Information Sciences 354 (2016) 178–196

Contents lists available at ScienceDirect

Information Sciences

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

Ordering-based pruning for improving the performance of ensembles of classifiers in the framework of imbalanced datasets

Mikel Galar[a], Alberto Fernández[b,*], Edurne Barrenechea[a,c], Humberto Bustince[a,c], Francisco Herrera[d,e]

[a] Departamento de Automática y Computación, Universidad Pública de Navarra, Pamplona, Spain
[b] Department of Computer Science, University of Jaén, Jaén, Spain
[c] Institute of Smart Cities (ISC), Universidad Pública de Navarra, Pamplona, Spain
[d] Department of Computer Science and Artificial Intelligence, University of Granada, Granada, Spain
[e] Faculty of Computing and Information Technology - North Jeddah, King Abdulaziz University (KAU), Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 23 May 2015
Revised 17 February 2016
Accepted 25 February 2016
Available online 18 March 2016

Keywords:
Imbalanced datasets
Tree-based ensembles
Ordering-based pruning
Bagging
Boosting

ABSTRACT

The scenario of classification with imbalanced datasets has gained a notorious significance in the last years. This is due to the fact that a large number of problems where classes are highly skewed may be found, affecting the global performance of the system. A great number of approaches have been developed to address this problem. These techniques have been traditionally proposed under three different perspectives: data treatment, adaptation of algorithms, and cost-sensitive learning.

Ensemble-based models for classifiers are an extension over the former solutions. They consider a pool of classifiers, and they can in turn integrate any of these proposals. The quality and performance of this type of methodology over baseline solutions have been shown in several studies of the specialized literature.

The goal of this work is to improve the capabilities of tree-based ensemble-based solutions that were specifically designed for imbalanced classification, focusing on the best behaving bagging- and boosting-based ensembles in this scenario. In order to do so, this paper proposes several new metrics for ordering-based pruning, which are properly adapted to address the skewed-class distribution. From our experimental study we show two main results: on the one hand, the use of the new metrics allows pruning to become a very successful approach in this scenario; on the other hand, the behavior of Under-Bagging model excels, achieving the highest gain with the usage of pruning, since the random undersampled sets that best complement each other can be selected. Accordingly, this scheme is capable of outperforming previous ensemble models selected from the state-of-the-art.

* Corresponding author. Tel.: +34 948166048; fax: +34 948168924.
E-mail addresses: mikel.galar@unavarra.es (M. Galar), ahilario@ujaen.es, alberto.fernandez@ujaen.es (A. Fernández), edurne.barrenechea@unavarra.es (E. Barrenechea), bustince@unavarra.es (H. Bustince), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.ins.2016.02.056
0020-0255/

1. Introduction

When working with classification tasks, it may be observed that datasets frequently present a very different distribution of examples within their classes. This issue is known as the problem of imbalanced classes [30,66], and it has been addressed throughout the last ten years [13]. Even so, the development of algorithms for learning classifiers in this scenario is still a hot topic of research [46,60]. This is mainly due to the high number of real applications that are affected by this condition. Enumerating some examples we may refer to bankruptcy prediction [37], medical data analysis [9,38] and bioinformatics [7,29], among others.

The presence of classes with few data can generate sub-optimal classification models, since there is a bias towards the majority class. This is due to the fact that, when the standard accuracy metric is considered, predicting the class with a higher number of examples is preferred during the learning process; therefore, the discrimination functions computed by the algorithm will be positively weighted towards the majority class [46]. Hence, there is an undeniable need for developing more precise approaches in order to reach the maximum precision in every class, independently of its representation or distribution. Furthermore, recent studies have shown that additional data intrinsic characteristics have a strong influence on the correct identification of the minority class examples [46,64].

Traditionally, solutions for this problem have been divided into three large groups [46,48], i.e. preprocessing [4] (to balance the example distribution per class), ad-hoc adaptation of standard algorithms [78], and the usage of cost-sensitive learning [19]. Any of the former approaches can be integrated into an ensemble-type classifier, thus empowering the achieved performance, as it has been shown in the specialized literature [22,23,46,65].

In summary, an ensemble is a collection of classifiers aimed at increasing the generalization capability of a single classifier, since classifiers in the ensemble are supposed to complement each other [59,62,75]. These classifiers are then jointly applied in order to obtain a single solution in agreement. Reader might guess that the more elements the ensemble has, the more reliable the solution will be, but there is a limit from which the accuracy does not improve or even worse, it could be degraded [83]. There are two main reasons for this behavior: (1) the difficulty in the decision process regarding possible contradictions or even redundancy among the components of the ensemble; and (2) the overfitting problem when adjusting the weights in a boosting-based ensemble.

In accordance with these issues, several proposals have been developed to carry out a selection of classifiers within the ensemble [5,82], which are named as *pruning* methods. The goal is to obtain a subset of the ensemble that solves the classification problem in an optimal way, i.e., maintaining or improving the accuracy of the system. In this paper we focus on ordering-based pruning, whose working procedure is based on a greedy approach and whose effectiveness in standard classification has been already proved [31,51]. This scheme starts from a trained ensemble composed of a large number of classifiers. Then, classifiers are iteratively selected one by one from the pool according to the maximization of a given metric and added to the final ensemble. This process is usually carried out until a pre-established number of classifiers are selected.

The heuristic metrics used for the ensemble pruning methodology were originally defined for standard classification tasks. In the scenario of imbalanced datasets, the effect of each classifier in the recognition of both classes must be analyzed in detail in order to obtain valid results. Therefore, ordering-based pruning metrics must be adapted this specific scenario, taking the data representation into account. Our objective is to focus on the class imbalance of the problem during the whole learning process. First, in the ensemble learning stage, via the use of those learning methods inherently adapted to this context [23]. Second, a posteriori, that is, in the classifier pruning step by selecting the most appropriate classifiers with our novel proposed metrics. As we will show in the experimental study, this positive synergy will allow us to boost the final performance of the system.

Specifically, the contributions of this paper can be summarized as follows:

- To use the ensemble pruning methodology in the context of imbalanced classification for improving the behavior of ensemble-based solutions in this framework.
- To develop novel ordering-based pruning metrics taking the properties of the class imbalance problem into account. In particular, we focus on the adaptation of five of the most popular schemes for ordering-based pruning [51].
- To carry out a thorough experimental study in order to analyze the usefulness of this methodology in the imbalanced scenario. More specifically, we carry out an exhaustive comparison of all the adaptation of the five metrics so as to verify their results with the state-of-the-art ensembles on the topic, which were those previously stressed in [23].
- To study the true benefits of the application of these new metrics both with respect to the baseline methodologies and the state-of-the-art models. It will be shown that incorporating ensemble-pruning allows one to go a step further into the performance of ensemble-based solution.

For a fair evaluation of the ordering-based pruning in imbalanced classification, we have selected the best bagging- and boosting-based ensemble models that were highlighted in our previous study on the topic [23]. Finally, the validation of the novel imbalanced pruning methodology will be carried out using a wide benchmark of 66 different problems commonly used in this area of research [46], and supported by means of the statistical analysis of the results [24].

The rest of this paper is organized as follows. Section 2.1 introduces classification with ensembles for the problem of imbalanced data, as well as the ordering-based pruning approach with the metrics considered to perform this process. Then, Section 3 contains the core part of the manuscript, in which we present our adaptations to imbalanced classification for

the pruning methodology. Next, the details about the experimental framework regarding datasets, algorithms, and statistical tests are provided in Section 4. The analysis and discussion of the experimental results is carried out in Section 5. Next, Section 6 provides the most significant findings achieved throughout the experimental analysis. Finally, Section 7 summarizes and concludes this work.

2. Ensemble learning and ordering-based pruning

In this section, we will first introduce the features of ensemble-based classifiers, and we will enumerate those approaches that have been developed in the field of imbalanced classification (Subsection 2.1). Next, we will describe the working procedure of ordering-based pruning in detail, and we will present some of the most commonly used metrics that can be used to guide the ordered aggregation (Subsection 2.2).

2.1. Ensemble methods

Ensemble-based classifiers, also known as multiple classifier systems [59], are composed by a set of so-called *weak learners*. The former name refers to the case that a classifier provides a better output than just random guessing, but not close enough to the true classification. This fact leaves room for improvement with respect to each independent member of the ensemble. Additionally, diversity among classifiers, is crucial (but not enough) for the success of these types of methods [34,41,68]. Finally, when a new query instance is submitted to the system, the predictions of all classifiers are aggregated in order to obtain a single output. In this way, the global combination aims to outperform the accuracy of the individual classifiers, i.e., to obtain a better generalization [76].

Ensemble-based methods have been successfully adapted to classification with imbalanced datasets [23]. This can be done by including preprocessing or cost-sensitive learning into the ensemble learning algorithm. In accordance with their good capabilities for addressing the class imbalance problem, these type of methods have attracted a great interest among researchers in the recent years [39,42,73].

