Knowledge-Based Systems 42 (2013) 97–110

Contents lists available at SciVerse ScienceDirect

Knowledge-Based Systems

journal homepage: www.elsevier.com/locate/knosys

Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches

Alberto Fernández [a,*], Victoria López [b], Mikel Galar [c], María José del Jesus [a], Francisco Herrera [b]

[a] Department of Computer Science, University of Jaén, Jaén, Spain
[b] Department of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Granada, Spain
[c] Department of Automatic and Computation, Public University of Navarra, Spain

ARTICLE INFO

Article history:
Received 24 July 2012
Received in revised form 15 January 2013
Accepted 18 January 2013
Available online 29 January 2013

Keywords:
Imbalanced data-sets
Multi-classification
Pairwise learning
Preprocessing
Cost-sensitive learning

ABSTRACT

The imbalanced class problem is related to the real-world application of classification in engineering. It is characterised by a very different distribution of examples among the classes. The condition of multiple imbalanced classes is more restrictive when the aim of the final system is to obtain the most accurate precision for each of the concepts of the problem.

The goal of this work is to provide a thorough experimental analysis that will allow us to determine the behaviour of the different approaches proposed in the specialised literature. First, we will make use of binarization schemes, i.e., one versus one and one versus all, in order to apply the standard approaches to solving binary class imbalanced problems. Second, we will apply several ad hoc procedures which have been designed for the scenario of imbalanced data-sets with multiple classes.

This experimental study will include several well-known algorithms from the literature such as decision trees, support vector machines and instance-based learning, with the intention of obtaining global conclusions from different classification paradigms. The extracted findings will be supported by a statistical comparative analysis using more than 20 data-sets from the KEEL repository.

1. Introduction

This paper is focused on the framework of imbalanced data-sets, also known as the class imbalance problem, which refers to the case where one or more class, usually the ones that which is of interest, is under represented in the data-set [8]. This problem occurs in many real-world classification tasks and has been defined as a challenge for the Data Mining community [65]. The main difficulty in approaching this problem is that standard learning algorithms consider a balanced training set which induces a bias towards the majority classes [53].

In the research community concerned with imbalanced data-sets, recent efforts have been focused on two-class imbalanced problems [38,8,44]. However, multiple-class imbalanced learning problems appear frequently. The correct identification of each kind of concept in these problems, is equally important when considering different decisions that must be made in these areas [32,70].

When multiple classes are present, the solutions proposed for binary-class problems may not be directly applicable, or may achieve a lower performance than expected. For example, solutions at data level [5,15,35] suffer from the increased search space, and solutions at algorithm level [66,31] become more complicated, as the learning algorithm must consider several small classes. Additionally, learning from multiple classes itself implies a difficulty for Data Mining algorithms, as the boundaries among the classes may overlap, causing the level of performance to decrease. In this situation, we may proceed by transforming the original multiple-class problem into binary subproblems, which are easier to discriminate, via class binarization techniques such as pairwise learning, also known as the one versus one (OVO) approach [34], or one class versus all (OVA) [48].

In this paper we develop a complete experimental study for the classification of multiple-class imbalanced data-sets, aiming to determine the best approaches to be applied in this scenario. Our goal is to show the optimal combination between binarization techniques: either with preprocessing approaches (oversampling and undersampling), or with the use of cost-sensitive learning for multiple-class imbalanced data-sets, in the case of both the OVO and OVA approaches. We also seek to experimentally determine the degree of synergy achieved between the combination of "divide-and-conquer" techniques (OVO and OVA) and preprocessing/cost-sensitive learning by contrasting their results with those of the approaches specifically designed to address imbalanced classification problems in the scenario of multiple classes. This last

* Corresponding author. Tel.: +34 953 213016; fax: +34 953 212472.
E-mail addresses: alberto.fernandez@ujaen.es (A. Fernández), vlopez@decsai.ugr.es (V. López), mikel.galar@unavarra.es (M. Galar), mjjesus@ujaen.es (M.J. del Jesus), herrera@decsai.ugr.es (F. Herrera).

0950-7051/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.knosys.2013.01.018

aspect of the study will be carried out by selecting three recent approaches: Static-SMOTE [21]; a cost-sensitive algorithm that weights the examples a priori to globally balance the data-set [71]; and an ensemble learning approach for multi-class imbalanced problems [58].

In order to develop this empirical study, we have chosen three different algorithms from different paradigms of Data Mining, including Decision Trees with C4.5 [47], Support Vector Machines (SVMs) [10,46] and the well-known *k*-Nearest Neighbour (kNN) [42] as an Instance-Based Learning approach. We have selected a wide benchmark of 24 multiple-class data-sets from the KEEL data-set repository[1] [1] within the experimental framework. The performance measure is based on the average accuracy rate (the mean value for the accuracy of each single class) and the significance of the results is supported by proper statistical analysis as suggested in the literature [11,29].

This paper is organised as follows. First, Section 2 introduces the problem of imbalanced data. Next, Section 3 presents some classification strategies for multiple-class imbalanced data-sets, i.e., those developed ad hoc and those based on OVO and OVA with the combination of preprocessing or cost-sensitive learning. In Section 4 the experimental framework for the study is established. The complete experimental study is carried out in Section 5. A thorough discussion is presented in Section 6. Finally, Section 7 summarises the work and draws conclusions from it.

2. Imbalanced data-sets in classification

In this section, we will first introduce the problem of imbalanced data-sets, paying special attention to the context of multiple classes. Then, we will describe the techniques that have been applied in order to deal with the imbalanced problem, namely preprocessing and cost-sensitive learning. Finally, we will present the evaluation metrics for this kind of classification problem focusing, as natural, on those applied in the framework of multiple classes.

2.1. The problem of imbalanced data-sets

In the classification problem field, the scenario of imbalanced data-sets appears when the numbers of examples that represent the different classes are very different [8]. The minority classes are usually the most important concepts to be learnt, since they represent rare cases [61] or because the data acquisition of these examples is costly [62]. In this work we use the imbalance ratio (IR) [44], defined as the ratio of the number of instances of the majority (known as the negative class) and the minority class (known as the positive class), to organise the different data-sets according to this measure [30].

Most learning algorithms aim to obtain a model with a high prediction accuracy and a good generalisation capability. However, this inductive bias towards such a model poses a serious challenge to the classification of imbalanced data [53]. First, if the search process is guided by the standard accuracy rate, the covering of the majority examples is benefited; second, classification rules that predict the positive class are often highly specialised 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. Furthermore, it is not easy to distinguish between noise examples and minority class examples and they can be completely ignored by the classifier.

Furthermore, from Ref. [35] we may conclude that "*the degree of imbalance is not the only factor that hinders learning. As it turns out, data-set complexity is the primary determining factor of classification deterioration, which, in turn, is amplified by the addition of a relative imbalance*". Specifically, we must stress the significance of several factors such as overlapping between classes [38], lack of representative data [59], small disjuncts [60,45], noisy data [49], dataset shift [43] and other issues which have interdependent effects with data distribution (imbalance).

2.2. Addressing the imbalanced problem: preprocessing and cost-sensitive learning

A large number of approaches have been proposed to deal with the two-class imbalance problem, both for standard learning algorithms and for ensemble techniques [26,39,51]. These approaches can be categorised in three groups:

1. *Data level solutions:* The objective consists of rebalancing the class distribution by sampling the data space to diminish the effect of class imbalance, acting as an external approach [7,5,20,55,27,54].
2. *Algorithmic level solutions:* These solutions try to adapt specific classification algorithms to reinforce the learning towards the positive class. Therefore, they can be defined as internal approaches that create new algorithms or modify existing ones to take the class imbalance problem into consideration [66,4,12,31,9].
3. *Cost-sensitive solutions:* These incorporate approaches at data level, at algorithmic level, or at both levels jointly, considering higher misclassification costs for the examples of the positive class with respect to the negative class, and therefore, trying to minimise higher cost errors [14,56,67,52,69,71].

The advantage of data level solutions is that they are more versatile, as their use is independent of the classifier selected. Furthermore, we may preprocess all data-sets before-hand in order to use them to train different classifiers. In this manner, we only need to prepare the data once. There are different rebalancing methods with which to preprocess the training data that can be classified into three groups:

- Undersampling methods that create a subset of the original data-set by eliminating some of the examples of the majority class.
- Oversampling methods that create a superset of the original data-set by replicating some of the examples of the minority class or creating new ones from the original minority class instances.
- Hybrid methods that combine the two previous methods, eliminating some of the examples before or after resampling, in order to reduce overfitting.

Regarding algorithmic level approaches, the idea is to choose an appropriate inductive bias for a specific classifier. Also, recognition-based one-class learning is used to model a system by using only the examples of the target class in the absence of counter examples. This approach does not try to partition the hypotheses space with boundaries that separate positive and negative examples, but it attempts to establish the boundaries which surround the target concept, for example in SVMs.

Cost-sensitive learning takes into account the variable cost of a misclassification of the different classes. The cost-sensitive learning process tries to minimise the total cost of misclassifications, but in this cost function, the minority class gains importance. Therefore, cost-sensitive learning supposes that there is a cost matrix available for the different types of errors; however, given a data-set, this matrix is not usually given [52,53].

[1] http://www.keel.es/dataset.php.

Table 1
Acronyms for the methodologies used throughout the experimental study.

