repository[1] (Alcalá-Fdez et al., 2011). We perform our experimental study focusing on the precision of the models using the Area Under the ROC curve (AUC) (Huang and Ling, 2005). This study is carried out using nonparametric tests to check whether there exist significant differences among the obtained results (Demšar, 2006; García and Herrera, 2008).

This contribution is organized as follows: first, Section 2 presents the problem of imbalanced data-sets and the metric we have employed in this context whereas Section 3 describes the main methodologies to address the problem: the preprocessing methods used, cost-sensitive classification and a wrapper approach to combine both. In Section 4 an analysis of preprocessing techniques versus cost-sensitive learning approaches can be found. Finally, the conclusions of this work are commented in Section 5.

2 IMBALANCED DATA-SETS IN CLASSIFICATION

In this section, we first introduce the problem of imbalanced data-sets and then we present the evaluation metrics for this type of classification problem which differs from usual measures in classification.

2.1 The Problem of Imbalanced Data-sets

In the classification problem field, the scenario of imbalanced data-sets appears frequently. The main property of this type of classification problem is that the examples of one class outnumber the examples of the other one (Japkowicz and Stephen, 2002; Guo et al., 2008; Sun et al., 2009; He and Garcia, 2009). The minority class usually represents the most important concept to be learnt, since it might be associated with exceptional and significant cases (Weiss, 2004), or because the data acquisition of these examples is costly (Weiss and Tian, 2008).

Since most of the standard learning algorithms consider a balanced training set, this situation may cause the obtention of suboptimal classification models, i.e. a good coverage of the majority examples whereas the minority ones are misclassified frequently; therefore, those algorithms which obtains a good behaviour in the framework of standard classification do not necessarily achieves the best performance for imbalanced data-sets (Fernandez et al., 2010). There are several reasons behind this behaviour which are enumerated below:

1. The use of global performance measures for guiding the search process, such as standard accuracy rate, may benefit the covering of the majority examples.
2. Classification rules that predict the positive class are often highly specialized and thus their coverage is very low, hence they are discarded in favour of more general rules, i.e. those that predict the negative class.
3. It is always difficult to distinguish between noise examples and minority class examples and they can be completely ignored by the classifier.

In recent years, the imbalanced learning problem has received a high attention in the machine learn-

[1] http://www.keel.es/data-sets.php

ing community. Specifically, regarding real world domains the importance of the imbalance learning problem is growing, since it is a recurring problem in many applications. As a few examples, we may find very high resolution airbourne imagery (Chen et al., 2011), face recognition (Kwak, 2008) and especially medical diagnosis (Lo et al., 2008; Mazurowski et al., 2008). It is important to remember that the minority class usually represents the concept of interest and it is the most difficult to obtain from real data, for example patients with illnesses in a medical diagnosis problem; whereas the other class represents the counterpart of that concept (healthy patients).

2.2 Evaluation in Imbalanced Domains

The evaluation criteria is a key factor in both assessing the classification performance and guiding the classifier modelling. In a two-class problem, the confusion matrix (shown in Table 1) records the results of correctly and incorrectly recognized examples of each class.

Table 1: Confusion matrix for a two-class problem.

	Positive prediction	Negative prediction
Positive class	True Positive (TP)	False Negative (FN)
Negative class	False Positive (FP)	True Negative (TN)

Traditionally, accuracy rate (Eq. (1)) has been the most commonly used empirical measure. However, in the framework of imbalanced data-sets, accuracy is no longer a proper measure, since it does not distinguish between the number of correctly classified examples of different classes. Hence, it may lead to erroneous conclusions, i.e., a classifier achieving an accuracy of 90% in a data-set with an IR value of 9, is not accurate if it classifies all examples as negatives.

$$Acc = \frac{TP+TN}{TP+FN+FP+TN} \tag{1}$$

According to the previous issue, in this work we use the Area Under the Curve (AUC) metric (Huang and Ling, 2005), which can be defined as

$$AUC = \frac{1+TP_{rate}-FP_{rate}}{2} \tag{2}$$

where TP_{rate} is the percentage of positive cases correctly classified as belonging to the positive class and FP_{rate} is the percentage of negative cases misclassified as belonging to the positive class.

3 ADDRESSING CLASSIFICATION WITH IMBALANCED DATA: PREPROCESSING AND COST-SENSITIVE LEARNING

A large number of approaches have been proposed to deal with the class imbalance problem. These approaches can be categorized into two groups: the internal approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration (Barandela et al., 2003; Sun et al., 2007; Ducange et al., 2010) and external approaches that preprocess the data in order to diminish the effect of their class imbalance (Batista et al., 2004; Estabrooks et al., 2004).

Regarding this, in this section we first introduce the main features of preprocessing techniques, focusing on SMOTE (Chawla et al., 2002) and SMOTE+ENN (Batista et al., 2004), which will be used along the experimental study. Next, we describe cost-sensitive learning and the C4.5-CS methodology (Ting, 2002). Finally, we present a framework to automatically detect a threshold for preprocessing using an underlying algorithm, in this case, a cost-sensitive approach.