There are two main types of techniques when building an ensemble, i.e., Bagging [8] and Boosting [21]:

- **Bagging:** it is the acronym for *bootstrap aggregating*. It trains a set of classifiers, each one with a different subset (known as "bag") of the original training data. The drawing of instances for each bag is made at random (with replacement), so that the original training set size is maintained. In this method, diversity is achieved by means of the resampling procedure. When classifying a new sample, all individual classifiers are fired and a majority or weighted vote is used to infer the class.
  We must point out that most of the ensemble techniques adapted for imbalanced classification have followed this scheme [23]. This is due to the simplicity for the integration of data preprocessing techniques into Bagging, which is made when each bootstrap replica is computed.
- **Boosting:** in this method, the whole training set is used to learn each classifier. However, in each round, i.e., when a new classifier is trained, the algorithm put its focus on the most difficult instances, that is, those that were misclassified in previous iterations. This can be achieved by weighting the instances in the dataset. Weights are equally set for all instances at the beginning. Then, misclassified instances get their weights increased, whereas correct hits result in a lower weight. Additionally, each individual classifier also gets a *score* depending on its overall accuracy over the training set. Higher confidence is given to more accurate classifiers. Finally, when a new instance is submitted, each classifier gives a weighted vote (according to its *score*), and the class label is selected by majority.
  When dealing with imbalanced data, these types of ensembles alter and bias the weight distribution used to train the next classifier towards the minority class in every iteration [23].

The description of the ensemble methods used in our current study is carried out in Section 4, where the setting of the experimental framework is presented.

2.2. Pruning in ordered ensembles

When building an ensemble model we must first consider the number of classifiers it is going to be composed of. We must be aware that a correct choice of this parameter value have a significant influence on the behavior of the final model. A low number of base classifiers may cause that the ensemble may not reach a high and stable classification accuracy. On the contrary, the more base classifiers are included, the higher the probability of redundant classifiers is, resulting on less diversity [43], and especially in the case of boosting, the higher the probability of over-fitting. Therefore, having too many classifiers more resources are wasted, both in terms of memory requirement to store the classifiers in the ensemble and processing time.

Ordering-based pruning methods [1] were initially defined for bagging-based ensembles, although they can also be applied in boosting-based ones. In this case, the order of aggregation is unspecified, i.e., classifiers were built in random order so

[1] Even though they are known as "pruning" methods, since all classifiers must be learn before-hand and they are reduced afterwards, they can also be regarded as "aggregation" procedures, since classifiers are added one at a time to the final ensemble.

that the learning process could be inherently parallelized. The basis of these approaches is that an enhancement of the behavior of the final ensemble will be achieved if those classifiers that are expected to perform better are first added [28,31,49,51,52,82]. In this way, the final size of the ensemble is also reduced without affecting accuracy of the whole model. In order to compute this estimated value of the goodness of the different classifiers to be added, several metrics can be considered, whose modification is proposed to work in an imbalanced scenario in this work.

These types of ensemble pruning are popular due to their effectiveness with respect to their low computational cost. As in every pruning technique, the process is carried out once all classifiers of the ensemble have been trained. Hence, pruning becomes a combinatorial optimization problem, where the best subset of classifiers must be found. However, optimal ensemble pruning is known to be an NP-complete combinatorial problem [67]. Therefore, most of the pruning techniques make use of an heuristic function to seek for the reduced set of classifiers. In the case of ordering-based pruning, a metric that measures the goodness of adding each classifier to the ensemble is defined and the classifier with the highest value is added to the final sub-ensemble. The same process is performed until the size of the sub-ensemble reaches the specified parameter value. Previous studies on this parameter value suggest that it should be established between 20 and 40% of all the classifiers [51,52]. We acknowledge that the use of a threshold number for the final set of classifiers may lead to sub-optimal models. However, we consider the same configuration which was already tested in a standard framework, supported by the robust results achieved in the former referenced studies.

In what follows, the pruning metrics considered in this work are described:

- *Reduce-Error pruning (RE)* [50]: this method works by first adding to the final sub-ensemble the classifier achieving the lowest classification error. Afterwards, in each iteration classifiers are ordered by the error they produce when added to the sub-ensemble. The one that achieves the largest error reduction (or performance improvement) is added.
- *Kappa pruning (Kappa)* [18,50]: in this method, the most diverse ensemble is sought. In order to do so, κ statistic, commonly used as a diversity measure in classifier ensembles [40], is used to measure pairwise diversity between classifiers. κ measures the level of agreement between the outputs of two classifiers, giving a value of 1 if they completely agree, 0 if they are statistically independent and negative values account for negative correlation. In the original Kappa pruning model, the pair of classifiers with the greatest diversity was added to the ensemble. However, this model was affected by the fact that it did not take into account the diversity of the unselected classifiers with respect to the ensemble, decreasing the performance of the whole ensemble [82].
  On this account, an improvement was proposed in [51], which is also considered in this work. Initially, the pair of classifiers with the greatest diversity is selected, and then the classifier achieving the largest diversity with respect to the sub-ensemble is selected, that is, κ is computed with respect to the sub-ensemble instead of among all unselected classifiers.
- *Complementarity Measure (Comp)* [57]: this method starts from the most accurate single classifier (as in *RE*), and then iteratively adds the classifier that better complements the sub-ensemble. *Complementarity* is measured as the number of correctly classified examples by the classifier from those that are misclassified by the sub-ensemble. This measure reflects how much the classifier could change the decision of the ensemble.
- *Margin Distance Minimization (MDM)* [57]: this is a more complex scheme than previous ones, based on certain distances among the output vectors of the ensembles. These output vectors have the length equal to the training set size, and their value at the ith position is either 1 or -1 depending on whether the ith example is classified or misclassified by the classifier. The signature vector of a sub-ensemble is computed as the sum of the vectors of the selected classifiers. To summarize, the aim is to add those classifiers with the objective of obtaining a signature vector of the sub-ensemble where all the components are positive, i.e., all examples are correctly predicted. For a wider description please refer to [51,57]. We have followed the implementation presented in [51], where a small improvement was presented with respect to the original model [57].
- *Boosting-Based pruning (BB)* [21,52]: this method selects the classifier that minimizes the cost with respect to the boosting scheme. This means that boosting algorithm is applied to compute the weights (costs) for each example in each iteration, but instead of training a classifier with these weights, the one that obtains the lowest cost from those in the pool is added to the sub-ensemble and weights are updated accordingly. Hence, it makes no difference whether classifiers were already learned using a boosting scheme or not. Different from the original boosting method, when no classifier has a weighted training error below 50%, weights are reinitialized (equal weights for all the examples) and the method continues (whereas in boosting it is stopped). Once classifiers are selected the scores assigned to each classifier by boosting are forgotten and not taken into account in the aggregation phase.

Detailed explanations of the corresponding metrics can be found in the corresponding source papers or in the previous analysis in [51]. Notice that these metrics are computed over the training sets, since there is usually not enough data so as to divide it into two sets: one for training the classifiers and the other to prune the ensemble. In these cases, the usage of an independent set for pruning do not compensate the decrease in accuracy in the classifiers of the ensemble.

3. A proposal for ordering-based pruning scheme for ensembles in imbalanced domains

Ordering-based pruning methods have shown to be effective in standard classification. However, the properties related to the scenario of imbalanced classification suggest the necessity of an adaptation of the standard metrics towards this

framework. In this research, we propose a novel definition of five heuristic metrics to boost the recognition of both the minority and majority classes of the problem. Additionally, for the sake of clarity, and also to allow researchers to easily reproduce this new approach, we try to keep their formulation as simple as possible.

In order to do so, we will first describe our adaptations of these metrics for carrying out the selection process in imbalanced classification ensembles (Subsection 3.1). Then, we will depict a graphical example in order to explain the goodness of this approach (Subsection 3.2).

3.1. New pruning metrics adapted for imbalanced domains

In this section, we present our novel ordering-based pruning schemes for classification with imbalanced datasets. The aim is to develop ad-hoc pruning solutions to enhance the behavior of ensemble techniques in this framework. In order to do so, for each one of the metrics introduced in the previous section, we propose a new contextualized adaptation for imbalanced classification (except for the cases that, from our point of view, it is not required). Proceeding this way, they will be able to manage the skewed class problem.

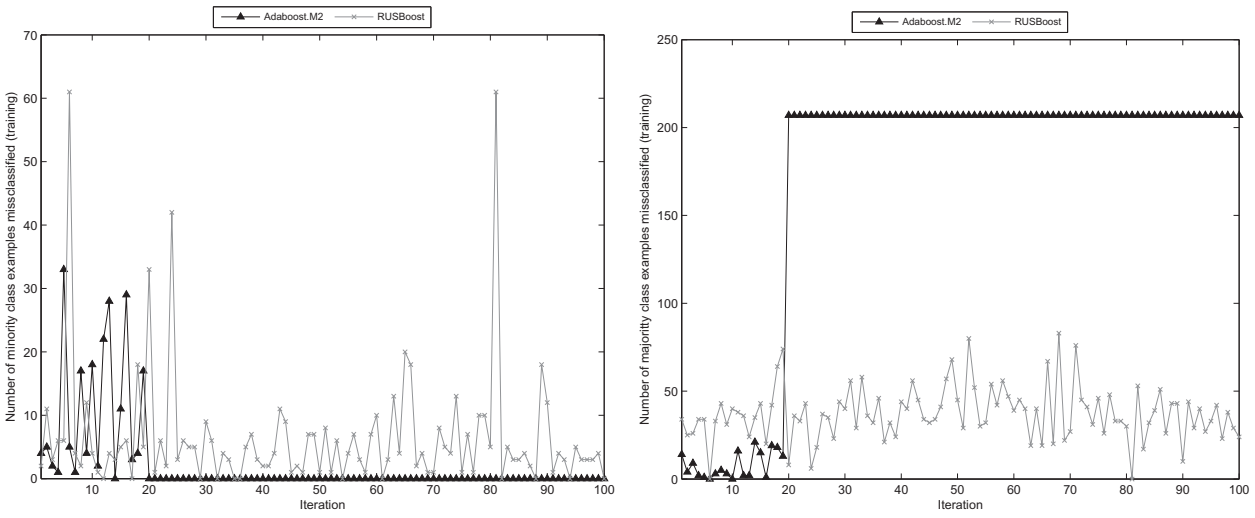
As we will show in the experimental study, the extension of some approaches is straightforward by changing the evaluation metric, whereas others are more complex and their behavior should be studied. Hence, these last adaptations will be also compared against the original models in the experimental analysis.