Acronym	Algorithm	Reference
NCL	Neighbourhood Cleaning Rule	[63]
OSS	One-Sided Selection	[40]
RUS	Random-Undersampling	[5]
TL	Tomek Links	[57]
ROS	Random-Oversampling	[5]
SMT	SMOTE	[7]
SMT–ENN	SMOTE + ENN	[5]
SL-SMT	Safe-Levels-SMOTE	[6]
CS	Instance Weighting Cost-Sensitive (for OVO)	[14,56,69]

In order to develop our experimental study, we have selected several representative methods from the specialised literature for the aforementioned groups, which deal with imbalanced classification. Specifically, we have chosen four undersampling and four oversampling techniques, and a cost-sensitive learning approach, which are summarised in Table 1. We must stress that the selection of these preprocessing mechanisms is based on previous studies and reviews on the topic in which their significance for this classification framework is highlighted [5,20,35]; furthermore, they are all available within the KEEL software tool (http://www.keel.es) [2].

- *Synthetic Minority Oversampling Technique (SMT) [7].* The minority 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 oversampling required, neighbours from the k-nearest neighbours are randomly chosen.
- *SMT + Edited* Nearest Neighbour (SMT + *ENN) [5].* When applying SMT, class clusters may not be well defined in cases where some majority class examples invade the minority class space. The opposite can also be true, since interpolating minority class examples can expand the minority class clusters, introducing artificial minority class examples too deeply into the majority class space. Inducing a classifier in such a situation can lead to overfitting. For this reason, the "SMT + ENN" hybrid approach applies Wilson's ENN rule [63] after SMOTE application to remove any example misclassified by its three nearest neighbours from the training set.
- *Safe-Level SMOTE (SL-SMT) [6].* As we described previously, SMT randomly synthesises minority instances along a line joining a minority instance and its selected nearest neighbours, ignoring nearby majority class instances. By contrast, SL-SMT carefully samples minority instances along the same line with a different weight degree, known as the safe level. The safe level is computed using the k-nearest minority class instances. Then, if the safe level of an instance is close to 0, the instance is considered to be noise. If it is close to k, the instance is considered safe. Therefore, it is a new variant from the original SMT preprocessing mechanism which aims at generating synthetic examples in safe areas of the training set.
- *Random-Oversampling (ROS) [5].* This is a non-heuristic method that aims to balance class distribution through the random replication of minority class examples. The disadvantage of this method is that it can increase the likelihood of overfitting occurring, as it makes exact copies of existing instances.
- *Random-Undersampling (RUS) [5].* Random-Undersampling is a non-heuristic method that aims to balance class distribution through the random elimination of majority class examples. The major drawback of Random-Undersampling is that this method can discard potentially useful data that could be important for the induction process.
- *Neighbourhood Cleaning Rule (NCL) [63].* For a two-class problem, this cleaning algorithm can be described in the following way: for each example e_i in the training set, its three nearest neighbours are found. If e_i belongs to the majority class and the classification given by its three nearest neighbours contradicts the original class of e_i, then e_i is removed. If e_i belongs to the minority class and its three nearest neighbours misclassify e_i, then the nearest neighbours that belong to the majority class are removed.
- *Tomek Links (TL) [57].* Given two examples e_i and e_j belonging to different classes, with $d(e_i, e_j)$ the distance between e_i and e_j, a (e_i, e_j) pair is called a TL if there is no example e_l, so that $d(e_i, e_l) < d(e_i, e_j)$ or $d(e_j, e_l) < d(e_i, e_j)$. If two examples form a TL, then either one of these examples is noise or both examples are borderline. TL can be used as an under-sampling method or as a data cleaning method. As an under-sampling method, only examples belonging to the majority class are eliminated, and as a data cleaning method, examples of both classes are removed.
- *One-Sided Selection (OSS) [40].* This is an under-sampling method resulting from the application of TL followed by the application of the Condensed Nearest Neighbour (CNN) rule [33]. TL is used as an under-sampling method to remove noisy and borderline majority class examples. Borderline examples can be considered "unsafe" since a small amount of noise can cause them to fall on the wrong side of the decision border. CNN aims to remove examples from the majority class that are distant from the decision border. The remainder examples, i.e., "safe" majority class examples and all minority class examples, are used for learning.
- *Instance Weighting (Cost-sensitive learning) [14,56,69].* In this approach, the instances of each class are differently weighted according to the misclassification costs established. Hence, the classifier strives to make fewer errors of the more costly type, resulting in a lower overall cost. Specifically, since a cost matrix is usually not provided, it is necessary to define the costs associated with the misclassification of training examples: if one positive example is classified as a negative one, the cost of this wrong classification is the IR of the data-set; whereas if one negative example is classified as a positive one, the assigned cost is one. Obviously, an accurate classification is considered to have no cost, since in this case classifying correctly must not penalise the output model.

Regarding all these methodologies, we must discuss the differences between heuristic and non-informed techniques. The former are more sophisticated approaches which aim to perform oversampling (mostly based on the SMT) or undersampling (based on CNN) of instances taking into account the distribution of the instances within the space of the problem. Hence, these procedures try to identify the most significant examples in the borderline areas to enhance the classification of the positive class. The latter selects random examples from the training set so that the distribution of examples is set to a desired value given by the user (usually a completely balanced distribution). We must point out that despite the former techniques being developed to obtain more robust results, the quality of the latter "random" approaches is very high (in spite of their simplicity).

When considering whether it is preferable to "add" or "remove" instances from the training set, several authors have shown the advantages of the oversampling approaches over the undersampling and cleaning techniques [5,20]. This may be due to the generation of a better defined borderline between the classes on account of the addition of more minority class examples in the overlapping areas. Furthermore, since cost-sensitive learning based on instance weighting follows a similar scheme to that of

oversampling, its behaviour is expected to be competitive with this kind of techniques.

However, the advantage of undersampling techniques lies in the reduction of the training time, which is especially significant in the case of highly imbalanced data-sets with a large number of instances. Another positive feature of these approaches is that they aim to smooth the discrimination areas of the classes, which also works quite well in conjunction with the oversampling techniques, as we have previously noted, i.e., SMT + ENN.

2.3. Evaluation in imbalanced domains

In the framework of imbalanced data-sets, standard metrics such as the accuracy rate should not be considered, since they do not distinguish between the number of correct classifications of the different classes, which may lead to erroneous conclusions. Regarding this issue, our objective is to make use of a performance metric that gives the same weight to each of the classes of the problem, independently of the number of examples it has. Therefore we will use the average accuracy [22]:

$$AvgAcc = \frac{1}{C}\sum_{i=1}^{C} TPR_i \quad (1)$$

where C stands for the number of classes and TPR_i is the True Positive Rate of the ith class (noted in percentage). We must point out that the main objective of this paper is to contrast the global classification performance of the algorithms, not just to focus on the accuracy of the minority classes. According to selected evaluation criteria, more robust techniques will be preferred.

3. Solving multiple-class imbalanced data-sets

In this section we describe the different methodologies to solve multiple-class imbalanced problems. In Section 3.1, we introduce a preprocessing mechanism based on SMT, which iteratively generates new samples from the least represented class at each step, known as Static-SMT [21]. Next, in Section 3.2, we present a global cost-sensitive approach that re-weights the instances from each class according to their ratio [71]. Section 3.3 describes AdaBoost.NC, a novel boosting-based methodology for addressing multi-class imbalance problems. Finally, in Section 3.4 we define a framework based on multi-classification learning [34,48] that will allow us to apply the standard techniques for dealing with imbalanced classification.

3.1. Static-SMT

In this preprocessing mechanism [21], the resampling procedure is applied in C steps, where C stands for the number of classes of the problem. In each iteration, the resampling procedure selects the minimum size class, and duplicates the number of instances of the class in the original data-set.

Synthetic examples are obtained by applying the SMT algorithm [7] only over the instances of the minority class. To determine the amount of examples to be to generated, and to create these examples, only the instances belonging to the original data-set are taken into account when duplicating the minority class by SMT.

3.2. Global-CS

In order to equilibrate the significance of the examples for the different classes on an imbalanced framework, Zhou and Liu propose in [71] to "re-sample" each class in a consistent manner by considering a factor of N_i/N_{max}, with N_i the number of examples of the ith class and N_{max} the number of examples for the majority class of the problem.

The simplest way to achieve this end is to replicate each instance of class i $\lfloor N_{max}/N_i \rfloor$ times and to select $N_{max}\%N_i$ additional random examples from the data-set. Then, this procedure is repeated for all the classes of the problem. We should note that the majority class will not increment its size since the factor N_{max}/N_{max} is 1.

3.3. AdaBoost.NC

Ensemble techniques have shown a very strong behaviour for imbalanced problems [26]. Wang and Yao have recently developed a study regarding the extension of boosting techniques for imbalance problems with "multi-minority" and "multi-majority" classes [58].

Their approach is based on AdaBoost algorithm [23] in combination with negative correlation learning [41]. The main procedure is quite similar to any boosting approach, in which the weights of the examples are updated with an ad hoc formula depending on the classification or misclassification given by both the classifier learned in the current iteration, and the global ensemble. Initial weights in this boosting approach are assigned in inverse proportion to the number of instances in the corresponding class. For more details regarding this approach please refer to [58].

3.4. Synergy of standard approaches for imbalanced data-sets and binarization techniques

Multiple classes imply an additional difficulty for Data Mining algorithms, as the boundaries among the classes may overlap, causing a decrease in the performance level. In this situation, we can proceed by transforming the original multiple-class problem into binary subsets, which are easier to discriminate, via a class binarization technique [3,13,25].

These techniques are very useful in overcoming the gap between two-class and multiple-class imbalanced data-sets, since they make it possible to apply the standard solutions introduced in Section 2.2. Hence, these methods will be composed of two simple steps, similar to those we have already developed in our former works [17]:

1. We divide the original multi-class problem into simpler binary subproblems.
2. For each subproblem obtained, we apply those solutions that have been developed to deal with two-class imbalanced data-sets.