3.1 Preprocessing Imbalanced Data-sets: Resampling Techniques

In the specialized literature, we can find some papers about resampling techniques studying the effect of changing the class distribution to deal with imbalanced data-sets.

Those works have proved empirically that, applying a preprocessing step in order to balance the class distribution, is usually an useful solution (Batista et al., 2004; Fernández et al., 2008; Fernández et al., 2010). Furthermore, the main advantage of these techniques is that they are independent of the underlying classifier.

Resampling techniques can be categorized into three groups or families:

1. *Undersampling Methods*, which create a subset of the original data-set by eliminating instances (usually majority class instances).
2. *Oversampling Methods*, which create a superset of the original data-set by replicating some instances or creating new instances from existing ones.
3. *Hybrids Methods*, which combine both sampling approaches.

Among these categories, there are several proposals where the simplest preprocessing are non heuristic methods such as random undersampling and random oversampling. In the first case, the major drawback is that it can discard potentially useful data, that could be important for the induction process. For random oversampling, several authors agree that this method can increase the likelihood of occurring overfitting, since it makes exact copies of existing instances.

According to the previous facts, more sophisticated methods have been proposed. Among them, SMOTE (Chawla et al., 2002) has become one of the most renowned approaches in this area. In brief, its main idea is to create new minority class examples by interpolating several minority class instances that lie together for oversampling the training set.

With this technique, the positive class is oversampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours are randomly chosen. This process is illustrated in Figure 1, where x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbours and r_1 to r_4 the synthetic data points created by the randomized interpolation.

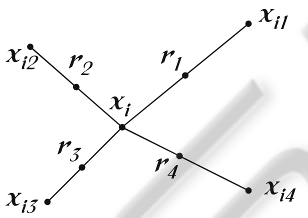

Figure 1: An illustration of how to create the synthetic data points in the SMOTE algorithm.

However, in oversampling techniques, and especially for the SMOTE algorithm, the problem of over generalization is largely attributed to the way in which it creates synthetic samples. Specifically, SMOTE generates the same number of synthetic data samples for each original minority example and does so without consideration to neighboring examples, which increases the occurrence of overlapping between classes (Wang and Japkowicz, 2004). For this reason we also consider a hybrid approach in this work, "SMOTE + ENN", where the Wilson's ENN Rule (Wilson, 1972) is used after the SMOTE application to remove from the training set any example misclassified by its three nearest neighbours.

3.2 Cost-sensitive Learning

Cost-sensitive learning takes into account the variable cost of a misclassification of the different classes (Domingos, 1999; Zadrozny et al., 2003). In this case, a cost matrix codifies the penalties $C(i,j)$ of classifying examples of one class as a different one; if we use the notation 1 for minority and 0 for majority class, $C(i,i) = TN$ or TP. These misclassification cost values can be given by domain experts, or learned via other approaches (Sun et al., 2009; Sun et al., 2007). Specifically, when dealing with imbalanced problems it is usually of most interest to recognize the positive instances rather than the negative ones and therefore, the cost when misclassifying a positive instance is higher than the cost of misclassifying a negative one.

Given the cost matrix, an example should be classified into the class that has the minimum expected cost. This is the minimum expected cost principle. The expected cost $R(i|x)$ of classifying an instance x into class i (by a classifier) can be expressed as:

$$R(i|x) = \sum_j P(j|x) \cdot C(i,j) \tag{3}$$

where $P(j|x)$ is the probability estimation of classifying an instance into class j. That is, the classifier will classify an instance x into positive class if and only if:

$$P(0|x) \cdot (C(1,0) - C(0,0)) \leq P(1|x)(C(0,1) - C(1,1))$$

Therefore, any given cost-matrix can be converted to one with $C(0,0) = C(1,1) = 0$. Under this assumption, the classifier will classify an instance x into positive class if and only if:

$$P(0|x) \cdot C(1,0) \leq P(1|x) \cdot C(0,1)$$

As $P(0|x) = 1 - P(1|x)$, we can obtain a threshold p^* for the classifier to classify an instance x into positive if $P(1|x) \geq p^*$, where

$$p^* = \frac{C(1,0)}{C(1,0) - C(0,1)} = \frac{FP}{FP + FN} \tag{4}$$

Another possibility is to "rebalance" the original training examples the ratio of:

$$p(1)FN : p(0)FP \tag{5}$$

where $p(1)$ and $p(0)$ are the prior probability of the positive and negative examples in the original training set.

In summary, two main general approaches have been proposed to deal with cost-sensitive problems:

1. **Direct Methods:** The main idea of building a direct cost-sensitive learning algorithm is to directly introduce and utilize misclassification costs into the learning algorithms.

   For example, in the context of decision tree induction, the tree-building strategies are adapted to minimize the misclassification costs. The cost information is used to: (1) choose the best attribute to split the data (Ling et al., 2004; Riddle et al., 1994); and (2) determine whether a subtree should be pruned (Bradford et al., 1998). On the other hand, other approaches based on genetic algorithms can incorporate misclassification costs in the fitness function (Turney, 1995).