- *Reduce-Error pruning with Geometric Mean (RE-GM)*: Since classification error is computed as the number of misclassified examples, it does not equally take into account both classes of the problem. Hence, it is biased towards the majority class, being unable to properly work with imbalanced class distributions. For this reason, we have considered the GM performance measure to establish the order of the classifiers in such a way that the one with largest value of this metric is considered when to be added to the ensemble.
- *Kappa pruning (Kappa)*: Given that this method does not make use of any performance measure affected by the class imbalance problem, no adaptation is needed for the framework of imbalanced classes. Specifically, kappa scores independently the successes for each class and aggregates them. This way of scoring is less sensitive to randomness caused by a different number of examples in each class.
- *Complementarity Measure (Comp)*: In this case, there is also no need for modification. The reason behind this behavior is that most of the ensembles overfits in favor of the minority class (given that weights are positively weighted towards these "difficult" examples). Regarding this fact, "Comp" metric tries to enhance the classification of the majority class examples without being detrimental to the minority ones, which are expected to be already well classified.
- *Margin Distance Minimization for imbalanced problems (MDM-Imb)*: As described in the previous section, this method selects the classifier to be added depending on the closest Euclidean distance between an objective point (where all components are positive) and the signature vector of the sub-ensemble after adding the corresponding classifier. As a consequence, every example has the same weight in the computation of the distance, which can bias the selection to those classifiers favoring the majority class. Therefore, we compute the distance for the majority and minority class examples independently. Then, distances are normalized by the number of examples used to compute them and added afterwards.
- *Boosting-Based pruning for imbalanced problems (BB-Imb)*: It is well-known that boosting by itself is not capable of managing class imbalance problem [23]. For this reason, we have also adapted this approach in a similar manner as in the case of *MDM*. In boosting, every example has initially the same weight and these are updated according to whether they are correctly classified or not. Hence, before finding the classifier that minimizes the total cost, we normalize the weights of the examples of each class by half of their sum, so that both classes has the same importance when selecting the classifier (even though each example of each class would have a different weight). This is only done before selecting the classifier, and then weights are updated according to the original (non-normalized ones).

3.2. Example of the behavior of ordering-based pruning in imbalanced datasets

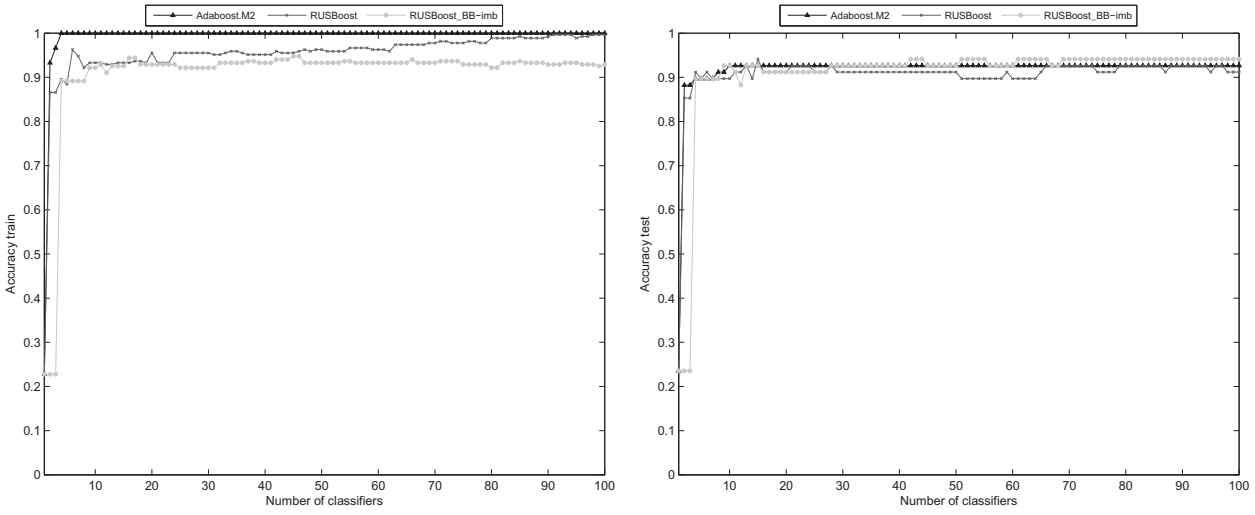
Standard ensemble learning algorithms, such as AdaBoost [21], focus on difficult examples disregard their class. When facing an imbalanced class problem, minority class examples are more likely to be positively weighted throughout the training process. In this sense, there is a limit for adding new classifiers to the ensemble, as all examples will be classified as the minority class. This issue justifies the need for using an specific approach to address the imbalance such as RUS-Boost [63], which removes instances from the majority class by randomly undersampling the dataset in each iteration. This behavior is shown in Fig. 1(a) and (b), from which we can observe that, in the case of AdaBoost.M2, new classifiers do not contribute to improve the recognition ability of the ensemble from 20 classifiers onwards. In contrast, we realize that RUS-Boost shows a wider diversity among classifiers.

In order to complement this analysis, we show how the performance evolves depending on the number of classifiers included within the ensemble. Specifically, we show the differences between the use of the standard accuracy metric (Fig. 2(a) and (b) and the Area Under the ROC curve (AUC) [35] (Fig. 3(a) and (b) for AdaBoost.M2, RUS-Boost, and RUS-Boost with boosting-Based pruning (*BB-Imb*). We must point out that in the case of *BB-Imb* a maximum of 21 classifiers are selected from the pool.



(a) Minority class (b) Majority class

Fig. 1. Number of misclassified examples when new classifiers are added to the ensemble of each class (ecoli1 dataset).



(a) Training (b) Test

Fig. 2. Variation of accuracy with respect to the number of classifiers used in the ensemble (ecoli1 dataset).

Considering the differences between the results in accuracy and AUC, the original AdaBoost approach has problems to recognize the minority class, and it presents a clear overfitting in both cases. With RUS-Boost, this behavior is corrected, but the performance is unstable with respect to the number of classifiers. Finally, the advantages are clear when the pruning mechanism is introduced, as it obtains the highest results for the AUC metric.

4. Experimental framework

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

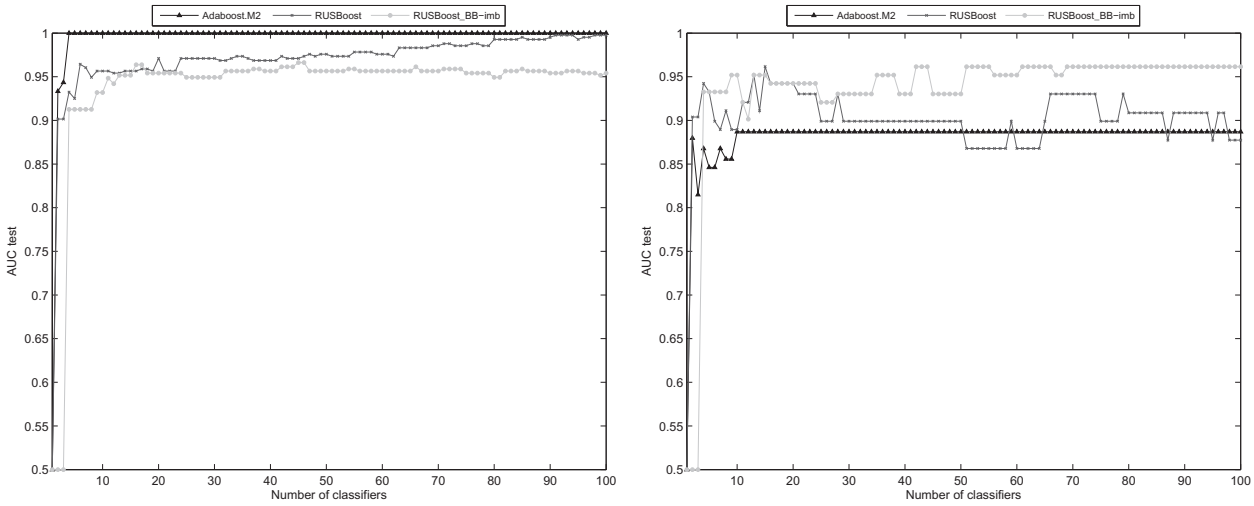


Fig. 3. Variation of AUC with respect to the number of classifiers used in the ensemble (ecoli1 dataset).

Table 1
Summary of imbalanced datasets used.