Specifically, there are two well-known approaches to reduce a multiple-class classification problem to a set of binary classification problems: the OVO (pairwise learning) and OVA approaches. These procedures will be described in the remainder of this section.

3.4.1. One-versus-one approach

The OVO approach [34] tries to train a classifier for each possible pair of classes, ignoring the examples that do not belong to the related classes. When classifying instances, a query is submitted to all binary models, and the predictions of these models are combined into an overall classification [36,37]. An example of this binarization technique is depicted in Fig. 1.

For those algorithms that do not have an associated certainty degree for each class, the most common way to generate the class label is to represent the output of each binary classifier in a code-matrix $\mathbb{M}$ [3]:

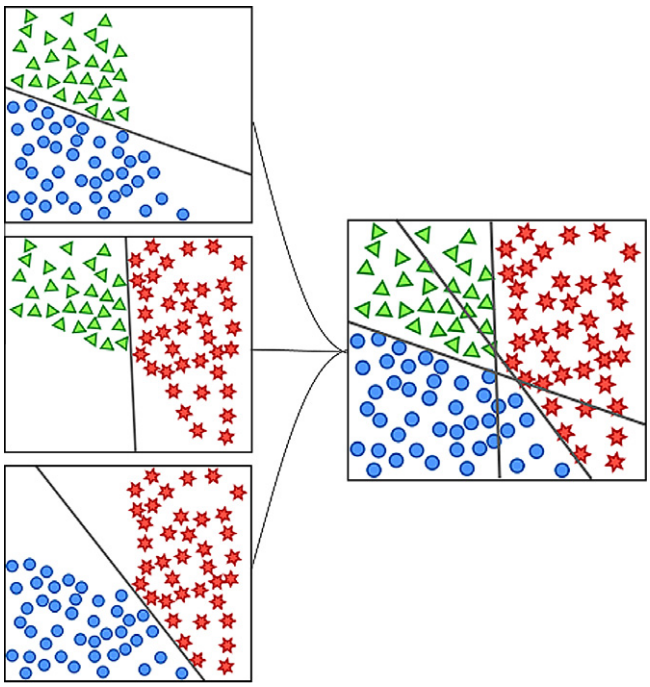

Fig. 1. Example of the OVO binarization technique for a 3-class problem.

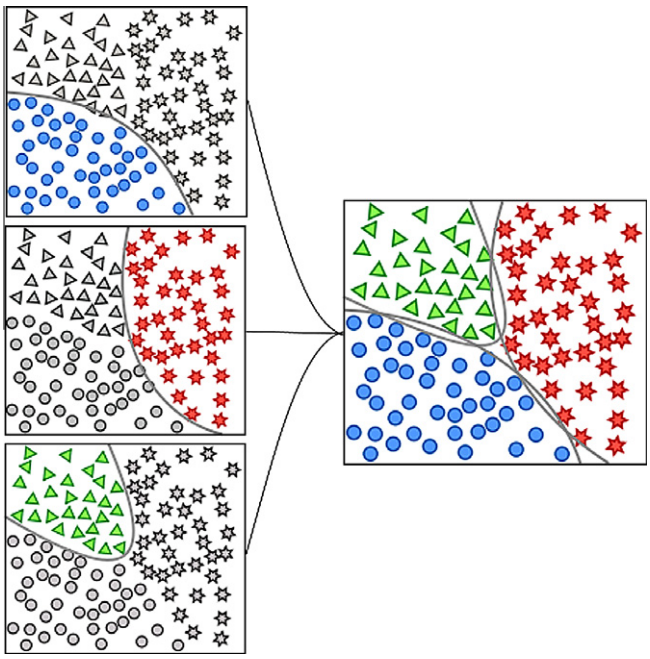

Fig. 2. Example of the OVA binarization technique for a 3-class problem.

$$\mathbb{M}(i,j) = \begin{cases} 1 & \text{if } output = i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Clearly, when $\mathbb{M}(i,j) = 1$ then $\mathbb{M}(j,i) = 0$ and vice versa. The final class is assigned by computing the maximum vote:

$$Class = \arg\max_{i=1,\ldots,C} \left\{ \sum_{j=1}^{C} \mathbb{M}_{i,j} \right\} \quad (3)$$

In cases where we have a pattern of output for which more than two classes obtain the same vote, the instance will be classified according to the maximum a priori probability, i.e., the majority class. If a vote remains tied, the class is assigned randomly from the previous possibilities.

3.4.2. One-versus-all approach

The OVA approach [48] builds a single classifier for each of the classes of the problem, considering the examples of the current class to be positives and the remaining instances negatives. An example of this binarization technique is depicted in Fig. 2.

At classification time, each model $F_1,\ldots,F_C$ will be fired in order to check the degree of membership of the query instance to its associated class (for most classifiers this value will be in $\{0,1\}$). Thus, the final decision function F for the system output can be easily made as

$$F(F_1,\ldots,F_C) = \arg\max_{i=1,\ldots,C}(F_i) \quad (4)$$

Again, in the case of a tie, the instance will be assigned to the majority class, or randomly among the majority classes if they have the same amount of examples.

4. Experimental framework

In this section we first provide details of the real-world multiple-class imbalanced problems chosen for the experiments (Section 4.1). Then we will describe the learning algorithms selected for this study and their configuration parameters (Sections 4.2 and 4.3 respectively). Next, we present the statistical tests applied to compare the results obtained with the different classifiers (Section 4.4). Finally, we introduce the information shown on the Webpage associated with the paper (Section 4.5).

4.1. Data-sets

There is no consensus in the research community on what threshold must be set up for a given data-set to suffer from the imbalance problem, either for two-class, or in multi-class problems. In this paper, and according to our previous work on the topic [20,18,19], we consider a data-set to be imbalanced when one of its classes has a distribution of examples below 40% of the number of instances that belong to the majority class; that is, if the IR is higher than 1.5.

We are aware, as we stated in Section 2.1, that the IR of a data-set is not the unique feature imposing a handicap on the classifiers in order to achieve good results for the different classes of the problem. However, the IR itself can be viewed as suggestive of a set of problems which need to be addressed in a special way. Specifically, throughout the experimental study we will observe that the results of the base classification algorithm and the simple OVO and OVA approaches will obtain a low performance in many data-sets, thus justifying its selection for the experimental framework according to the IR data characteristic.

Table 2 summarises the properties of the selected data-sets. It shows, for each data-set, the number of examples (#Ex.), the number of attributes (#Atts.), the number of classes (#Cl.) and the IR. Furthermore, we show the number of instances per class in Table 3. In the case of missing values (*autos*, *cleveland*, *dermatology* and *post-operative*) we have removed those instances from the data-set.

Estimates of the accuracy rate were obtained by means of a 10-fold cross-validation. That is, we split the data-set into 10 folds, each one containing 10% of the patterns of the data-set. For each fold, the algorithm was trained with the examples contained in the remaining folds and then tested with the current fold. We must point out that the data-set partitions employed in this paper are available for download at the KEEL data-set repository [1] so that any interested researcher can use the same data for comparison.

Table 2
Summary description of the data-sets.

id	Data-set	#Ex.	#Atts.	#Cl.	IR
Aut	Autos	159	25	6	16.00
Bal	Balance	625	4	3	5.88
Cle	Cleveland	467	13	5	12.62
Con	Contraceptive	1473	9	3	1.89
Der	Dermatology	358	33	6	5.55
Eco	Ecoli	336	7	8	71.50
Fla	Flare	1066	11	6	7.70
Gla	Glass	214	9	6	8.44
Hay	Hayes-Roth	160	4	3	2.10
Led	Led7digit	500	7	10	1.54
Lym	Lymphography	148	18	4	40.5
New	New-thyroid	215	5	3	5.00
Nur	Nursery	12690	8	5	2160.00
Pag	Page-blocks	5472	10	5	175.46
Pos	Post-operative	87	8	3	62
Sat	Satimage	6435	36	7	2.45
Shu	Shuttle	57999	9	5	4558.60
Spl	Splice	3190	60	3	2.16
Thy	Thyroid	7200	21	3	40.16
Win	Wine	178	13	3	1.48
Wqr	Wine-Quality-Red	1599	11	11	68.10
Wqw	Wine-Quality-White	4898	11	11	439.60
Yea	Yeast	1484	8	10	92.60
Zoo	Zoo	101	16	7	10.25

4.2. Algorithms selected for the study

A brief description of the three algorithms selected for our study is given in the remainder of this section. All these algorithms are available within the KEEL software tool [2].

- *C4.5 Decision Tree*. C4.5 [47] is a decision tree generating algorithm. It induces classification rules in the form of decision trees from a set of given examples. The decision tree is constructed top-down using the normalised information gain (difference in entropy) that results from choosing an attribute to split the data. The attribute with the highest normalised information gain is that which is used to make the decision.
- *Support Vector Machines*. An SVM [10] constructs a hyperplane or set of hyperplanes in a high-dimensional space. A good separation is achieved by the hyperplane that has the largest distance to the nearest training data-points of any class (the so-called functional margin), as in general, the larger the margin the lower the generalisation error of the classifier.
  In order to solve the quadratic problem that arises from SVMs, there are many techniques, mostly reliant on heuristics, for breaking the problem down into smaller, more-manageable chunks. A common method for solving the quadratic problem is Platt's Sequential Minimal Optimization algorithm [46], which breaks the problem down into 2-dimensional sub-problems that may be solved analytically, eliminating the need for a numerical optimisation algorithm [16].
- *Instance-Based Learning*. We will make use of the most common approach, kNN [42]. This is a type of instance-based learning, or lazy learning, where the function is only approximated locally and all computation is deferred until classification. The kNN algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbours, with the object being assigned to the most common class amongst its k nearest neighbours (k is a positive integer, typically small). If $k = 1$, then the object is simply assigned to the class of its nearest neighbour.
  The training phase of the algorithm only consists of storing the feature vectors and class labels of the training samples. In the classification phase, k is a user-defined constant, and an unlabeled vector (a query or test point) is classified by assigning the most frequent label from among the k training samples nearest to that query point.
  Usually Euclidean distance is used as the distance metric; however this is only applicable to continuous variables. Consequently, in this work we make use of the Heterogeneous Value Difference Metric (HVDM) [64]. This metric computes the distance between two input vectors x and y as follows:

$$\mathrm{HVDM}(x,y) = \sqrt{\sum_{a=1}^{m} d_a^2(x_a, y_a)} \tag{5}$$

where m is the number of attributes. The function $d_a(x, y)$ returns a distance between the two values x and y for attribute a and is defined as:

Table 3
Number of instances per class.