2. **Meta-learning:** This methodology implies the integration of a "preprocessing" mechanism for the training data or a "postprocessing" of the output, in such a way that the original learning algorithm is not modified. Cost-sensitive meta-learning can be further classified into two main categories: *thresholding* and *sampling*, which are based on expressions (4) and (5) respectively:

   - **Thresholding** is based on the Bayes decision theory that assign instances to the class with minimum expected cost, as introduced above. For example, a typical decision tree for a binary classification problem assigns the class label of a leaf node depending on the majority class of the training samples that reach the node. A cost-sensitive algorithm assigns the class label to the node that minimizes the classification cost (Domingos, 1999; Zadrozny and Elkan, 2001).
   - **Sampling** is based on modifying the training data-set. The most popular technique lies in resampling the original class distribution of the training data-set according to the cost decision matrix by means of undersampling/oversampling (Zadrozny et al., 2003) or assigning instance weights (Ting, 2002). These modifications have shown to be effective and can also be applied to any cost insensitive learning algorithm (Zhou and Liu, 2006).

In this work, we will make use of the cost-sensitive C4.5 decision tree (C4.5-CS) proposed in (Ting, 2002). This method changes the class distribution such that the tree induced is in favor of the class with high weight/cost and is less likely to commit errors with high cost. Specifically, the computation of the split criteria for C4.5 (normalized information gain) is modified to take into account the a priori probabilities according to the number of samples for each class.

The standard greedy divide-and-conquer procedure for inducing minimum error trees can then be used without modification, except that $W_j(t)$ (6) is used instead of $N_j(t)$ (number of instances of class j) in the computation of the test selection criterion in the tree growing process and the error estimation in the pruning process.

$$W(j) = C(j)\frac{N}{\sum_i C(i)N_i} \tag{6}$$

C4.5-CS also introduces another optional modification that alters the usual classification process after creating the decision tree. Instead of classifying using the minimum error criteria, it is advisable to classify using the expected misclassification cost in the last part of the classification procedure. The expected misclassification cost for predicting class i with respect to the instance x is given by

$$EC_i(x) \propto \sum_j W_j(t(x))cost(i,j) \tag{7}$$

where $t(x)$ is the leaf of the tree that instance x falls into and $W_j(t)$ is the total weight of class j training instances in node t.

3.3 Hybridization. Automatically Countering Imbalance

The different solutions used to deal with the imbalanced problem have been presented in the previous subsections. So the question now is "Can we use both techniques together and achieve better results?".

In this section we describe a procedure to integrate the cost-sensitive learning and preprocessing approaches into one, quite similar to the one proposed in (Chawla et al., 2008), which consists in a wrapper paradigm that discovers the amount of resampling needed for a data-set based on optimizing evaluation functions which can include the cost associated to the classification. This wrapper infrastructure applies cross-validation to first discover the best amounts of undersampling and oversampling, applies the preprocessing algorithms with the amounts estimated and finally runs the algorithm used over the preprocessed data-set. Figure 2 shows the algorithm procedure.

The undersampling estimation starts with no undersampling for all majority classes and obtains baseline results on the training data. Then it traverses through the search space of undersampling percentages in decrements of Sample Decrement (in this case 10%), in a greedy iterative fashion, to increase performance over the minority classes without sacrificing performance on the majority class.

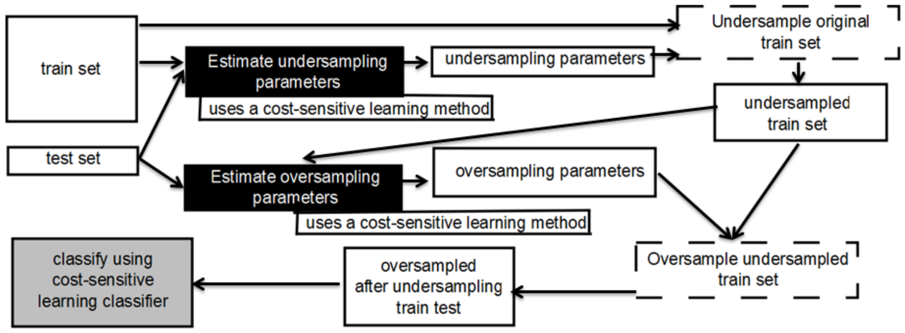
train set
test set
Estimate undersampling parameters
uses a cost-sensitive learning method
undersampling parameters
Undersample original train set
undersampled train set
Estimate oversampling parameters
uses a cost-sensitive learning method
oversampling parameters
Oversample undersampled train set
oversampled after undersampling train test
classify using cost-sensitive learning classifier

Figure 2: Illustration on the Wrapper Undersample SMOTE Algorithm. Dashed lines means resampling actions, black boxes represent the parameters estimation and the final result is in grey.

The oversampling algorithm evaluates different amounts of SMOTE at steps of 100% (the number of examples from the minority class). This is a greedy search, and at each step the new performance estimates become the new baseline. That is, the initial baseline is the performance obtained via the Wrapper Undersample. If SMOTE=100% improves the performance over that baseline by some margin Increment Min, then the performance achieved at SMOTE=100% becomes the new baseline. The amount of SMOTE is then incremented by Sample Increment, and another evaluation is performed to check if the performance increase at new SMOTE amount is at least greater than Increment Min. This process repeats, greedily, until no performance gains are observed.