Name	#Ex.	#Atts.	IR	Name	#Ex.	#Atts.	IR
glass1	214	9	1.82	glass04vs5	92	9	9.22
ecoli0vs1	220	7	1.86	ecoli0346vs5	205	7	9.25
wisconsin	683	9	1.86	ecoli0347vs56	257	7	9.28
pima	768	8	1.90	yeast05679vs4	528	8	9.35
iris0	150	4	2.00	ecoli067vs5	220	6	10.00
glass0	214	9	2.06	vowel0	988	13	10.10
yeast1	1484	8	2.46	glass016vs2	192	9	10.29
vehicle2	846	18	2.52	glass2	214	9	10.39
vehicle1	846	18	2.52	ecoli0147vs2356	336	7	10.59
vehicle3	846	18	2.52	led7digit02456789vs1	443	7	10.97
haberman	306	3	2.68	ecoli01vs5	240	6	11.00
glass0123vs456	214	9	3.19	glass06vs5	108	9	11.00
vehicle0	846	18	3.23	glass0146vs2	205	9	11.06
ecoli1	336	7	3.36	ecoli0147vs56	332	6	12.28
newthyroid2	215	5	4.92	cleveland0vs4	1771	13	12.62
newthyroid1	215	5	5.14	ecoli0146vs5	280	6	13.00
ecoli2	336	7	5.46	ecoli4	336	7	13.84
segment0	2308	19	6.01	shuttle0vs4	1829	9	13.87
glass6	214	9	6.38	yeast1vs7	459	8	13.87
yeast3	1484	8	8.11	glass4	214	9	15.47
ecoli3	336	7	8.19	pageblocks13vs4	472	10	15.85
pageblocks0	5472	10	8.77	abalone918	731	8	16.68
ecoli034vs5	200	7	9.00	glass016vs5	184	9	19.44
yeast2vs4	514	8	9.08	shuttle2vs4	129	9	20.50
ecoli067vs35	222	7	9.09	yeast1458vs7	693	8	22.10
ecoli0234vs5	202	7	9.10	glass5	214	9	22.81
glass015vs2	506	8	9.12	yeast2vs8	482	8	23.10
yeast0359vs78	172	9	9.12	yeast4	1484	8	28.41
yeast0256vs3789	1004	8	9.14	yeast1289vs7	947	8	30.56
yeast02579vs368	1004	8	9.14	yeast5	1484	8	32.78
ecoli046vs5	203	6	9.15	yeast6	1484	8	39.15
ecoli01vs235	244	7	9.17	ecoli0137vs26	281	7	39.15
ecoli0267vs35	244	7	9.18	abalone19	4174	8	128.87

4.1. Benchmark data

Table 1 shows the 66 benchmark problems selected for our study, in which the name, number of examples, number of attributes, and IR (ratio between the majority and minority class instances) are shown. Datasets are ordered with respect to their degree of imbalance. Multi-class problems were modified to obtain two-class imbalanced problems, defining the joint of one or more classes as positive and the joint of one or more classes as negative, as defined in the name of the dataset.

For the experimental analysis, we will take into account both the AUC [35] and Geometric Mean (GM) of the true rates [2] measures. Both metrics aim at maximizing the joint performance of the classes, but their different intrinsic factors allow a complementary analysis from the experimental results. The former allows to determine the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}), whereas the latter attempts to maximize the accuracy of each one of the two classes at the same time (sensitivity and specificity). The estimates for these metrics will be obtained by means of a Distribution Optimally Balanced Stratified Cross-Validation (DOB-SCV) [56], as suggested in the specialized literature for working in imbalanced classification [47]. DOB-SCV avoids dataset shift [55], which hinders the results obtained in the experimental analysis. This procedure is carried out using 5 folds, aiming to include enough positive class instances in the different folds. In this way, we avoid additional problems in the data distribution, especially for highly imbalanced datasets. In accordance with the stochastic nature of the learning methods, these 5 folds are generated with 5 different seeds, and each one of the 5-fold cross-validation is run 5 times. Therefore, experimental results for each method and dataset are computed with the average of 125 runs.

4.2. Description of the ensemble approaches

In this section we briefly survey the selected ensemble methods for our current study. The first three algorithms are Bagging-based, the next three are Boosting-based ensembles, whereas the last one is a hybrid approach.

First of all, we briefly recall the operating procedure of the most widely used data-level methods prior to introduce the operation procedure of each ensemble model. These are the random undersampling method and the Synthetic Minority Oversampling Technique (SMOTE) [12]:

- **Random Undersampling**. It balances the class distribution by randomly eliminating majority class examples. Its major drawback is that potentially useful data can be discarded.
- **SMOTE**. It is an oversampling method that creates minority class examples by interpolating several minority class instances that lie together (the k Nearest Neighbors).

The ensemble-based models considered are presented hereafter.

- *SMOTE-Bagging* [72]: In this case, SMOTE oversampling procedure [12] is applied before training each classifier. All instances will probably take part in at least one bag, but each bootstrapped replica will contain many more instances than the original dataset (twice the number of majority class examples). The number of majority class instances to be included is fixed throughout all the process, but the resampling rate for minority class instances is set in each iteration (ranging from 10% in the first iteration to 100% in the last, always being multiple of 10). Then, the rest of the instances from the minority class are generated by SMOTE algorithm. Majority class instances are resampled with replacement.
- *Under-Bagging* [3]: On the contrary to *SMOTE-Bagging*, *Under-Bagging* procedure uses undersampling instead of oversampling, and more specifically, it does it at random. Also, in contrast to *SMOTE-Bagging* all bags are built following the same procedure (in all bags all the minority class instances are considered).
- *Roughly-Bagging* [6,32]: It is based on undersampling, but the main difference is that it aims at equalizing the sampling probability of each class, instead of fixing the sample size as a constant. The size of the majority examples is determined probabilistically according to a negative binomial distribution, whereas the number of minority examples is always the same. As a consequence, the class distribution of the sampled subsets may become slightly imbalanced, favoring diversity. In both classes, resampling with or without replacement can be applied. The implementation considered in this work includes all minority class examples in the bag, and does the resampling with replacement only over the majority class.
- *SMOTE-Boost* [14]: this method introduces synthetic instances using SMOTE [12] before computing the new weights of the examples in the Boosting procedure. Thus, the weights of these new examples will be proportional to the total number of instances in the new dataset. Then, the examples from the original training set are normalized according to the new data distribution.
- *RUS-Boost* [63]: it works similarly to SMOTE-Boost, but removing instances from the majority class by random undersampling the dataset in each iteration. In this case, it is not necessary to assign new weights to the instances. It is enough by simply normalizing the weights of the remaining instances in the new dataset with respect to their total sum of weights.
- *EUS-Boost* [22]: this technique follows the same scheme as RUS-Boost but with two main differences: (1) it uses the Evolutionary UnderSampling (EUS) approach [26] as preprocessing method; (2) it promotes the diversity of the ensemble using the Q-statistic diversity measure [77] in the evolutionary process.
- *EasyEnsemble* [44]: is a hybrid approach that combines both Bagging and Boosting. Specifically, it uses Bagging as main ensemble learning method, but in spite of training a classifier for each new bag, each bag is trained using AdaBoost [21]. In order to account for class imbalanced, each bag is balanced by means of random undersampling.

A complete taxonomy for ensemble methods in learning with imbalanced classes can be found in a recent review [23], where the reader may refer for a wider description of these techniques.

4.3. Selected parameters

Classification. In order to do so, we will make use of the best behaving ensemble algorithms highlighted in our previous study at [23], i.e., SMOTE-Bagging [72], Under-Bagging [3], SMOTE-Boost [14], RUS-Boost [63], and Easy-Ensemble [44].

Additionally, we have included a new ensemble learning model, EUS-Boost [22] which has been shown to be very robust for highly imbalanced datasets. We must recall that the description of these models were given in Section 2.1. Reader may also refer to [23] in order to get a thorough description of the former ensemble methods. In all ensemble, we will employ the C4.5 decision tree [61] as baseline classifier, since almost all the previous ensemble methodologies were proposed in combination with C4.5. Additionally, it has been widely used to deal with imbalanced data-sets [10,45,58]. Finally the significance of C4.5 can be stressed with respect to its presence as one of the top-ten data-mining algorithms [76].

Next, we detail the parameter values for the different learning algorithms selected in this study. The selection of the former has been made according to typical configurations from the specialized literature:

1. **C4.5**: the confidence level will be set at 0.25, with 2 being the minimum number of item-sets per leaf, and the application of pruning will be used to obtain the final tree.
2. **Ensemble classifier structure**: In order to apply the pruning procedure, we will learn 100 *classifiers* for each ensemble, choosing a subset of only 21 *classifiers* following the previous study on ordering-based ensemble pruning in standard classification [51].
   The baseline ensemble models for comparison will only use 10 classifiers for Boosting approaches and 40 for Bagging, in accordance with the optimal parameters found in our previous studies [23]. We must point out that it makes no sense to make the comparison with 100 classifiers as it has been stressed that this option does not generally achieve better results, in accordance with the issues raised throughout this manuscript.
   For *SMOTE-Bagging* and *SMOTE-Boost*, SMOTE configuration will be the standard with a 50% class distribution, 5 neighbors for generating the synthetic samples, and Heterogeneous Value Difference Metric for computing the distance among the examples. In the case of *Roughly-Bagging*, the negative binomial distribution is constructed with $p = 0.5$ and n equal to the number of minority examples. For *EUS-Boost*, EUS configuration is the recommended one with GM as evaluation measure, majority selection only, Euclidean distance, promoting the balancing of the dataset (with balancing factor equal to 0.2). Finally, in the case of *EasyEnsemble*, we will consider 4 bags and 10 iterations for the AdaBoost algorithm.
3. **CHC optimization process:** The *EUS-Boost* ensemble algorithm was developed from the CHC algorithm (CHC stands for Cross Generational elitist selection, Heterogeneous recombination and Cataclysmic mutation). The number of individuals has been set to 50 chromosomes, and a total of 10, 000 evaluations will be carried out throughout the genetic process.