Data	Examples	C1	C2	C3	C4	C5	C6	C7	C8	C9	C10
aut	159	46	13	48	29	20	3	–	–	–	–
bal	625	49	288	288	–	–	–	–	–	–	–
cle	467	164	36	35	55	13	164	–	–	–	–
con	1473	629	333	511	–	–	–	–	–	–	–
der	358	60	111	71	48	48	20	–	–	–	–
eco	336	143	77	2	2	35	20	5	42	–	–
fla	1066	331	239	211	147	95	43	–	–	–	–
gla	214	70	76	17	13	9	29	–	–	–	–
hay	160	65	64	31	–	–	–	–	–	–	–
led	500	45	37	51	57	52	52	47	57	53	49
lym	148	61	81	4	2	–	–	–	–	–	–
new	215	150	35	30	–	–	–	–	–	–	–
nur	12690	2	4320	4266	328	4044	–	–	–	–	–
pag	5472	4913	329	87	115	28	–	–	–	–	–
pos	87	62	24	1	–	–	–	–	–	–	–
sat	6435	1358	626	707	1508	703	1533	–	–	–	–
shu	57999	8903	45586	3267	49	171	13	10	–	–	–
spl	3190	767	768	1655	–	–	–	-	–	–	–
thy	7200	6666	368	166	–	–	–	–	–	–	–
win	178	59	71	48	–	–	–	–	–	–	–
wre	1599	681	638	199	53	18	10	–	–	–	–
wwh	4898	2198	1457	880	175	163	20	5	–	–	–
yea	1484	244	429	463	44	35	51	163	30	20	5
zoo	101	41	13	10	20	8	5	4	–	–	–

$$d_a(x,y) = \begin{cases} 1, & \text{if } x \text{ or } y \text{ are unknown; otherwise...} \\ normalized\_vdm_a(x,y), & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x,y), & \text{if } a \text{ is linear} \end{cases} \quad (6)$$

The function $d_a(x, y)$ uses one of two functions (defined below), depending on whether the attribute is nominal or numerical. Note that in practice the square root in Eq. (8) is not usually performed because the distance is always positive, and the nearest neighbour(s) will still be nearest whether or not the distance is squared. Since the distance for each input variable is given in the range [0,1], distances are often normalised by dividing the distance for each variable by the range of that attribute. In the case of HVDM, the situation is more complicated because the nominal and numeric distance values come from different types of measurements: numeric distances are computed from the difference between two linear values, normalised by standard deviation, while nominal attributes are computed from a sum of C differences of probability values (where C is the number of output classes). It is therefore necessary to find a way to scale these two different kinds of measurements into approximately the same range to give each variable a similar influence on the overall distance measurement.

Since 95% of the values in a normal distribution fall within two standard deviations of the mean, the difference between numeric values is divided by four standard deviations to scale each value into a range that is usually of width 1. The function *normalized_diff* is defined as shown below (with σ_a the standard deviation of the numeric values of attribute a):

$$normalized\_diff_a(x,y) = \frac{|x-y|}{4\sigma_a} \quad (7)$$

For the function *normalized_vdm* the following formula was considered:

$$normalized\_vdm_a^2(x,y) = \sqrt{\sum_{c=1}^{C} \left| \frac{N_a,x,c}{N_a,x} - \frac{N_a,y,c}{N_a,y} \right|^2} \quad (8)$$

4.3. Parameters

Next, we detail the parameter values for the different learning algorithms selected in this study, which have been set considering the recommendations of the corresponding authors:

1. *C4.5*
   For C4.5 we have set a confidence level of 0.25, the minimum number of item-sets per leaf was set to 2 and the application of pruning was used to obtain the final tree.
2. *SVM*
   For the SVM we have chosen *Gaussian reference functions*, with an internal parameter of 0.25 for each kernel function and a penalty parameter of the error term of 100.0.
3. *kNN*
   In this case we have selected three neighbours for determining the output class, applying the HVDM as distance metric.

Regarding preprocessing techniques, the cleaning procedures employ three neighbours to determine whether an instance corresponds to noise or not. In the case of SMT and related preprocessing techniques, we will consider the *5-nearest neighbours of the minority class* to generate the synthetic samples, and *balance both classes to the 50% distribution*. In our preliminary experiments we have tried several percentages for the distribution between the classes and we have obtained the best results with a strictly balanced distribution. Finally, for AdaBoost.NC we have set up the penalty strength (λ parameter) to 2 and the number of classifiers composing the ensemble to 51, as suggested by the authors [58].

Although we acknowledge that the tuning of the parameters for each method for each particular problem could lead to better results (mainly in SVM), we preferred to maintain a baseline performance of each method as the basis for comparison. Since we are not comparing base classifiers, our hypothesis is that the methods which win on average on all problems, would also win if a better setting was performed. Furthermore, in a framework where no method is tuned, the winning methods tend to correspond to those which are most robust, which is also a desirable characteristic.

4.4. Statistical tests for performance comparison

In this paper, we use the hypothesis testing techniques to provide statistical support for the analysis of the results [28,50]. Specifically, we will use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with these types of tests [11].

We apply the Wilcoxon signed-rank test [50] as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms, as the analogous of the paired t-test. This procedure computes the differences between the performance scores of the two classifiers on ith out of N_{ds} data-sets. The differences are ranked according to their absolute values, from smallest to largest, and average ranks are assigned in the case of ties. We call R^+ the sum of ranks for the data-sets on which the second algorithm outperformed the first, and R^- the sum of ranks for the opposite. Let T be the smallest of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for N_{ds} degrees of freedom (Table B.12 in [68]), the null hypothesis of equality of means is rejected.

This statistical test allows us to know whether a hypothesis of comparison of means could be rejected at a specified level of significance α. It is also very interesting to compute the p-value associated with each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. In this manner, we can know whether two algorithms are significantly different and how different they are.

In addition, we consider the method of aligned ranks of the algorithms in order to show graphically how good a method is with respect to its partners. The first step to compute this ranking is to obtain the average performance of the algorithms in each data-set. Next, we compute the subtractions between the accuracy of each algorithm minus the average value for each data-set. Then, we rank all these differences in a descending way and, finally, we average the rankings obtained by each algorithm. In this manner, the algorithm which achieves the *lowest average ranking* is the best one.

These tests are suggested in the studies presented in [11,28,29], where its use in the field of machine learning is highly recommended. Any interested reader can find additional information on the Website http://sci2s.ugr.es/sicidm/, together with the software for applying the statistical tests.

4.5. Web page associated with the paper

In order to provide additional material to the paper content, we have developed a Web page at (http://sci2s.ugr.es/multi-imbalanced/), where we have included the following information:

- A complete description of the techniques for addressing imbalanced data-sets (presented in Section 2.2).
- A description of the classification algorithms used in this study.
- The data-sets partitions employed in the paper.

- Some Excel files with the train and test results for all the algorithms using the average accuracy, so that any interested researcher can use them to include their own results and extend the present study. Furthermore, we include the complete tables of results obtained with the mean f-measure metric. Since the conclusions extracted with both metrics are similar, using this additional metric we reinforce the lessons learned in the course of this work.

5. Experimental study

In this section, we present the empirical analysis of our methodology for multiple-class imbalanced problems. This study is divided into three parts:

1. First, in Section 5.1 we develop an analysis on the synergy of the different preprocessing approaches and the cost-sensitive learning method for both multi-classification methodologies (OVO and OVA) in order to show the best suited techniques in this context.
2. Then, we carry out an OVO versus OVA analysis in Section 5.2, using the preprocessing mechanisms and the cost-sensitive learning selected in the previous item.
3. Finally, we carry out a study to contrast the performance of the best combinations of pairwise learning and preprocessing/cost-sensitive learning with respect to the "ad hoc" methodologies for multiple-class imbalanced data-sets, i.e., Global-CS [71], Static-SMT [21] and AdaBoost.NC [58]. We also compare the results of the multi-classification techniques with those achieved by the original algorithm (only for C4.5 and kNN) and without preprocessing, in order to highlight the goodness of the combination of both techniques. This study is shown in Section 5.3.

In each section of this study we carry out a statistical analysis using the Wilcoxon non-parametrical test [50,28] between the different approaches for each one of the four selected classification paradigms. We must also point out that the complete table of results for each algorithm can be found on the associated Web-page (http://sci2s.ugr.es/multi-imbalanced/).

5.1. Analysis of the combination of preprocessing and cost-sensitive approaches with multi-classification

In this first part of the study, we want to determine whether there is a method or a set of methods for preprocessing and/or cost-sensitive learning that have a better interaction with the multi-classification schemes.