However, there is an important caveat to the search to avoid being trapped in a local maximum. If the average does not improve by 5% we have to verify that we have not settled on a local maximum. In order to do so, we look ahead two more steps at increasing amounts of SMOTE. If the look-ahead does not result in an improvement in performance, then the amount of SMOTE is reset to the value discovered prior to the look-ahead. This is done to allow SMOTE to introduce additional examples with the aim of improving performance. However, if the addition of examples does not help, then we go back to using the lesser amount of SMOTE discovered prior to the look-ahead.

We can use different measures to evaluate the performance of the classifier to estimate the sampling parameters. In our case, different from (Chawla et al., 2008), we use cost-sensitive learning algorithms as base classifiers, and therefore a logical evaluation criteria is the cost itself. Cost is calculated as shown in Equation 8 when we assume $C(+|+) = C(-|-) = 0$ (as it is usual in imbalanced classification).

$$cost = FNrate \cdot C(-|+) + FPrate \times C(+|-) \quad (8)$$

4 EXPERIMENTAL STUDY

In this section, we will perform an analysis to determine the performance of the different alternatives used for imbalanced classification. Our aim is to analyze three different issues:

1. The improvement obtained by preprocessing datasets and cost-sensitive learning over the original algorithm.
2. The possible differences between the rebalancing techniques versus cost-sensitive learning and in which cases.
3. Whether a hybrid methodology that combines a preprocessing approach and a cost-sensitive learning algorithm supposes a positive synergy and enables the achievement of more accurate results.

First, we present our experimental framework with the data-set employed in our analysis and the statistical tests that will allow us to support the extracted findings. Then, we will show the results from our study and we will discuss the main issues that will arise from the aforementioned analysis.

4.1 Experimental Framework

In order to analyze the preprocessing approach against the cost-sensitive learning strategy, we have selected 66 data-sets from the KEEL data-set repository[2] (Alcalá-Fdez et al., 2011). These data-sets are summarized in Table 2, where we denote the number of examples (#Ex.), number of attributes (#Atts.), class name of each class (positive and negative), class distribution and IR.

To develop the different experiments we consider a *5-folder cross-validation model*, i.e., five random partitions of data with a 20% and the combination of

[2] http://www.keel.es/data-sets.php

SCITEPRESS Science and Technology Publications

Table 2: Summary of imbalanced data-sets.