We must also point out that all the algorithms from the state-of-the-art selected in this study are available within the KEEL software tool [1].

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 [24]. 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 [17,25]. Any interested reader can find additional information on the Website http://sci2s.ugr.es/sicidm/.

First of all, we consider the method of aligned ranks of the algorithms in order to show at a first glance how good a method is with respect to its partners. In order to compute this ranking, the first step is to obtain the average performance of the algorithms in each dataset. Next, we compute the subtractions between the accuracy of each algorithm minus the average value for each dataset, which is computed using the output results for all algorithms considered in the comparison. 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 that achieves the **lowest average ranking** is the best one.

The Friedman Aligned test [24] will be used to check whether there are significant differences among the results, and the Holm post-hoc test [33] in order to find which algorithms reject the hypothesis of equality with respect to a selected control method in a $1 \cdot n$ comparison. We will compute the adjusted p-value (APV) associated with each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. This value differs from the standard p-value in the sense that it determines univocally whether the null hypothesis of equality is rejected at a significance level α. For more details on the APV we refer the reader to [24].

Regarding pairwise comparisons, we will make use of Wilcoxon signed-rank test [74] to find out whether significant differences exist between a pair of algorithms. This procedure computes the differences between the performance scores of the two classifiers on each one of the available datasets (N_{ds}). The differences are ranked according to their absolute values, from smallest to largest, and average ranks are assigned in case of ties. We call R^+ the sum of ranks for the datasets 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 [79]), the null hypothesis of equality of means is rejected.

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/prune-imbalanced/), where we have included the following information:

- A more complete description of the ensemble techniques for addressing imbalanced data-sets.

Table 2
Average test results for the standard and imbalanced pruning metrics (using AUC).

Ensemble	BB	BB-Imb	MDM	MDM-Imb
SMOTE-Bagging	.8602 ± .0632	**.8635 ± .0610**	.8596 ± .0629	**.8625 ± .0622**
Under-Bagging	**.8755 ± .0564**	.8734 ± .0544	.8653 ± .0563	**.8699 ± .0558**
Roughly-Bagging	**.8747 ± .0555**	.8737 ± .0546	.8659 ± .0568	**.8709 ± .0557**
SMOTE-Boost	.8486 ± .0673	**.8514 ± .0658**	.8447 ± .0653	**.8507 ± .0660**
RUS-Boost	.8652 ± .0585	**.8679 ± .0540**	.8575 ± .0604	**.8650 ± .0564**
EUS-Boost	.8629 ± .0647	**.8697 ± .0544**	.8559 ± .0668	**.8602 ± .0639**
EasyEnsemble	.8456 ± .0616	**.8678 ± .0549**	.8392 ± .0540	**.8675 ± .0558**

Table 3
Average test results for the standard and imbalanced pruning metrics (using GM).

Ensemble	BB	BB-Imb	MDM	MDM-Imb
SMOTE-Bagging	.8401 ± .0859	**.8454 ± .0812**	.8391 ± .0858	**.8446 ± .0830**
Under-Bagging	**.8689 ± .0674**	.8674 ± .0636	.8554 ± .0686	**.8636 ± .0655**
Roughly-Bagging	.8673 ± .0666	**.8673 ± .0647**	.8550 ± .0709	**.8643 ± .0660**
SMOTE-Boost	.8215 ± .0964	**.8249 ± .0943**	.8130 ± .0955	**.8239 ± .0945**
RUS-Boost	.8550 ± .0730	**.8628 ± .0599**	.8390 ± .0807	**.8595 ± .0632**
EUS-Boost	.8471 ± .0895	**.8642 ± .0612**	.8346 ± .0956	**.8476 ± .0835**
Easy-Ensemble	.8088 ± .0963	**.8618 ± .0626**	.7750 ± .0882	**.8612 ± .0639**

- The data-sets partitions employed in the paper.
- Several Excel files containing full training and test results for all the experiments carried out in this study. In this way, any interested researcher can use them to include their own results and extend the present study.

5. Experimental study

In this section, we will carry out our experimental analysis into three incremental steps:

1. First, we will study the goodness of the new metrics developed specifically for dealing with imbalanced classification with respect to the original ones. As we have mentioned (Subsection 5.1), we will study whether the new metrics based on *BB* and *MDM* make a difference with respect to the baseline ones. Notice that this comparison is not needed for the rest of the metrics because their extension was straightforward.
2. Then, we will perform a complete comparison among all different pruning metrics and the baseline (unpruned) ensembles. This will lead us to the best approach for each particular ensemble method (Subsection 5.2). The ultimate goal is to confirm whether the pruning methodology is consolidated as a better option than not carrying out pruning for addressing the classification with imbalanced data using ensembles.
3. Finally, once we have highlighted the best synergies between ensemble techniques in imbalanced classification and pruning schemes, we will carry out a full experimental analysis among all methodologies. This analysis will lead us to discover whether there is a combination that stands out among all ensemble models after being pruned (Subsection 5.3).

We must point out that all the findings extracted throughout this experimental analysis are based in the output of statistical tests, i.e., average ranking and *p*-values. However, we have also included the average performance results to provide a reference of the global quality of the different methodologies selected for this study. In this way, any interested researcher can be aware of the performance shown in this work in contrast with their own methods.

5.1. Pruning metrics for imbalanced classification

Our first analysis is focused on determining whether the new proposed metrics, specifically designed for dealing with class imbalance, are well-suited for this problem with respect to the original ones, i.e., *BB* and *MDM*. The average experimental results in testing phase for these metrics are shown in Table 2 for AUC and Table 3 for GM.

For both performance measures, we may observe a similar behavior. In the case of *BB* and *BB-Imb* metrics, we find that in all cases the metric adapted for imbalanced classification achieves a higher average performance, with the only exceptions of Under-Bagging and Roughly-Bagging, in which the relative differences are below 1%. Regarding *MDM* and *MDM-Imb*, looking at the results the need for the imbalanced approach stands out. Finally, the robustness of the imbalanced metrics must be stressed in accordance with the low standard deviation shown with respect to the standard case.

In order to statistically determine the best suited metric for each ensemble model, we carry out a Wilcoxon pairwise test for both AUC and GM in Tables 4 and 5. We have included a symbol for stressing whether significant differences are found at 95% confidence degree ($*$) or at 90% (+). Finally, we also show the number of problems in which the baseline scheme wins/ties/loses in performance with respect to our new proposed imbalanced metric. In this way, we stress an additional factor of improvement for the best performing approach.

Table 4
Wilcoxon test to compare the standard [R^+] and imbalanced pruning metrics [R^-] regarding the AUC metric. Symbol $*$ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the standard algorithm.

Ensemble	Comparison	R^+	R^-	p-value	W/T/L
SMOTE-Bagging	BB vs. BB-Imb	540.0	1671.0	0.00028$*$	20/3/43
	MDM vs. MDMimb	436.0	1775.0	0.00002	16/4/46
Under-Bagging	BB vs. BB-Imb	1277.0	934.0	0.27939	36/3/27
	MDM vs. MDMimb	831.5	1379.5	0.07246+	25/6/35
Roughly-Bagging	BB vs. BB-Imb	1014.0	1197.0	0.53957	32/4/30
	MDM vs. MDMimb	1349.5	861.5	0.10376	27/5/34
SMOTE-Boost	BB vs. BB-Imb	1176.0	1035.0	0.74441	10/24/32
	MDM vs. MDMimb	495.5	1715.5	0.00003$*$	9/25/32
RUS-Boost	BB vs. BB-Imb	975.0	1236.0	0.43512	26/3/37
	MDM vs. MDMimb	1176.0	1035.0	0.74441	37/4/25
EUS-Boost	BB vs. BB-Imb	1024.0	1187.0	0.62692	28/3/35
	MDM vs. MDMimb	1133.0	1078.0	0.91346	34/4/28
Easy-Ensemble	BB vs. BB-Imb	669.5	1541.5	0.00439$*$	27/2/37
	MDM vs. MDMimb	868.0	1343.0	0.10161	30/4/32

Table 5
Wilcoxon test to compare the standard [R^+] and imbalanced pruning metrics [R^-] regarding the GM metric. Symbol $*$ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the standard algorithm.