We show in Table 4 the mean results in test with the average accuracy metric (noted in percentage), together with the corresponding average rank (computed as indicated in Section 4.4), for the three algorithms, namely C4.5, SVM and kNN. This table is divided by rows into two parts, which correspond to the results using OVO and OVA schemes respectively, both with the basic approach (Std-OVO and Std-OVA) and with all the selected preprocessing techniques and cost-sensitive learning. The average rank is computed for each of these two parts separately.

From this table we observe that the best average performance and ranking mostly corresponds to oversampling and cost-sensitive learning techniques, both for OVO and OVA and independently of the classifier selected for the learning task.

The high differences regarding performance and rank values for both the oversampling approaches and cost-sensitive learning are enough to determine the robustness of the use of this type of technique. Therefore, we may select them as good and solid approaches in combination with the multi-classification scheme for multiple-class imbalanced data-sets, and they will be used as representative methods for the following sections of this experimental study.

5.2. Study of the use of OVO versus OVA

Our aim is to analyse when the cooperation between the multi-classification approach and preprocessing has a greater positive effect, whether for the OVO or OVA scheme. Please recall that the average results for the three algorithms with the different preprocessing and cost-sensitive learning approaches for OVO and OVA were shown in Table 4.

Observing this table of results, there is an average gap of two points of performance between the corresponding OVO and OVA schemes, which determine the goodness of the former. This behaviour can be found for the three learning methodologies, but it is especially evident in the case of the C4.5 algorithm.

Table 4
Average test results and rankings for the OVO and OVA schemes with and without preprocessing and cost-sensitive learning with the average accuracy metric.

Method	Adaptation	C4.5		SVM		kNN	
		Avg-Acc	Avg. rank	Avg-Acc	Avg. rank	Avg-Acc	Avg. rank
OVO	Std-OVO	69.97	–	69.14	–	67.76	–
	ROS	72.35	(4) 89.75	72.41	(3) 80.54	68.95	(4) 100.04
	SL-SMT	72.35	(2) 87.37	72.32	(4) 82.62	69.28	(3) 98.08
	SMT–ENN	70.84	(6) 116.65	71.73	(5) 89.46	67.42	(2) 97.04
	SMT	**72.74**	(**1**) **76.17**	72.58	(2) 78.04	**70.31**	(**1**) **72.75**
	CS	71.95	(3) 88.15	**72.70**	(**1**) **76.35**	68.67	(5) 106.42
	NCL	68.29	(9) 145.92	66.02	(8) 148.04	64.21	(7) 124.54
	OSS	65.92	(8) 143.19	67.96	(7) 138.50	63.30	(9) 137.17
	RUS	71.13	(5) 105.58	68.90	(6) 130.15	64.63	(6) 113.58
	TL	69.74	(7) 123.73	65.42	(9) 152.79	65.07	(8) 126.87
OVA	Std-OVA	67.00	–	66.23	–	65.28	–
	ROS	66.07	(6) 109.48	**70.47**	(**1**) **71.42**	66.62	(4) 108.02
	SL-SMT	66.91	(3) 93.48	69.72	(4) 82.83	66.63	(5) 111.08
	SMT–ENN	65.84	(5) 102.62	69.79	(5) 86.00	68.54	(2) 85.25
	SMT	**68.14**	(**1**) **70.31**	70.18	(3) 75.25	**68.89**	(**1**) **71.94**
	CS	67.29	(2) 86.94	70.39	(2) 72.81	66.62	(6) 111.17
	NCL	64.88	(7) 114.87	64.00	(9) 155.17	66.09	(7) 120.58
	OSS	61.48	(9) 153.96	63.54	(8) 151.42	64.04	(9) 134.33
	RUS	61.61	(8) 148.75	64.44	(7) 142.35	62.67	(8) 128.40
	TL	66.77	(4) 96.08	65.43	(6) 139.25	66.45	(3) 105.73

Table 5
Wilcoxon test for the comparison between OVO and OVA.

Algorithm	Preprocessing	R^+ (OVO)	R^- (OVA)	p-Value
C4.5	ROS	276.0	24.0	0.000301
	SL-SMT	275.0	25.0	0.000336
	SMT–ENN	266.0	34.0	0.000873
	SMT	264.0	36.0	0.001070
	CS	258.0	42.0	0.001935
SVM	ROS	187.0	113.0	0.283977
	SL-SMT	193.5	82.5	0.087314
	SMT–ENN	171.0	129.0	0.539027
	SMT	194.0	106.0	0.203576
	CS	205.0	95.0	0.112804
kNN	ROS	217.0	83.0	0.053784
	SL-SMT	214.0	86.0	0.065350
	SMT–ENN	154.0	146.0	0.897697
	SMT	177.0	123.0	0.432035
	CS	201.0	99.0	0.141175

Table 6
Average test results and rankings for the standard classification approaches and the OVO schemes with preprocessing and cost-sensitive learning using the average accuracy metric.

Method	Adaptation	C4.5		SVM		kNN	
		Avg-Acc	Avg. rank	Avg-Acc	Avg. rank	Avg-Acc	Avg. rank
Std	Base	71.28	(7) 139.60	–	–	62.41	(10) 173.29
	Global-CS	72.25	(6) 114.69	**73.04**	(**1**) **92.40**	67.17	(9) 133.46
	Static-SMT	70.18	(9) 148.90	70.53	(8) 132.52	68.09	(7) 125.75
	AdaBoost.NC	**74.03**	(**1**) **75.35**	71.70	(7) 108.62	69.29	(6) 122.52
OVO	Std-OVO	69.97	(10) 161.15	69.14	(9) 150.12	67.76	(8) 128.04
	ROS	72.35	(3) 107.98	72.41	(5) 99.19	68.95	(3) 107.02
	SL-SMT	72.35	(4) 110.65	72.32	(4) 99.12	69.28	(2) 102.83
	SMT–ENN	70.84	(8) 139.88	71.73	(6) 106.46	67.42	(4) 110.50
	SMT	72.74	(2) 94.98	72.58	(2) 93.02	**70.31**	(**1**) **87.42**
	CS	71.95	(5) 111.83	72.70	(3) 95.04	68.67	(5) 114.17

In order to contrast the previous findings, we carry out three different statistical analyses (Wilcoxon tests), one for each learning algorithm. This study is shown in Table 5, which is divided into three parts: the first shows the results for C4.5; the next part for SVM; and the last part for kNN. For all these tests, we compare OVO and OVA with a preprocessing mechanism, showing the sum of the ranks for the OVO approach in R^+ and for the OVA scheme in R^-.

This test concludes that the OVO classification methodology is statistically better than the OVA versions in almost all the cases of study with a high degree of confidence. In a few cases of SVM and kNN we found no significant differences, i.e., SMT–ENN, SMT and CS; in the former the sum of the ranks for the OVO scheme are quite superior, but not sufficient to obtain a degree of significance above 90%. For the latter, this could be due to the relationship between the working procedure of this algorithm and the way the oversampling is carried out in these cases; that is, both of them use the distance among the minority instances, which does not imply any changes regarding the decision procedure.

5.3. Comparative analysis for pairwise learning with preprocessing/cost-sensitive learning and "ad hoc" approaches

We aim to study the quality of the results achieved with the dual methodology "OVO + oversampling" and/or "OVO + cost-sensitive learning" versus the use of the learning algorithms in isolation. We will carry out a comparison with two different schemes:

1. The standard version of learning algorithms; i.e., the basic classification approach (for C4.5 and kNN) and the multi-classification algorithms (OVO). This step will allow us to determine the significance of the proposed combination of methodologies.
2. Solutions to the multiple-class imbalanced data-sets are not based on a binarization stage. We will contrast the results for the "OVO-oversampling" and "OVO + cost-sensitive learning" with a global cost-sensitive learning approach [71], Static-SMT [21], and AdaBoost.NC [58], as introduced in Section 3.

We have divided this analysis into three parts, each one of them regarding a different learning algorithm selected in our study. Table 6 shows the average results in test with the average accuracy, so we can observe the performance of the classification schemes, together with the average ranking of these methodologies, which will allow us to determine the robustness of each approach better than just considering the experimental results. For the sake of completeness, we show the whole table of results in test within each subsection, so that the reader can observe the performance of the different classification schemes for every single data-set.

In order to extract well-founded conclusions we will carry out a Wilcoxon test to determine whether there are significant differences between the studied approaches. The structure of these tables is as follows: the p-value is shown for the comparison between the method of the column (oversampling) with those of the rows (standard and "ad hoc" approaches). The lower the p-value, the higher the differences between both approaches.

5.3.1. C4.5. decision tree

The study for the C4.5 decision tree is shown in Table 7, where we show the average performance results. Regarding the comparison for the basic algorithm and the simple OVO scheme (see Table 8), the null hypothesis of equality against the baseline classifier and OVO approaches are rejected in the case of the combination with ROS, SL-SMT, SMT and CS. This supports the goodness of the

Table 7
Complete average accuracy test results for C4.5 with the standard and ad hoc learning algorithms and pairwise learning with preprocessing/cost-sensitive learning.