Data-sets	**#Ex.**	**#Atts.**	**Class (-,+)**	**%Class(-; +)**	**IR**
Glass1	214	9	(build-win-non_float-proc; remainder)	(35.51, 64.49)	1.82
Ecoli0vs1	220	7	(im; cp)	(35.00, 65.00)	1.86
Wisconsin	683	9	(malignant; benign)	(35.00, 65.00)	1.86
Pima	768	8	(tested-positive; tested-negative)	(34.84, 66.16)	1.90
Iris0	150	4	(Iris-Setosa; remainder)	(33.33, 66.67)	2.00
Glass0	214	9	(build-win-float-proc; remainder)	(32.71, 67.29)	2.06
Yeast1	1484	8	(nuc; remainder)	(28.91, 71.09)	2.46
Vehicle1	846	18	(Saab; remainder)	(28.37, 71.63)	2.52
Vehicle2	846	18	(Bus; remainder)	(28.37, 71.63)	2.52
Vehicle3	846	18	(Opel; remainder)	(28.37, 71.63)	2.52
Haberman	306	3	(Die; Survive)	(27.42, 73.58)	2.68
Glass0123vs456	214	9	(non-window glass; remainder)	(23.83, 76.17)	3.19
Vehicle0	846	18	(Van; remainder)	(23.64, 76.36)	3.23
Ecoli1	336	7	(im; remainder)	(22.92, 77.08)	3.36
New-thyroid2	215	5	(hypo; remainder)	(16.89, 83.11)	4.92
New-thyroid1	215	5	(hyper; remainder)	(16.28, 83.72)	5.14
Ecoli2	336	7	(pp; remainder)	(15.48, 84.52)	5.46
Segment0	2308	19	(brickface; remainder)	(14.26, 85.74)	6.01
Glass6	214	9	(headlamps; remainder)	(13.55, 86.45)	6.38
Yeast3	1484	8	(me3; remainder)	(10.98, 89.02)	8.11
Ecoli3	336	7	(imU; remainder)	(10.88, 89.12)	8.19
Page-blocks0	5472	10	(remainder; text)	(10.23, 89.77)	8.77
Ecoli034vs5	200	7	(p,imL,imU; om)	(10.00, 90.00)	9.00
Yeast2vs4	514	8	(cyt; me2)	(9.92, 90.08)	9.08
Ecoli067vs35	222	7	(cp,omL,pp; imL,om)	(9.91, 90.09)	9.09
Ecoli0234vs5	202	7	(cp,imS,imL,imU; om)	(9.90, 90.10)	9.10
Glass015vs2	172	9	(build-win-non_float-proc,tableware, build-win-float-proc; ve-win-float-proc)	(9.88, 90.12)	9.12
Yeast0359vs78	506	8	(mit,me1,me3,erl; vac,pox)	(9.88, 90.12)	9.12
Yeast02579vs368	1004	8	(mit,cyt,me3,vac,erl; me1,exc,pox)	(9.86, 90.14)	9.14
Yeast0256vs3789	1004	8	(mit,cyt,me3,exc; me1,vac,pox,erl)	(9.86, 90.14)	9.14
Ecoli046vs5	203	6	(cp,imU,omL; om)	(9.85, 90.15)	9.15
Ecoli01vs235	244	7	(cp,im; imS,imL,om)	(9.83, 90.17)	9.17
Ecoli0267vs35	224	7	(cp,imS,omL,pp; imL,om)	(9.82, 90.18)	9.18
Glass04vs5	92	9	(build-win-float-proc,containers; tableware)	(9.78, 90.22)	9.22
Ecoli0346vs5	205	7	(cp,imL,imU,omL; om)	(9.76, 90.24)	9.25
Ecoli0347vs56	257	7	(cp,imL,imU,pp; om,omL)	(9.73, 90.27)	9.28
Yeast05679vs4	528	8	(me2; mit,me3,exc,vac,erl)	(9.66, 90.34)	9.35
Ecoli067vs5	220	6	(cp,omL,pp; om)	(9.09, 90.91)	10.00
Vowel0	988	13	(hid; remainder)	(9.01, 90.99)	10.10
Glass016vs2	192	9	(ve-win-float-proc; build-win-float-proc, build-win-non_float-proc,headlamps)	(8.89, 91.11)	10.29
Glass2	214	9	(Ve-win-float-proc; remainder)	(8.78, 91.22)	10.39
Ecoli0147vs2356	336	7	(cp,im,imU,pp; imS,imL,om,omL)	(8.63, 91.37)	10.59
Led7digit02456789vs1	443	7	(0,2,4,5,6,7,8,9; 1)	(8.35, 91.65)	10.97
Glass06vs5	108	9	(build-win-float-proc,headlamps; tableware)	(8.33, 91.67)	11.00
Ecoli01vs5	240	6	(cp,im; om)	(8.33, 91.67)	11.00
Glass0146vs2	205	9	(build-win-float-proc,containers,headlamps, build-win-non_float-proc;ve-win-float-proc)	(8.29, 91.71)	11.06
Ecoli0147vs56	332	6	(cp,im,imU,pp; om,omL)	(7.53, 92.47)	12.28
Cleveland0vs4	177	13	(0; 4)	(7.34, 92.66)	12.62
Ecoli0146vs5	280	6	(cp,im,imU,omL; om)	(7.14, 92.86)	13.00
Ecoli4	336	7	(om; remainder)	(6.74, 93.26)	13.84
Yeast1vs7	459	8	(nuc; vac)	(6.72, 93.28)	13.87
Shuttle0vs4	1829	9	(Rad Flow; Bypass)	(6.72, 93.28)	13.87
Glass4	214	9	(containers; remainder)	(6.07, 93.93)	15.47
Page-blocks13vs2	472	10	(graphic; horiz.line,picture)	(5.93, 94.07)	15.85
Abalone9vs18	731	8	(18; 9)	(5.65, 94.25)	16.68
Glass016vs5	184	9	(tableware; build-win-float-proc, build-win-non_float-proc,headlamps)	(4.89, 95.11)	19.44
Shuttle2vs4	129	9	(Fpv Open; Bypass)	(4.65, 95.35)	20.5
Yeast1458vs7	693	8	(vac; nuc,me2,me3,pox)	(4.33, 95.67)	22.10
Glass5	214	9	(tableware; remainder)	(4.20, 95.80)	22.81
Yeast2vs8	482	8	(pox; cyt)	(4.15, 95.85)	23.10
Yeast4	1484	8	(me2; remainder)	(3.43, 96.57)	28.41
Yeast1289vs7	947	8	(vac; nuc,cyt,pox,erl)	(3.17, 96.83)	30.56
Yeast5	1484	8	(me1; remainder)	(2.96, 97.04)	32.78
Ecoli0137vs26	281	7	(pp,imL; cp,im,imU,imS)	(2.49, 97.51)	39.15
Yeast6	1484	8	(exc; remainder)	(2.49, 97.51)	39.15
Abalone19	4174	8	(19; remainder)	(0.77, 99.23)	128.87

4 of them (80%) as training and test. For each data-set we consider the average results of the five partitions. The data-sets used in this study use the partitions provided by the repository in the imbalanced classification data-set section[3].

Furthermore, we have to identify the misclassification costs associated to the positive and negative class for the cost-sensitive learning versions. If we misclassify a positive sample as a negative one the associated misclassification cost is the IR of the data-set ($C(+,-) = IR$) whereas if we misclassify a negative sample as a positive one the associated cost is 1 ($C(-,+) = 1$). The cost of classifying correctly is 0 ($C(+,+) = C(-,-) = 0$) because guessing the correct class should not penalize the built model.