Ensemble	Comparison	R^+	R^-	p-value	W/T/L
SMOTE-Bagging	BB vs. BB-Imb	423.0	1788.0	0.00010$*$	16/3/47
	MDM vs. MDMimb	361.0	1850.0	0.00000	13/4/49
Under-Bagging	BB vs. BB-Imb	1210.0	1001.0	0.51988	35/3/28
	MDM vs. MDMimb	779.5	1431.5	0.02933	25/6/35
Roughly-Bagging	BB vs. BB-Imb	1129.0	1082.0	0.86913	31/4/31
	MDM vs. MDMimb	1373.5	837.5	0.06648	28/5/33
SMOTE-Boost	BB vs. BB-Imb	601.0	1610.0	0.00094$*$	11/24/31
	MDM vs. MDMimb	555.5	1655.5	0.00005$*$	11/25/30
RUS-Boost	BB vs. BB-Imb	829.0	1382.0	0.08954+	22/3/41
	MDM vs. MDMimb	1096.0	1115.0	0.83615	36/4/26
EUS-Boost	BB vs. BB-Imb	851.0	1360.0	0.11222	24/3/39
	MDM vs. MDMimb	1028.0	1183.0	0.56773	32/4/30
Easy-Ensemble	BB vs. BB-Imb	605.5	1605.5	0.00115$*$	25/2/39
	MDM vs. MDMimb	785.0	1426.0	0.03055$*$	28/4/34

The results of these tests agree with our previous remarks. In the case of *BB-Imb*, differences are clearer with GM, in which the standard *BB* metric is outperformed in all cases but in Under-Bagging and Roughly-Based Bagging (for EUS-Boost p-value is 0.11222, near to the 90% of confidence degree). Even though with AUC the differences are not statistically as meaningful, taking into account both measures, it can be concluded that *BB-Imb* is better suited for this framework. Regarding *MDM-Imb* we can also stress the goodness of this approach in accordance with the ranks and the p-values for all types of ensemble algorithms. However, boosting-based approaches with undersampling, i.e., RUS-Boost and EUS-Boost do not show significant differences (although ranks are in favor of the new model). Therefore, it can be concluded that due to the way these models are trained, they become less affected to class imbalance in pruning phase (but anyway ranks are in favor of the imbalanced approach, and hence it does not hinder the pruning mechanism).

In accordance with this analysis, we have selected the adapted imbalanced metrics *BB-Imb* and *MDM-Imb* for subsequent analyses.

5.2. Analysis of the pruning metrics of ensembles in imbalanced classification

This section is devoted to study the behavior of the selected pruning metrics for the learning of ensembles in the scenario of imbalanced classification. We aim to highlight which approach or approaches allow one to enhance the performance of the baseline ensemble methodologies designed for this problem. Our main objective is to analyze whether the ordering-based pruning approach is also able to excel in this framework.

For this experimental analysis, we will compare 6 different approaches including the *baseline* scheme (without pruning), the 2 metrics already studied in the previous section (*BB-Imb* and *MDM-Imb*), and the 3 remaining metrics *Comp*, *Kappa*, and *RE* (using GM).

Table 6
Average test results (*AUC*), ranks (Friedman aligned) and APVs (Holm test) for **all ensemble pruning metrics**. Control method is pointed out with asterisks. Symbol $*$ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the control algorithm.

Ensemble	Method	AUC Test	Ranking	APV (Holm test)	W/T/L
SMOTE-Bagging	Baseline	.8645 ± .0587	175.90 (2)	0.27803	27/1/38
	BB-Imb	.8635 ± .0610	208.36 (3)	0.01330*	28/1/37
	MDM-Imb	.8625 ± .0622	228.40 (6)	0.00100*	26/1/39
	Comp	.8632 ± .0598	211.20 (4)	0.01318*	26/1/39
	Kappa	**.8681 ± .0578**	154.29 (1)	********	-/-/-
	RE-GM	.8633 ± .0602	212.84 (5)	0.01318*	28/1/37
Under-Bagging	Baseline	.8647 ± .0516	249.59 (5)	0.00000*	13/4/49
	BB-Imb	.8734 ± .0544	160.28 (3)	0.57376	27/4/35
	MDM-Imb	.8699 ± .0558	204.35 (4)	0.00315*	19/4/43
	Comp	.8748 ± .0539	144.90 (2)	0.76941	32/3/31
	Kappa	.8609 ± .0460	292.81 (6)	0.00000*	11/3/52
	RE-GM	**.8752 ± .0545**	139.06 (1)	********	-/-/-
Roughly-Bagging	Baseline	.8644 ± .0510	253.81 (5)	0.00000*	16/3/47
	BB-Imb	.8737 ± .0546	150.88 (3)	1.00000	38/3/25
	MDM-Imb	.8709 ± .0557	183.45 (4)	0.20313	26/3/37
	Comp	**.8745 ± .0538**	147.05 (1)	********	-/-/-
	Kappa	.8578 ± .0428	307.72 (6)	0.00000*	12/3/51
	RE-GM	.8744 ± .0543	148.08 (2)	1.00000	34/3/29
SMOTE-Boost	Baseline	.8523 ± .0655	171.88 (2)	1.0000	35/2/29
	BB-Imb	.8514 ± .0658	221.32 (5)	0.03701*	27/3/36
	MDM-Imb	.8507 ± .0660	226.67 (6)	0.02045*	22/3/41
	Comp	.8514 ± .0643	181.19 (3)	1.00000	10/42/14
	Kappa	.8447 ± .0672	220.48 (4)	0.03701*	31/3/32
	RE-GM	**.8525 ± .0642**	169.46 (1)	********	-/-/-
RUS-Boost	Baseline	.8654 ± .0605	170.75 (4)	1.0000	20/3/43
	BB-Imb	.8679 ± .0540	160.97 (3)	1.00000	31/3/32
	MDM-Imb	.8650 ± .0564	183.43 (5)	0.69528	26/3/37
	Comp	**.8682 ± .0563**	156.33 (1)	********	-/-/-
	Kappa	.7987 ± .0657	358.06 (6)	0.00000*	0/2/64
	RE-GM	.8677 ± .0562	161.45 (2)	1.00000	31/3/32
EUS-Boost	Baseline	.8678 ± .0586	176.43 (4)	0.55659	28/4/34
	BB-Imb	**.8697 ± .0544**	150.05 (1)	********	-/-/-
	MDM-Imb	.8602 ± .0639	199.09 (5)	0.05539+	16/3/47
	Comp	.8651 ± .0629	159.59 (2)	0.63215	35/3/28
	Kappa	.8133 ± .0688	335.65 (6)	0.00000*	1/2/63
	RE-GM	.8647 ± .0638	170.17 (3)	0.62513	33/3/30
Easy-Ensemble	Baseline	.8645 ± .0533	173.19 (3)	0.68909	20/2/44
	BB-Imb	**.8678 ± .0549**	154.36 (1)	********	-/-/-
	MDM-Imb	.8675 ± .0558	167.96 (2)	0.68909	33/4/29
	Comp	.8549 ± .0576	198.31 (4)	0.08214+	27/4/35
	Kappa	.8398 ± .0546	291.69 (6)	0.00000*	14/4/48
	RE-GM	.8539 ± .0589	205.45 (5)	0.04136*	31/4/31

In order to do so, Tables 6 and 7 show the average test results with AUC and GM performance measures, respectively. These tables also include the statistical comparison, showing the average ranks computed by the Friedman aligned test (whose computation was previously described in Section 4.4), and the APVs obtained by means of a Holm test. We explicitly stress whether there are statistical differences with a degree of confidence higher than 95% (symbol $*$) or 90% (symbol $+$). In this case, we also show the number of wins/ties/loses for each approach in comparison with the control method, i.e., that with the highest rank. This will serve as a complementary measure to the *p*-value for pointing out the degree of improvement achieved by the best heuristic metric.

From this study, we may stress the following conclusions:

1. The goodness of the pruning scheme for achieving high quality and competitive solutions in the scenario of imbalanced datasets. In all cases, the application of the pruning schemes result on a higher performance and ranking than that of the baseline model, as the pruning scheme wins in almost two-thirds of the benchmark problems. This is especially representative in the case of Under-Bagging, in which we observe significant differences between the best pruning model (*RE-GM*) and the standard ensemble learning algorithm.
2. The good synergy shown by *BB-Imb* rule with almost every ensemble, according to its average ranking for all types of ensemble models. This behavior is clearer for the Boosting and Bagging-based approaches with undersampling, that is, Under-Bagging, Roughly-Bagging, RUS-Boost, EUS-Boost and Easy-Ensemble.

Table 7
Average test results (*GM*), ranks (Friedman aligned) and APVs (Holm test) for **all ensemble pruning metrics**. Control method is pointed out with asterisks. Symbol $*$ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the control algorithm.