Data	Base	Std-OVO	Global-CS	Static-SMT	AdaB. NC	ROS	SL-SMT	SMT–ENN	SMT	CS
aut	80.76	76.25	**84.26**	82.04	83.16	81.42	80.53	76.25	80.11	81.31
bal	55.55	56.93	55.93	55.30	**60.84**	55.57	54.90	52.35	54.29	54.20
cle	29.24	24.61	27.65	24.91	25.29	28.95	32.46	29.06	**33.89**	27.35
con	51.72	50.08	49.83	47.19	50.14	48.05	49.82	**52.31**	50.09	49.74
der	93.48	95.65	93.56	94.83	94.79	95.71	95.37	95.65	95.61	**96.33**
eco	70.72	59.69	66.28	65.15	**74.79**	72.89	71.48	70.97	70.99	73.65
fla	59.24	59.09	64.20	64.01	58.64	64.39	62.97	59.09	**64.74**	63.72
gla	63.71	65.45	70.95	63.71	**73.97**	68.81	64.76	70.58	70.84	65.44
hay	83.49	83.49	83.49	86.03	**88.17**	82.86	83.49	70.08	83.49	83.49
led	71.40	70.52	69.43	**72.55**	62.78	71.59	71.34	71.32	71.64	70.72
lym	67.67	61.28	69.27	67.81	66.49	**72.51**	62.86	61.95	60.91	70.77
new	91.39	92.28	91.67	90.56	**95.11**	90.11	91.44	90.33	92.50	91.44
nur	88.30	88.45	93.51	87.76	93.45	93.33	93.69	92.79	**94.07**	93.58
pag	84.53	83.80	88.28	85.55	90.59	**91.52**	90.87	89.88	90.24	90.69
pos	46.62	48.33	38.90	21.34	40.13	37.13	**48.89**	47.78	48.33	31.92
sat	83.19	83.70	83.73	83.19	**88.78**	84.28	84.02	84.74	84.18	84.08
shu	92.19	97.98	**98.55**	95.05	98.47	96.69	96.69	94.70	96.84	96.79
spl	94.11	94.90	94.11	94.33	93.41	94.90	94.81	94.61	94.88	**94.92**
thy	98.65	98.32	98.94	98.65	99.24	99.02	98.94	98.28	**99.28**	98.93
win	94.98	91.24	94.32	94.24	**96.46**	91.24	92.27	87.75	92.02	91.35
wqr	31.87	27.05	34.01	31.87	**39.48**	33.30	34.25	32.57	34.01	35.84
wqw	38.78	32.32	38.78	38.78	**50.04**	41.41	41.20	36.64	42.35	40.49
yea	50.24	48.18	47.66	51.77	**55.94**	50.72	50.98	50.70	52.10	52.30
zoo	88.83	89.73	96.69	87.64	**96.69**	90.11	88.45	89.73	88.45	87.73
avg	71.28	69.97	72.25	70.18	**74.03**	72.35	72.35	70.84	72.74	71.95

Table 8
Wilcoxon test: *p*-values obtained for OVO + preprocessing/CS versus basic approaches for multiple-class learning (C4.5).

	ROS	SL-SMT	SMT–ENN	SMT	CS
Base	0.0333	0.0297	0.7642	0.0231	0.0839
Std-OVO	0.0254	0.0156	1.0000	0.0075	0.0411
Global-CS	0.7642	0.7495	1.0000	0.2246	1.0000
Static-SMT	0.0114	0.0789	0.8303	0.0357	0.0158
AdaB.NC	1.0000	1.0000	1.0000	1.0000	1.0000

application of oversampling and cost-sensitive learning in order to achieve a higher precision in all the classes of the problem.

In the analysis versus Global-CS, Static-SMT and AdaBoost.NC, we may observe three different behaviours depending upon the methodology:

1. First, AdaBoost.NC excels as the best approach overall in terms of the quality of its obtained results. This behaviour was expected a priori, since it is known that ensemble methodologies with "weak learners", such as decision trees, are designed to maximise the accuracy of the base classifiers by focusing on difficult examples. Additionally, it uses much many classifiers than the remaining approaches, 51 decision trees in total.
2. Second, Static-SMT is clearly outperformed by ROS, SL-SMT, SMT and CS, which were shown to be the best methodologies on average for the pairwise learning scheme.
3. Finally, the Global-CS approach achieves a high performance and the statistical pairwise comparison shows no differences in any case; however we must highlight the comparison between OVO + SMT and Global-CS, in which the former obtains a *p*-value closest to the threshold that statistically determines a superior behaviour.

5.3.2. Support vector machines

We can observe the complete test results for the SVM algorithm in Table 9. The trend in this case is quite similar to that of C4.5 where, despite AdaBoost.NC dropping in performance, the combination of OVO + preprocessing, and the Global-CS approach still achieve the highest global performance from among the remaining techniques. Additionally, we must highlight the high quality of the average results in contrast to those obtained by C4.5.

The statistical study for SVM is developed in Table 10. The conclusions regarding the comparison between the standard OVO approach and its combination with oversampling/cost-sensitive learning are similar to those extracted in the case of the C4.5 algorithm, in which the latter allows the achievement of enhanced results with respect to the standard OVO approach in the scenario of multiple-class imbalanced data-sets.

According to the analysis of the Global-CS methodology, we observe that its performance is truly competitive with that obtained by OVO and preprocessing, mainly due to the features of the SVM approach, which internally applies a binarization step, causing both methodologies to share a similar behaviour. However, Static-SMT is outperformed by SMT and CS, and a low *p*-value is also achieved for ROS and SMT–ENN, suggesting that its synergy with pairwise learning is not positive in this case. Finally, as stated at the beginning of this section, AdaBoost.NC shows a decrease in its performance with respect to C4.5. This could be due to the fact that ensemble methodologies do not work well with "strong classifiers" such as SVMs and therefore the expected improvement of the results is not achieved in this case. However, it obtains the highest accuracy for several data-sets, showing a good behaviour for these problems in comparison with the remaining approaches.

5.3.3. k-Nearest neighbour

Finally, the results for the kNN algorithm are shown in Table 11. The average performance in this case is somewhat lower than those obtained for C4.5 and SVM, but the extracted conclusions are equivalent. Again, the synergy between OVO and oversampling/cost-sensitive learning shows a very positive behaviour, especially in the case of SMT and SL. Surprisingly, the Global-CS approach, from which we observed a very robust performance in the two previous classification paradigms, shows a decrease in the quality of its results.

The statistical study based on the Wilcoxon test (Table 12) determines the goodness of the SMT and SL-SMT approaches, following similar findings observed in the previous analysis for C4.5

Table 9
Complete average accuracy test results for SVM with the standard and ad hoc learning algorithms and pairwise learning with preprocessing/cost-sensitive learning.

Data	Std-OVO	Global-CS	Static-SMT	AdaB.NC	ROS	SL- SMT	SMT–ENN	SMT	CS
aut	74.81	76.87	74.58	69.70	75.31	77.31	77.49	77.49	**77.88**
bal	91.08	**91.63**	**91.63**	90.64	**91.63**	**91.63**	**91.63**	**91.63**	**91.63**
cle	33.62	34.38	31.74	30.06	35.61	35.97	33.65	34.60	**36.88**
con	48.10	51.66	49.01	**53.18**	50.95	51.40	50.95	51.72	50.48
der	95.82	95.78	95.60	**97.08**	95.93	94.30	95.93	95.78	95.44
eco	70.12	67.95	70.03	65.17	69.37	68.96	**70.59**	68.21	68.19
fla	61.47	63.45	64.21	63.29	**64.91**	64.06	63.63	63.63	64.23
gla	58.83	64.72	58.31	55.62	62.42	**68.02**	61.69	63.95	67.91
hay	56.19	57.78	**64.29**	57.22	58.41	58.89	54.05	55.00	56.83
led	73.68	72.79	73.17	72.77	**73.73**	73.28	73.31	73.31	73.53
lym	72.74	82.60	82.74	82.04	**82.81**	74.13	70.33	70.79	82.39
new	95.17	96.89	95.78	92.67	94.67	96.89	95.56	**97.11**	96.89
nur	99.39	97.83	95.25	**99.84**	97.77	99.83	78.73	99.82	97.77
pag	63.94	**91.67**	69.04	88.29	89.09	89.34	87.93	88.47	89.32
pos	49.63	35.45	**50.75**	44.20	34.82	33.75	40.90	34.83	37.12
sat	80.71	84.81	80.77	**87.72**	84.50	84.58	84.63	84.54	84.47
shu	65.27	**92.68**	63.70	83.87	84.25	84.51	84.17	84.39	84.14
spl	88.18	79.32	95.31	**96.14**	80.25	80.16	95.26	94.67	79.75
thy	79.64	92.60	81.52	**93.03**	91.67	92.22	89.89	90.85	92.04
win	**97.77**	**97.77**	97.22	95.98	97.22	97.22	97.68	97.22	**97.77**
wqr	28.83	39.33	30.74	37.22	39.74	37.93	**40.92**	38.34	37.82
wqw	25.55	**34.56**	27.53	14.67	34.09	33.29	33.30	33.82	33.41
yea	54.66	55.49	54.45	55.39	55.69	55.08	56.22	**56.74**	55.91
zoo	94.07	95.02	**95.35**	95.02	93.02	93.02	93.02	95.02	93.02
avg	69.14	**73.04**	70.53	71.70	72.41	72.32	71.73	72.58	72.70

Table 10
Wilcoxon test: *p*-values obtained for OVO + preprocessing/CS versus basic approaches for multiple-class learning (SVM).

	ROS	SL-SMT	SMT–ENN	SMT	CS
Std-OVO	0.0293	0.0184	0.0383	0.0082	0.0184
Global-CS	1.0000	1.0000	1.0000	1.0000	1.0000
Static-SMT	0.2113	0.4651	0.2296	0.0946	0.0839
AdaB.NC	0.3897	0.5203	0.8081	0.5531	0.4490

and SVM; that is, outperforming both the base and the standard OVO methodology. As stated previously, we observe that significant differences are found for SMT, SL-SMT and even ROS with respect to Global-CS. Otherwise, regarding the comparison with Static-SMT, this methodology obtains a more robust behaviour in conjunction with the kNN algorithm. Finally, the analysis for AdaBoost.NC is identical to that carried out for SVM, in which this approach obtains a good average performance but, in the contrary case of C4.5, now the pairwise methodology with SMT is shown to be a better procedure for addressing this type of problem.