Finally, statistical analysis needs to be carried out in order to find significant differences among the results obtained by the studied methods (Demšar, 2006; García et al., 2009; García et al., 2010). Since the study is split in parts comparing a group of algorithms, we use non-parametric statistical tests for multiple comparisons. Specifically, we use the Iman-Davenport test (Sheskin, 2006) to detect statistical differences among a group of results and the Shaffer post-hoc test (Shaffer, 1986) in order to find out which algorithms are distinctive among an $n \times n$ comparison.

Furthermore, we consider the average ranking of the algorithms in order to show graphically how good a method is with respect to its partners. This ranking is obtained by assigning a position to each algorithm depending on its performance for each data-set. The algorithm which achieves the best accuracy in a specific data-set will have the first ranking (value 1); then, the algorithm with the second best accuracy is assigned rank 2, and so forth. This task is carried out for all data-sets and finally an average ranking is computed as the mean value of all rankings.

4.2 Contrasting Preprocessing and Cost-sensitive Learning in Imbalanced Data-sets

Table 3 shows the average results in training and test together with the corresponding standard deviation for the seven versions of the C4.5 algorithm used in the study: the base classifier, the base classifier used over the preprocessed data-sets, the cost-sensitive version of the algorithm and the hybrid versions of it. We stress in **boldface** the best results achieved for the prediction ability of the different techniques.

Table 3: Average table of results using the AUC measure for the C4.5 variety of algorithms.

Algorithm	AUC_{tr}	AUC_{tst}
C4.5	0.8774 ± 0.0392	0.7902 ± 0.0804
C4.5 SMOTE	0.9606 ± 0.0142	0.8324 ± 0.0728
C4.5 SENN	0.9471 ± 0.0154	**0.8390** ± **0.0772**
C4.5CS	0.9679 ± 0.0103	0.8294 ± 0.0758
C4.5 Wr_SMOTE	0.9679 ± 0.0103	0.8296 ± 0.0763
C4.5 Wr_US	0.9635 ± 0.0139	0.8245 ± 0.0760
C4.5 Wr_SENN	0.9083 ± 0.0377	0.8145 ± 0.0712

From this table of results it can be observed that the highest average value corresponds to preprocess-

[3] http://www.keel.es/imbalanced.php

Table 4: Shaffer test for the C4.5 variety of algorithms using the AUC measure.

C4.5	None	SMOTE	SENN	CS	Wr_SMOTE	Wr_US	Wr_SENN
None	x	-(6.404E-6)	-(4.058E-8)	-(6.404E-6)	-(7.904E-6)	-(.00341)	=(.37846)
SMOTE	+(6.404E-6)	x	=(1.0)	=(1.0)	=(1.0)	=(1.0)	+(.04903)
SENN	+(4.058E-8)	=(1.0)	x	=(1.0)	=(1.0)	=(.22569)	+(.00152)
CS	+(6.404E-6)	=(1.0)	=(1.0)	x	=(1.0)	=(1.0)	+(.04903)
Wr_SMOTE	+(7.904E-6)	=(1.0)	=(1.0)	=(1.0)	x	=(1.0)	+(.04903)
Wr_US	+(.00341)	=(1.0)	=(.22569)	=(1.0)	=(1.0)	x	=(1.0)
Wr_SENN	=(.37846)	-(.04903)	-(.00152)	-(.04903)	-(.04903)	=(1.0)	x

ing approaches closely followed by the cost-sensitive learning approach and one version of the wrapper routine. This suggests the goodness of the preprocessing and cost-sensitive learning approaches.

In order to compare the results, a multiple comparison test is used to find the performance relationship between the different versions studied. The results of the statistical analysis of the C4.5 family are as follows. For the sake of a visual comparison, Figure 3 shows the average ranking for these approaches. Under the AUC measure, the Iman-Davenport test detects significant differences among the algorithms, since the *p*-value returned (1.88673E-10) is lower than our α-value (0.05). The differences found are analyzed with a Shaffer test, shown in Table 4. In this table, a “+” symbol implies that the algorithm in the row is statistically better than the one in the column, whereas “-” implies the contrary; “=” means that the two algorithms compared have no significant differences. In brackets, the adjusted *p*-value associated to each comparison is shown.

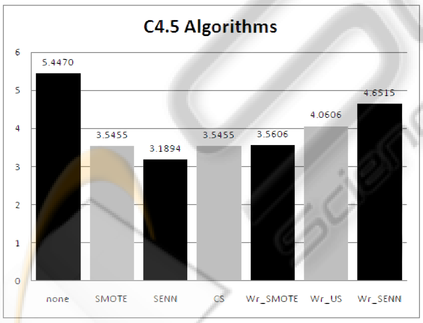

Figure 3: Average rankings using the AUC measure for the C4.5 variety of algorithms.

Observing the results from Tables 3 and 4, we conclude that the standard C4.5 approach is outperformed by most of the methodologies that deal with imbalanced data. All methodologies, the hybrid version that uses only an oversampling step with SMOTE+ENN, have significant differences versus the base C4.5 classifier. This analysis answers our first question of the study, that is, the classification performance is degraded in an imbalance scenario having a bias towards the majority class examples and the use of the aforementioned techniques (preprocessing and cost-sensitive learning) allow us to obtain a better discrimination of the examples of both classes resulting in an overall good classification for all concepts of the problem (positive and negative classes).