Ensemble	Method	GM Test	Ranking	APV (Holm test)	
SMOTE-Bagging	Baseline	.8477 ± .0778	168.18 (2)	0.11389	26/1/39
	BB-Imb	.8454 ± .0812	214.17 (3)	0.00020$*$	25/1/40
	MDM-Imb	.8446 ± .0830	234.23 (6)	0.00001$*$	24/1/41
	Comp	.8447 ± .0804	216.67 (4)	0.00017$*$	24/1/41
	Kappa	**.8547 ± .0741**	136.68 (1)	********	-/-/-
	RE-GM	.8445 ± .0816	221.03 (5)	0.00009$*$	25/1/40
Under-Bagging	Baseline	.8585 ± .0574	246.63 (5)	0.00001$*$	15/4/47
	BB-Imb	.8674 ± .0636	156.70 (3)	1.00000	30/4/32
	MDM-Imb	.8636 ± .0655	200.15 (4)	0.03478$*$	21/4/41
	Comp	**.8691 ± .0618**	149.96 (2)	1.00000	30/3/33
	Kappa	.8555 ± .0472	287.67 (6)	0.00000$*$	9/3/54
	RE-GM	.8683 ± .0646	149.86 (1)	********	-/-/-
Roughly-Bagging	Baseline	.8581 ± .0564	247.28 (5)	0.00000$*$	18/5/45
	BB-Imb	.8673 ± .0647	149.76 (1)	********	-/-/-
	MDM-Imb	.8643 ± .0660	180 (4)	0.38718	23/5/38
	Comp	**.8681 ± .0626**	153.4 (2)	1.00000	26/3/37
	Kappa	.8513 ± .0431	300.78 (6)	0.00000$*$	11/3/52
	RE-GM	.8672 ± .0650	159.78 (3)	1.00000	32/4/30
SMOTE-Boost	Baseline	.8269 ± .0942	165.42 (1)	********	-/-/-
	BB-Imb	.8249 ± .0943	223.81 (5)	0.01229$*$	25/2/39
	MDM-Imb	.8239 ± .0945	229.01 (6)	0.00707$*$	22/2/42
	Comp	.8246 ± .0919	181.26 (3)	0.85318	29/2/35
	Kappa	.8146 ± .1007	224.40 (4)	0.01229$*$	31/2/33
	RE-GM	**.8268 ± .0909**	167.06 (2)	0.93424	29/2/35
RUS-Boost	Baseline	.8557 ± .0774	175.59 (5)	0.70641	22/2/40
	BB-Imb	**.8628 ± .0599**	148.67 (1)	********	-/-/-
	MDM-Imb	.8595 ± .0632	173.09 (4)	0.70641	23/4/39
	Comp	.8594 ± .0685	166.53 (2)	0.70641	29/3/34
	Kappa	.7789 ± .0811	355.18 (6)	0.00000$*$	0/2/64
	RE-GM	.8581 ± .0693	171.94 (3)	0.70641	28/3/35
EUS-Boost	Baseline	.8600 ± .0727	176.70 (3)	0.12924	22/4/40
	BB-Imb	**.8642 ± .0612**	139.88 (1)	********	-/-/-
	MDM-Imb	.8476 ± .0835	193.22 (5)	0.02970$*$	13/3/50
	Comp	.8509 ± .0855	172.28 (2)	0.12924	32/3/31
	Kappa	.7939 ± .0868	326.09 (6)	0.00000$*$	2/2/62
	RE-GM	.8499 ± .0873	182.80 (4)	0.09373+	28/3/35
Easy-Ensemble	Baseline	.8596 ± .0591	163.75 (2)	0.85009	22/2/42
	BB-Imb	**.8618 ± .0626**	156.42 (1)	********	-/-/-
	MDM-Imb	.8612 ± .0639	172.31 (3)	0.85009	32/4/30
	Comp	.8272 ± .0850	205.98 (5)	0.03860$*$	27/4/35
	Kappa	.8300 ± .0660	272.48 (6)	0.00000$*$	13/4/49
	RE-GM	.8263 ± .0870	220.03 (4)	0.00564$*$	28/4/34

3. In addition to *BB-Imb*, the remaining performance-based metrics *Comp* and *RE-GM*, present a robust behavior in all cases. It is not straightforward to highlight one scheme over the other, as they achieve a similar average ranking among all ensemble algorithms.
4. Finally, *Kappa* ordering model achieves the best result for SMOTE-Bagging. However, in the remaining cases, it is always outperformed by the corresponding control method in the statistical comparison, i.e., *BB-Imb*, *Comp* and *RE-GM*. Hence, it should be highlighted that metrics based on somehow measuring performance of the sub-ensemble work better in ensembles for class imbalance problem, but in the case of SMOTE-Bagging they may lead to greater overfitting (to which also leads SMOTE). In this case, promoting diversity using Kappa helps in improving performance.

The former analysis is independent of the metric of performance selected, thus stressing the robustness of these findings. For the subsequent analysis where we will compare the best ensemble methods among them, we will select *Kappa* scheme for SMOTE-Bagging, *RE-GM* for Under-Bagging and SMOTE-Boost, and *BB-Imb* for Roughly-Bagging, RUS-Boost, EUS-Boost and Easy-Ensemble, according to the results obtained in this section.

We acknowledge that in some cases there were different top schemes but, for the sake of simplifying the experimental analysis, we have selected just the one with the highest GM value as a representative element for the final intra-comparison.

5.3. Intra-family comparison for ensemble ordering-based pruning in imbalanced classification

Finally, we aim to investigate the behavior of the best pruning schemes for each ensemble algorithm in contrast with the remaining ones. This analysis will lead us to discover whether there is a combination that stands out among all approaches. In accordance with the former, Tables 8 and 9 show the average performance in AUC and GM for all algorithms of comparison in the 66 selected problems.

As we did in the previous experimental analysis, we carry out a statistical study in order to determine whether significant differences are found among the selected methods. Therefore, we summarize these results in Tables 10 and 11 using AUC and GM metrics, respectively. These tables show the average performance results, together with the average ranking (computed with the Friedman aligned tests) and the APVs obtained by the Holm test. Again, we explicitly stress whether there are statistical differences with a significance level higher than 95% (symbol $*$) or 90% (symbol $+$), and the number of wins,l ties, and loses versus the control algorithm.

Among all algorithms for comparison, Under-Bagging with *RE-GM* and Roughly-Based Bagging with *BB-Imb* pruning approaches are highlighted as the best methods overall. Both achieve the highest average results and ranking and, particularly for Under-Bagging, it statistically outperforms all remaining methodologies. As in previous analysis, the conclusions extracted from the statistical comparison are independent of the performance metric employed.

It should be mentioned that in the previous review [23]. Under-Bagging was highlighted as one of the best approaches together with SMOTE-Bagging and RUS-Boost. It is interesting to observe that the introduction of ensemble pruning techniques into this framework has allowed Under-Bagging to make a difference with respect to previous approaches, even with respect to more recent ones as EUS-Boost. In the case of Under-Bagging, every classifier is trained in the same way after randomly undersampling the original data-set. Hence, pruning allows one to select from those classifiers the ones which better complement each other, increasing the final performance due to the elimination of classifiers that could reduce diversity and performance as consequence. Furthermore, notice that the number of classifiers used is reduced from 40 to 21. SMOTE-Bagging is also benefited by a simplification of the ensemble, but its final precision was not as high as other approaches (mainly in GM), perhaps being affected by the overfitting introduced by SMOTE. Otherwise, boosting-based approaches use more classifiers than in the original models (21 vs. 10), although they are also improved. In this case we must point out that the selection problem is different from Under-Bagging, where all classifiers were independently learned. In the former case, they are correlated due to the boosting learning procedure, making the possible improvement to be limited, as shown by the experimental results.

6. Lessons learned and future work

In this paper we have stressed the good properties of ordering-based pruning approaches for ensemble learning. However, some of the metrics that guide the former process are not suited for classification with imbalanced datasets. In this way, we have proposed several new schemes to provide adapted solutions for this context, being able to achieve high quality results.

We have carried out our experimental analysis in three different stages so as to contrast our initial hypothesis: (1) checking the validity of the adapted pruning metrics versus the standard ones; (2) stressing the best synergy between the proposed pruning metrics and the different selected ensemble approaches for imbalanced classification; and (3) emphasizing the best method overall, in order to provide additional support to the goodness of this new proposed methodology.

Therefore, from the development of this thorough study, we may emphasize 4 important lessons learned that may help other researchers to understand the intrinsic features of this framework:

1. Pruning mechanism is positively biased when using the new adapted heuristics metrics for imbalanced classification. This superior performance was clearly established by the results achieved from the statistical tests.
2. In all cases, the use of the imbalanced pruning metrics allows the enhancement of the baseline ensemble approaches. Furthermore, in the case of Under-Bagging the ordering-based pruning allows one to even find statistical differences among the results. Finally, we must stress also the good behavior of Roughly-Based Bagging which performs similarly to Under-Bagging, since both share common characteristics regarding their working procedure.
3. Three heuristic metrics have excelled over the rest: (1) *BB-Imb*, (2) *Comp*, and (3) *RE-GM*. They have shown a clear synergy with every ensemble learning approach.
4. Finally, we must stress that the main quality of the "a-posteriori" pruning is that it allows carrying out a supervised selection from "randomness", only considering those classifiers that present a better cooperation. This fact is clearly observed in the Under-Bagging technique, which achieves the highest benefit from this methodology. In this approach, all classifiers of the ensemble are the most independent among them, as they are built in a random way without taking into account the previous classifiers, such as in boosting. Therefore, experimental results show that this scheme outperforms all results from the state-of-the-art in ensemble learning for imbalanced classification.

This study opens the way for interesting future work on the topic under different perspectives. A first approach is the analysis of additional ways for enhancing the diversity in the ensemble construction, and thus to obtain a global system with higher quality. On this account, several authors have already proposed some alternatives that must be studied in depth [36,69].

Table 8
Test Results and standard deviation for the selected ordering-based pruning schemes using the AUC metric. From the leftmost to the rightmost column the name of the dataset, IR and ensemble classifiers are shown, i.e. SMOTE-Bagging (SMT-Bag), Under-Bagging (U-Bag), Roughly-Based Bagging (RB-Bag), SMOTE-Boost (SMT-B), RUS-Boost (RUS-B), EUS-Boost (EUS-B), and Easy-Ensemble (Easy).