6. Lessons learned and future work

This paper has provided an empirical analysis of several methods for dealing with multiple-class imbalanced problems, most of them based on the combination of binary approaches and OVO and OVA strategies, completed with other ad hoc methods designed for this problem. We structured the analysis in three sections, developing a scalable study that determined, step by step, the most representative solutions and finally carried out a global comparison. From this study we can emphasise seven important lessons learned:

a. Regarding the synergy of preprocessing and binarization techniques, the oversampling methodologies have shown a more robust behaviour than those based on undersampling and cleaning procedures for multiple-class imbalanced problems. In the case of the former, this could be due to the fact that many data-sets have some classes with a very low number of examples and thus, equalizing the distribution of classes implies the removal of many instances that may have relevant information in order to determine the classification boundary. Regarding cleaning techniques, they behave similarly, so that a small quantity of examples for some minority classes is not enough to determine those majority instances which contribute with noise to bias the classification.
b. Furthermore, considering the OVO versus OVA comparison, OVO methods in general have shown better behaviour, especially according to the average performance obtained. The reason behind this higher quality of results is primarily that the pairwise learning technique confronts a lower subset of instances and is therefore less likely to obtain imbalanced training-sets, which is the disadvantage in this case.
Additionally, we must be aware that in this case the decision boundaries of each binary problem may be considerably simpler than the OVA strategy. Finally, OVO was shown to be more accurate for rule learning algorithms (C4.5), a finding in accordance with previous studies [24].
c. We have determined that a positive synergy is achieved with the combination of the standard solutions for binary imbalanced problems and the ensemble-based "divide-and-conquer" techniques. This "dual-methodology" outperforms the basic and multi-classifier approaches, thus highlighting the significance of the application of the standard solutions to deal with classification with imbalanced data-sets.
d. We must stress that the best of techniques studied are those based on OVO with SMT and OVO with cost-sensitive learning. First, because of their overall performance, and also because higher differences were found when comparing these techniques with the remaining methodologies during the statistical analysis. On the other hand, among the oversampling techniques, we have observed that, in spite of its simplicity, Random-Oversampling achieved good results in comparison with the rest and with the more sophisticated approaches.

Table 11
Complete average accuracy test results for kNN with the standard and ad hoc learning algorithms and pairwise learning with preprocessing/cost-sensitive learning.

Data	Base	Std-OVO	Global-CS	Static-SMT	AdaB. NC	ROS	SL-SMT	SMT–ENN	SMT	CS
aut	55.62	70.78	75.71	**77.88**	70.49	76.22	76.72	75.78	75.78	77.78
bal	60.28	60.86	56.29	55.67	49.46	54.14	54.75	**61.40**	56.15	53.70
cle	26.56	34.04	30.64	29.72	29.49	**36.75**	33.44	32.51	32.68	34.74
con	42.24	43.46	42.58	42.58	45.53	44.24	44.66	**47.52**	44.44	44.32
der	96.94	96.94	94.86	95.13	95.22	96.82	95.92	**97.13**	96.49	95.93
eco	72.29	72.40	71.79	70.53	70.43	73.54	73.85	72.75	**74.38**	72.70
fla	48.32	60.78	56.46	57.30	**64.02**	62.67	59.84	33.19	60.54	60.69
gla	66.11	69.96	71.73	74.16	69.19	73.87	**75.02**	70.60	71.52	74.23
hay	24.80	68.29	48.06	49.40	61.83	73.29	**79.48**	44.80	72.82	77.82
led	45.38	22.91	42.20	43.21	**72.97**	21.78	19.06	38.41	30.71	20.43
lym	68.44	73.50	77.88	**83.99**	81.21	73.02	75.10	72.81	74.68	72.81
new	88.78	91.17	95.17	**96.50**	91.83	94.28	95.39	94.00	96.00	95.39
nur	82.07	94.10	93.25	**97.01**	92.94	94.90	94.94	73.39	95.21	94.79
pag	72.75	81.71	83.93	84.97	84.63	85.38	86.14	**92.65**	92.51	86.20
pos	40.98	45.31	39.87	40.06	30.93	43.01	**46.42**	40.05	38.91	34.70
sat	89.35	89.64	89.58	89.66	87.27	90.25	90.12	90.21	**90.29**	90.06
shu	91.15	86.66	91.02	92.71	**96.13**	89.73	89.73	91.58	92.67	89.73
spl	77.50	**95.36**	93.70	89.43	80.24	95.00	94.82	94.08	94.97	94.67
thy	58.14	78.52	62.86	69.14	67.34	80.27	80.01	**86.91**	85.72	80.10
win	96.06	96.73	**98.10**	97.14	96.06	96.25	96.25	95.30	96.25	95.30
wqr	25.99	26.65	26.57	27.37	36.35	29.27	29.50	**36.71**	36.28	29.10
wqw	28.15	27.31	29.90	30.13	**42.05**	32.15	32.67	37.91	37.74	32.70
yea	51.13	51.36	50.45	51.22	**54.45**	50.17	50.53	50.15	52.45	51.91
zoo	88.83	87.88	89.52	89.29	**93.00**	87.88	88.36	88.36	88.36	88.36
avg	62.41	67.76	67.17	68.09	69.29	68.95	69.28	67.43	**70.31**	68.67

Table 12
Wilcoxon test: *p*-values obtained for OVO + preprocessing/CS versus basic approaches for multiple-class learning (kNN).

	ROS	SL-SMT	SMT–ENN	SMT	CS
Base	0.0058	0.0045	0.0288	0.0018	0.0170
Std-OVO	0.0235	0.0288	0.5581	0.0069	0.1491
Global-CS	0.0696	0.0288	0.6373	0.0184	0.2246
Static-SMT	0.3531	0.2246	0.8751	0.1491	0.6171
AdaB.NC	0.4155	0.5203	1.0000	0.1491	0.5581

e. The global cost-sensitive approach [71] and the AdaBoost.NC ensemble have been shown to be competitive with the aforementioned techniques. The former achieved a superior performance when applied with SVMs and the latter in the case of the C4.5 decision tree. We must also state that for the sake of focusing on the minority classes, it is mandatory for AdaBoost.NC to initialise the weights of the examples with respect to the number of examples per class.
f. Regarding the comparison between OVO plus strategies for imbalanced classification, and ad hoc approaches, we must stress several advantages that make the use of the former preferable, such as efficiency, simplicity in the adaptation of existing classification approaches, and the possibility of combining them with new and more sophisticated techniques for addressing data imbalance.
g. Finally and not least importantly, the behaviour of the studied methodologies has been practically identical independently of the classifier used. In other words, the conclusions highlighted throughout the experimental study were basically the same in all cases, thus providing a higher support for the findings summarised here.

We have identified throughout this paper that binarization techniques with an appropriate preprocessing or cost-sensitive strategy are simple but useful mechanisms to improve classifiers' performance in imbalanced domains, but still there is still future work that remain to be addressed regarding this topic:

a. *Non-competetent examples in OVO strategy (as stated in [37])*: not all classifiers are trained with all the instances in the data-set, but in testing phase, the new instance is submitted to all classifiers. The classifiers which have not been trained with the instance from the class of the new example will make a prediction that will probably negatively affect the final results, as these classifiers are not competent. This is a crucial issue for imbalanced data-sets since misclassifications induce wrong weighting values in the score matrix, implying a higher cost according to the evaluations metrics applied in this scenario.
b. *Scalability*: two main challenges need to be studied; the adaptation to data-sets with a large number of classes should be considered, since the learning of the decision boundaries and their combinations can be directly affected by this issue. In addition, the number of examples composing each one of those classes must be also taken into account.
c. *The OVO strategy as a decision making problem*: new aggregations with which to combine the score matrix from OVO classifiers must be developed, aiming with the unclassifiable region when standard voting is used. Additionally, new trends of study can be oriented to dealing with the imbalance degree at the final decision step rather than simply in the learning phase with both preprocessing and instance weighting.
d. *Intrinsic data characteristics:* we must stress the significant effect of the IR on the classifiers' performance, but we are aware that there other data intrinsic characteristics that can be taken into account such as small sample size, small disjuncts, class overlapping and dataset shift. Overcoming these problems in conjunction with the pairwise learning scheme could be key to developing new approaches that improve the correct identification of the different minority and majority classes of the problem.

7. Concluding remarks

We have presented a complete experimental study for the classification of multiple-class imbalanced data-sets with the aim of

laying the basis for the achievement of high quality solutions. We have contrasted the use of the combination of binarization techniques with both preprocessing of instances and cost-sensitive learning with several ad hoc approaches such as an instance weighting cost-sensitive learning and an SMT based preprocessing technique developed for multiple classes.

We have tested the quality of these approaches using three algorithms based on different paradigms; namely, decision trees, SVMs and instance-based learning. The experimental results obtained in this study, supported by the corresponding statistical procedure, allow us to stress the good behaviour achieved by the synergy between pairwise learning and oversampling/cost-sensitive learning, which obtained the best global results for all the classification algorithms used in this study. We must also stress the robustness of global instance weighting based on cost-sensitive learning and AdaBoost.NC ensemble approaches, both of which have been shown to be competitive with respect to the combination of OVO and oversampling/cost-sensitive learning in terms of average performance.

Finally, we must emphasise that this work provides the basis for the achievement of high quality solutions for imbalanced data-sets with multiple classes, but its significance lies also in the fact that it opens future trends of research, as discussed in depth in the paper.