Comparing the results when applying preprocessing we can see that the performance of these methods is not statistically different for any of its versions. In addition, the performance of those preprocessing methods is also not different to the cost-sensitive learning version of C4.5. This second part of the study has reflected that the two employed solutions are quite similar between them and it was not possible to highlight one of them as the most adequate one for classification. For that reason, the question on which approach is preferable for addressing classification with imbalanced data-sets is still unresolved.

Finally, regarding the hybridization of cost-sensitive learning and preprocessing by using a wrapper routine, it can be seen that there are significant differences both between the different hybrid versions and with the other alternatives. The hybrid version that uses an oversampling step with SMOTE+ENN is outperformed by all the other versions except the base version. The rest of the hybrid versions are not statistically different from the performance of usual approaches for imbalanced classification. Therefore, we cannot state that the hybridization in decision trees produces a positive synergy between the two techniques. According to these results, the preliminary version of the hybrid technique can be further improved both applying a finest combination of the individual approaches or by using more specific methods with a better synergy between them.

5 CONCLUSIONS

In this work we have analyzed the behaviour of preprocessing and cost-sensitive learning in the framework of imbalanced data-sets in order to determine whether there are any significant differences between

SciTePress
Science and Technology Publications

both approaches and therefore which one of them is preferred and in which cases. Additionally, we have proposed a hybrid approach that integrates both approaches together.

First of all, we have determined that both methodologies improve the overall performance for the classification with imbalanced data, which was the expected behaviour. Next, the comparison between preprocessing techniques against cost-sensitive learning hints that there are no differences among the different preprocessing techniques. The statistical study, supported by a large collection of more than 60 imbalanced data-sets, lets us say that both preprocessing and cost-sensitive learning are good and equivalent approaches to address the imbalance problem.

Finally, we have shown that our preliminary versions of hybridization techniques are truly competitive with the standard methodologies. We must stress that this is a very interesting trend for research as there is still room for improvement regarding hybridization between preprocessing and cost-sensitive learning.

ACKNOWLEDGEMENTS

This work has been supported by the Spanish Ministry of Science and Technology under Projects TIN2011-28488 and TIN2008-06681-C06-02 and the Andalusian Research Plans P10-TIC-6858 and TIC-3928. V. López holds a FPU scholarship from Spanish Ministry of Education.

REFERENCES

Alcalá-Fdez, J., Fernández, A., Luengo, J., Derrac, J., García, S., Sánchez, L., and Herrera, F. (2011). KEEL data–mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multi–Valued Logic and Soft Computing*, 17(2-3):255–287.

Barandela, R., Sánchez, J. S., García, V., and Rangel, E. (2003). Strategies for learning in class imbalance problems. *Pattern Recognition*, 36(3):849–851.

Batista, G. E. A. P. A., Prati, R. C., and Monard, M. C. (2004). A study of the behaviour of several methods for balancing machine learning training data. *SIGKDD Explorations*, 6(1):20–29.

Bradford, J. P., Kunz, C., Kohavi, R., Brunk, C., and Brodley, C. E. (1998). Pruning decision trees with misclassification costs. In *Proceedings of the 10th European Conference on Machine Learning (ECML'98)*, pages 131–136.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over–sampling technique. *Journal of Artificial Intelligent Research*, 16:321–357.

Chawla, N. V., Cieslak, D. A., Hall, L. O., and Joshi, A. (2008). Automatically countering imbalance and its empirical relationship to cost. *Data Mining and Knowledge Discovery*, 17(2):225–252.

Chawla, N. V., Japkowicz, N., and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *SIGKDD Explorations*, 6(1):1–6.

Chen, X., Fang, T., Huo, H., and Li, D. (2011). Graph–based feature selection for object–oriented classification in VHR airborne imagery. *IEEE Transactions on Geoscience and Remote Sensing*, 49(1):353–365.

Demšar, J. (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research*, 7:1–30.

Domingos, P. (1999). Metacost: A general method for making classifiers cost–sensitive. In *Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99)*, pages 155–164.

Drown, D. J., Khoshgoftaar, T. M., and Seliya, N. (2009). Evolutionary sampling and software quality modeling of high-assurance systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 39(5):1097–1107.

Ducange, P., Lazzerini, B., and Marcelloni, F. (2010). Multi–objective genetic fuzzy classifiers for imbalanced and cost–sensitive datasets. *Soft Computing*, 14(7):713–728.

Elkan, C. (2001). The foundations of cost–sensitive learning. In *Proceedings of the 17th IEEE International Joint Conference on Artificial Intelligence (IJCAI'01)*, pages 973–978.

Estabrooks, A., Jo, T., and Japkowicz, N. (2004). A multiple resampling method for learning from imbalanced data sets. *Computational Intelligence*, 20(1):18–36.

Fernández, A., del Jesus, M. J., and Herrera, F. (2010). On the 2–tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data–sets. *Information Sciences*, 180(8):1268–1291.

Fernández, A., García, S., del Jesus, M. J., and Herrera, F. (2008). A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data–sets. *Fuzzy Sets and Systems*, 159(18):2378–2398.

Fernandez, A., García, S., Luengo, J., Bernadó-Mansilla, E., and Herrera, F. (2010). Genetics-based machine learning for rule induction: State of the art, taxonomy and comparative study. *IEEE Transactions on Evolutionary Computation*, 14(6):913–941.

García, S., Fernández, A., and Herrera, F. (2009). Enhancing the effectiveness and interpretability of decision tree and rule induction classifiers with evolutionary training set selection over imbalanced problems. *Applied Soft Computing*, 9:1304–1314.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2009). A study of statistical techniques and performance measures for genetics–based machine learning: accuracy and interpretability. *Soft Computing*, 13(10):959–977.

García, S., Fernández, A., Luengo, J., and Herrera, F. (2010). Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180(10):2044–2064.

García, S. and Herrera, F. (2008). An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons. *Journal of Machine Learning Research*, 9:2607–2624.

Guo, X., Dong, Y. Y. C., Yang, G., and Zhou, G. (2008). On the class imbalance problem. In *Proceedings of the 4th International Conference on Natural Computation (ICNC'08)*, volume 4, pages 192–201.

He, H. and Garcia, E. A. (2009). Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284.

Huang, J. and Ling, C. X. (2005). Using AUC and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 17(3):299–310.

Japkowicz, N. and Stephen, S. (2002). The class imbalance problem: a systematic study. *Intelligent Data Analysis Journal*, 6(5):429–450.

Kwak, N. (2008). Feature extraction for classification problems and its application to face recognition. *Pattern Recognition*, 41(5):1718–1734.

Ling, C. X., Yang, Q., Wang, J., and Zhang, S. (2004). Decision trees with minimal costs. In Brodley, C. E., editor, *Proceedings of the 21st International Conference on Machine Learning (ICML'04)*, volume 69 of *ACM International Conference Proceeding Series*, pages 69–77. ACM.

Lo, H.-Y., Chang, C.-M., Chiang, T.-H., Hsiao, C.-Y., Huang, A., Kuo, T.-T., Lai, W.-C., Yang, M.-H., Yeh, J.-J., Yen, C.-C., and Lin, S.-D. (2008). Learning to improve area-under-FROC for imbalanced medical data classification using an ensemble method. *SIGKDD Explorations*, 10(2):43–46.

Mazurowski, M. A., Habas, P. A., Zurada, J. M., Lo, J. Y., Baker, J. A., and Tourassi, G. D. (2008). Training neural network classifiers for medical decision making: The effects of imbalanced datasets on classification performance. *Neural Networks*, 21(2–3).

Quinlan, J. (1993). *C4.5: Programs for Machine Learning*. Morgan Kauffman.

Riddle, P., Segal, R., and Etzioni, O. (1994). Representation design and brute–force induction in a boeing manufacturing domain. *Applied Artificial Intelligence*, 8:125–147.

Shaffer, J. (1986). Modified sequentially rejective multiple test procedures. *Journal of the American Statistical Association*, 81(395):826–831.

Sheskin, D. (2006). *Handbook of parametric and nonparametric statistical procedures*. Chapman & Hall/CRC.

Su, C.-T. and Hsiao, Y.-H. (2007). An evaluation of the robustness of MTS for imbalanced data. *IEEE Transactions on Knowledge and Data Engeneering*, 19(10):1321–1332.

Sun, Y., Kamel, M. S., Wong, A. K. C., and Wang, Y. (2007). Cost–sensitive boosting for classification of imbalanced data. *Pattern Recognition*, 40(12):3358–3378.

Sun, Y., Wong, A. K. C., and Kamel, M. S. (2009). Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(4):687–719.

Ting, K. M. (2002). An instance–weighting method to induce cost–sensitive trees. *IEEE Transactions on Knowledge and Data Engineering*, 14(3):659–665.

Turney, P. (1995). Cost-sensitive classification: Empirical evaluation of a hybrid genetic decision tree induction algorithm. *Journal of Artificial Intelligence Research*, 2:369–409.

Wang, B. and Japkowicz, N. (2004). Imbalanced data set learning with synthetic samples. In *Proceedings of the IRIS Machine Learning Workshop*.

Weiss, G. M. (2004). Mining with rarity: a unifying framework. *SIGKDD Explorations*, 6(1):7–19.

Weiss, G. M. and Tian, Y. (2008). Maximizing classifier utility when there are data acquisition and modeling costs. *Data Mining and Knowledge Discovery*, 17(2):253–282.

Wilson, D. L. (1972). Asymptotic properties of nearest neighbor rules using edited data. *IEEE Transactions on Systems, Man and Cybernetics*, 2(3):408 –421.

Wu, X. and Kumar, V., editors (2009). *The Top ten algorithms in data mining*. Data Mining and Knowledge Discovery Series. Chapman and Hall/CRC press.

Yang, Q. and Wu, X. (2006). 10 challenging problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604.

Zadrozny, B. and Elkan, C. (2001). Learning and making decisions when costs and probabilities are both unknown. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD'01)*, pages 204–213.

Zadrozny, B., Langford, J., and Abe, N. (2003). Cost–sensitive learning by cost–proportionate example weighting. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03)*, pages 435–442.

Zhou, Z.-H. and Liu, X.-Y. (2006). Training cost–sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77.