Acknowledgments

This work was partially supported by the Spanish Ministry of Science and Technology under Projects TIN2008-06681-C06-02, TIN2011-28488 and the Andalusian Research Plans P11-TIC-7765, P10-TIC-6858 and TIC-3928. V. López holds a FPU scholarship from Spanish Ministry of Education.

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Multple-Valued Logic Soft Comput. 17 (2–3) (2011) 255–287.
[2] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms to data mining problems, Soft Comput. 13 (3) (2009) 307–318.
[3] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.
[4] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recogn. 36 (3) (2003) 849–851.
[5] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, SIGKDD Explor. Newsl. 6 (1) (2004) 20–29.
[6] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: safe-level-synthetic minority over-sampling TEchnique for handling the class imbalanced problem, in: PAKDD'09, Lecture Notes in Computer Science, vol. 5476, Springer, 2009.
[7] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, Smote: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[8] N.V. Chawla, N. Japkowicz, A. Kolcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explor. Newsl. 6 (1) (2004) 1–6.
[9] D.A. Cieslak, T.R. Hoens, N.V. Chawla, W.P. Kegelmeyer, Hellinger distance decision trees are robust and skew-insensitive, Data Min. Knowl. Discov. 24 (1) (2012) 136–158.
[10] C. Cortes, V. Vapnik, Support vector networks, Mach. Learn. 20 (1995) 273–297.
[11] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[12] C. Diamantini, D. Potena, Bayes vector quantizer for class-imbalance problem, IEEE Trans. Knowl. Data Eng. 21 (5) (2009) 638–651.
[13] T.G. Dietterich, An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization, Mach. Learn. 40 (2000) 139–157.
[14] P. Domingos, Metacost: a general method for making classifiers cost sensitive, in: Fifth International Conference on Knowledge Discovery and Data Mining (KDD'99), 1999.
[15] A. Estabrooks, T. Jo, N. Japkowicz, A multiple resampling method for learning from imbalanced data sets, Comput. Intell. 20 (1) (2004) 18–36.
[16] R.-E. Fan, P.-H. Chen, C.-J. Lin, Working set selection using the second order information for training SVM, J. Mach. Learn. Res. 6 (2005) 1889–1918.
[17] A. Fernández, M. del Jesus, F. Herrera, Multi-class imbalanced data-sets with linguistic fuzzy rule based classification systems based on pairwise learning, in: IPMU'2010, LNAI, vol. 6178, 2010.
[18] A. Fernández, M.J. del Jesus, F. Herrera, Hierarchical fuzzy rule based classification systems with genetic rule selection for imbalanced data-sets, Int. J. Approx. Reason. 50 (3) (2009) 561–577.
[19] A. Fernández, M.J. del Jesus, F. Herrera, On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets, Inform. Sci. 180 (8) (2010) 1268–1291.
[20] A. Fernández, S. García, M.J. del Jesus, F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, Fuzzy Set. Syst. 159 (18) (2008) 2378–2398.
[21] F. Fernández-Navarro, C. Hervás-Martínez, P.A. Gutiérrez, A dynamic over-sampling procedure based on sensitivity for multi-class problems, Pattern Recogn. 44 (2011) 1821–1833.
[22] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recogn. Lett. 30 (2009) 27–38.
[23] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: 13th. Int. Conf. Mach. Learn. (ICML'96), 1996.
[24] J. Fürnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.
[25] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recogn. 44 (8) (2011) 1761–1776.
[26] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for class imbalance problem: Bagging, boosting and hybrid based approaches, IEEE Trans. Syst., Man, Cybern. C 42 (4) (2012) 463–484.
[27] S. García, J. Derrac, I. Triguero, C.J. Carmona, F. Herrera, Evolutionary-based selection of generalized instances for imbalanced classification, Knowl. Based Syst. 25 (1) (2012) 3–12.
[28] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Comput. 13 (10) (2009) 959–977.
[29] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[30] V. García, J. Sánchez, R. Mollineda, On the effectiveness of preprocessing methods when dealing with different levels of class imbalance, Knowl. Based Syst. 25 (1) (2012) 13–21.
[31] N. García-Pedrajas, J. Pérez-Rodríguez, M. García-Pedrajas, D. Ortiz-Boyer, C. Fyfe, Class imbalance methods for translation initiation site recognition in DNA sequences, Knowl. Based Syst. 25 (1) (2012) 22–34.
[32] G. Giacinto, R. Perdisci, M.D. Rio, F. Roli, Intrusion detection in computer networks by a modular ensemble of one-class classifiers, Inform. Fusion 9 (1) (2008) 69–82.
[33] P. Hart, The condensed nearest neighbor rule, IEEE Trans. Inform. Theory 14 (1968) 515–516.
[34] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Statist. 26 (2) (1998) 451–471.
[35] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[36] E. Hüllermeier, K. Brinker, Learning valued preference structures for solving classification problems, Fuzzy Set. Syst. 159 (18) (2008) 2337–2352.
[37] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recogn. 43 (1) (2010) 128–142.
[38] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intell. Data Anal. 6 (5) (2002) 429–450.
[39] T.M. Khoshgoftaar, J.V. Hulse, A. Napolitano, Comparing boosting and bagging techniques with noisy and imbalanced data, IEEE Trans. Syst., Man, Cybern. A 41 (3) (2011) 552–568.
[40] M. Kubat, S. Matwin, Addressing the curse of imbalanced training sets: one-sided selection, in: International Conference on Machine Learning, 1997.
[41] Y. Liu, X. Yao, Simultaneous training of negatively correlated neural networks in an ensemble, IEEE Trans. Syst., Man, Cybern. B 29 (6) (1999) 716–725.
[42] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley and Sons, 2004.
[43] J.G. Moreno-Torres, F. Herrera, A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction, in: 10th International Conference on Intelligent Systems Design and Applications(ISDA2010), 2010.
[44] A. Orriols-Puig, E. Bernadó-Mansilla, Evolutionary rule-based systems for imbalanced datasets, Soft Comput. 13 (3) (2009) 213–225.
[45] A. Orriols-Puig, E. Bernadó-Mansilla, D.E. Goldberg, K. Sastry, P.L. Lanzi, Facetwise analysis of XCS for problems with class imbalances, IEEE Trans. Evol. Comput. 13 (2009) 260–283.
[46] J. Platt, Fast training of support vector machines using sequential minimal optimization, in: Advances in Kernel Methods – Support Vector Learning, MIT Press, Cambridge, MA, 1998.
[47] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufman Publishers, San Mateo-California, 1993.
[48] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[49] C. Seiffert, T.M. Khoshgoftaar, J.V. Hulse, A. Folleco, An empirical study of the classification performance of learners on imbalanced and noisy software quality data, Inform. Sci., in press, doi: 10.1016/j.ins.2010.12.016.

[50] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, second ed., Chapman & Hall, CRC, 2006.
[51] P. Soda, A multi-objective optimisation approach for class imbalance learning, Pattern Recogn. 44 (8) (2011) 1801–1810.
[52] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recogn. 40 (2007) 3358–3378.
[53] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recogn. 23 (4) (2009) 687–719.
[54] M.A. Tahira, J. Kittlera, F. Yan, Inverse random under sampling for class imbalance problem and its application to multi-label classification, Pattern Recogn. 45 (10) (2012) 3738–3750.
[55] Y. Tang, Y.-Q. Zhang, N.V. Chawla, SVMs modeling for highly imbalanced classification, IEEE Trans. Syst., Man, Cybern. B 39 (1) (2009) 281–288.
[56] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Trans. Knowl. Data Eng. 14 (3) (2002) 659–665.
[57] I. Tomek, Two modifications of CNN, IEEE Trans. Syst. Man Commun. 6 (1976) 769–772.
[58] S. Wang, X. Yao, Multiclass imbalance problems: analysis and potential solutions, IEEE Trans. Syst., Man, Cybern. B 42 (4) (2012) 1119–1130.
[59] M. Wasikowski, X.-W. Chen, Combating the small sample class imbalance problem using feature selection, IEEE Trans. Knowl. Data Eng. 22 (10) (2010) 1388–1400.
[60] G. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, J. Artif. Intell. Res. 19 (2003) 315–354.
[61] G.M. Weiss, Mining with rarity: a unifying framework, SIGKDD Explor. Newsl. 6 (1) (2004) 7–19.
[62] G.M. Weiss, Y. Tian, Maximizing classifier utility when there are data acquisition and modeling costs, Data Min. Knowl. Discov. 17 (2) (2008) 253–282.
[63] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Trans. Syst. Man Commun. 2 (3) (1972) 408–421.
[64] D. Wilson, T. Martinez, Improved heterogeneous distance functions, J. Artif. Intell. Res. 6 (1997) 1–34.
[65] Q. Yang, X. Wu, 10 Challenging problems in data mining research, Int. J. Inform. Tech. Decis. 5 (4) (2006) 597–604.
[66] B. Zadrozny, C. Elkan, 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), 2001.
[67] B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03), 2003.
[68] J.H. Zar, Biostatistical Analysis, Prentice Hall, Upper Saddle River, New Jersey, 1999.
[69] H. Zhao, Instance weighting versus threshold adjusting for cost-sensitive classification, Knowl. Inform. Syst. 15 (3) (2008) 321–334.
[70] X.-M. Zhao, X. Li, L. Chen, K. Aihara, Protein classification with imbalanced data, Proteins 70 (2008) 1125–1132.
[71] Z.-H. Zhou, X.-Y. Liu, On multi-class cost-sensitive learning, Comput. Intell. 26 (3) (2010) 232–257.