Dataset	IR	SMT-Bag_Kappa	U-Bag_RE-GM	RB-Bag_BB-Imb	SMT-B_RE-GM	RUS-B_BB-Imb	EUS-B_BB-Imb	Easy_BB-Imb
glass1	1.82	.7848 ± .0471	.7961 ± .0498	.7873 ± .0468	.8031 ± .0602	**.8157 ± .0499**	.8046 ± .0522	.8108 ± .0524
ecoli0vs1	1.86	**.9783 ± .020**3	.9774 ± .0200	.9753 ± .0199	.9761 ± .0212	.9743 ± .0218	.9749 ± .0198	.9742 ± .0212
wisconsin	1.86	**.9722 ± .0116**	.9702 ± .0115	.9702 ± .0123	.9668 ± .0128	.9713 ± .0114	.9699 ± .0121	.9714 ± .0103
pima	1.90	.7515 ± .0270	**.7523 ± .0263**	.7532 ± .0284	.7329 ± .0258	.7321 ± .0275	.7378 ± .0292	.7507 ± .0270
iris0	2.00	.9880 ± .0214	**.9900 ± .0200**	**.9900 ± .0200**	**.9900 ± .0200**	**.9900 ± .0200**	**.9900 ± .0200**	**.9900 ± .0200**
glass0	2.06	.8377 ± .0455	.8467 ± .0423	.8461 ± .0401	.8461 ± .0610	**.8608 ± .0417**	.8502 ± .0413	.8490 ± .0415
yeast1	2.46	.7206 ± .0212	**.7311 ± .0227**	.7307 ± .0212	.7186 ± .0239	.7143 ± .0235	.7155 ± .0228	.7301 ± .0219
vehicle2	2.52	.9707 ± .0133	.9735 ± .0111	.9750 ± .0126	**.9830 ± .0116**	.9813 ± .0095	.9813 ± .0105	.9797 ± .0103
vehicle1	2.52	.7638 ± .0305	.7976 ± .0297	.7953 ± .0265	.7648 ± .0350	.7987 ± .0294	**.8055 ± .0254**	.8000 ± .0274
vehicle3	2.52	.7598 ± .0326	.7989 ± .0220	.8001 ± .0252	.7515 ± .0262	.7945 ± .0290	**.8055 ± .0243**	.8041 ± .0220
haberman	2.68	**.6622 ± .0386**	.6586 ± .0379	.6552 ± .0372	.6487 ± .0438	.6346 ± .0537	.6353 ± .0522	.6429 ± .0525
glass0123vs456	3.19	**.9477 ± .0298**	.9376 ± .0332	.9420 ± .0310	.9168 ± .0543	.9467 ± .0366	.9463 ± .0387	.9443 ± .0340
vehicle0	3.23	.9605 ± .0136	.9551 ± .0151	.9588 ± .0128	**.9686 ± .0161**	.9661 ± .0118	.9659 ± .0130	.9622 ± .0120
ecoli1	3.36	.9012 ± .0255	.9044 ± .0287	.9094 ± .0286	.8710 ± .0393	.9071 ± .0323	.8995 ± .0342	**.9103 ± .0321**
newthyroid2	4.92	.9668 ± .0315	.9633 ± .0379	.9676 ± .0338	**.9833 ± .0233**	.9812 ± .0253	.9712 ± .0336	.9682 ± .0333
newthyroid1	5.14	.9655 ± .0395	.9500 ± .0577	.9505 ± .0542	**.9817 ± .0293**	.9685 ± .0423	.9617 ± .0504	.9584 ± .0546
ecoli2	5.46	.9107 ± .0484	.9065 ± .0495	.8983 ± .0500	**.9142 ± .0473**	.8942 ± .0468	.9014 ± .0431	.8994 ± .0467
segment0	6.01	.9904 ± .0086	.9899 ± .0080	.9890 ± .0083	.9938 ± .0079	**.9962 ± .0048**	.9959 ± .0050	.9945 ± .0064
glass6	6.38	**.9275 ± .0485**	.9264 ± .0482	.9247 ± .0473	.9233 ± .0619	.9252 ± .0483	.9255 ± .0449	.9231 ± .0485
yeast3	8.11	.9318 ± .0237	**.9343 ± .022**6	.9300 ± .0243	.8863 ± .0319	.9298 ± .0219	.9297 ± .0214	.9322 ± .0230
ecoli3	8.19	.8638 ± .0702	.8723 ± .0675	.8675 ± .0690	.8419 ± .0927	.8687 ± .0638	.8715 ± .0645	**.8810 ± .0534**
pageblocks0	8.77	.9574 ± .0115	.9633 ± .0093	.9629 ± .0087	.9512 ± .0108	**.9640 ± .0093**	.9633 ± .0091	.9639 ± .0096
ecoli034vs5	9.00	**.9257 ± .0778**	.9102 ± .0762	.9166 ± .0815	.8986 ± .0919	.9020 ± .0835	.9033 ± .0899	.8993 ± .0819
yeast2vs4	9.08	.9314 ± .0341	.9408 ± .0371	.9486 ± .0318	.8805 ± .0641	**.9458 ± .0348**	.9452 ± .0339	.9455 ± .0300
ecoli067vs35	9.09	.8588 ± .0795	**.8649 ± .0745**	.8570 ± .0701	.8615 ± .0838	.8612 ± .0746	.8622 ± .0802	.8601 ± .0779
ecoli0234vs5	9.10	**.9035 ± .0960**	.8948 ± .0864	.9037 ± .0830	.8928 ± .0931	.8836 ± .0931	.8910 ± .0951	.8902 ± .0943
glass015vs2	9.12	.7130 ± .1596	.7201 ± .1351	.7534 ± .1390	.7221 ± .1106	.7309 ± .1198	**.7629 ± .1257**	.6945 ± .1336
yeast0359vs78	9.12	.7241 ± .0644	.7345 ± .0621	.7388 ± .0545	.6603 ± .0595	.7373 ± .0484	**.7382 ± .0542**	.7266 ± .0523
yeast02579vs368	9.14	.8036 ± .0356	**.8171 ± .0292**	.8084 ± .0235	.7879 ± .0390	.8041 ± .0382	.7931 ± .0370	.8106 ± .0294
yeast0256vs3789	9.14	**.9155 ± .0232**	.9129 ± .0157	.9108 ± .0162	.8963 ± .0296	.9040 ± .0219	.9055 ± .0218	.9127 ± .0151
ecoli046vs5	9.15	**.9185 ± .0740**	.9103 ± .0681	.9172 ± .0718	.8950 ± .0740	.8956 ± .0805	.8954 ± .0908	.8878 ± .0735
ecoli01vs235	9.17	.8960 ± .0768	**.9058 ± .0668**	.9076 ± .0662	.8720 ± .0804	.8777 ± .0678	.8864 ± .0678	.8825 ± .0700
ecoli0267vs35	9.18	.8521 ± .0868	.8638 ± .0881	.8572 ± .0898	.8650 ± .0877	**.8731 ± .0923**	.8652 ± .0961	.8690 ± .0917
glass04vs5	9.22	.9865 ± .0172	.9940 ± .0121	.9940 ± .0121	.9848 ± .0338	**.9940 ± .0121**	**.9940 ± .0121**	**.9940 ± .0121**
ecoli0346vs5	9.25	**.9146 ± .0889**	.8934 ± .0860	.9039 ± .0955	.8874 ± .1136	.8713 ± .0888	.8818 ± .0833	.8781 ± .0938
ecoli0347vs56	9.28	.8696 ± .0862	.8973 ± .0737	.8933 ± .0772	**.8990 ± .0739**	.8739 ± .0754	.8799 ± .0791	.8871 ± .0780
yeast05679vs4	9.35	.8282 ± .0510	.8248 ± .0523	.8221 ± .0480	.7894 ± .0658	**.8294 ± .0503**	.8276 ± .0476	.8177 ± .0470
ecoli067vs5	10.00	.8903 ± .0509	.8921 ± .0721	.8907 ± .0602	**.8981 ± .0705**	.8682 ± .0692	.8660 ± .0670	.8910 ± .0709
vowel0	10.10	.9851 ± .0121	.9687 ± .0215	.9682 ± .0208	**.9901 ± .0154**	.9823 ± .0145	.9854 ± .0128	.9795 ± .0170
glass016vs2	10.29	.7139 ± .1684	.7348 ± .1447	.7173 ± .1553	**.7435 ± .1318**	.7202 ± .1435	.7346 ± .1300	.7046 ± .1496
glass2	10.39	.7595 ± .1117	.7582 ± .1145	.7692 ± .1231	.7176 ± .1197	**.7729 ± .1039**	.7684 ± .1080	.7330 ± .1167
ecoli0147vs2356	10.59	.8563 ± .0758	.8704 ± .0773	.8591 ± .0754	**.8783 ± .0797**	.8291 ± .0746	.8466 ± .0723	.8550 ± .0795
led7digit02456789vs1	10.97	**.8565 ± .0877**	.8336 ± .0736	.8319 ± .0742	.7985 ± .1409	.8342 ± .0732	.8271 ± .0650	.8381 ± .0795
ecoli01vs5	11.00	**.9179 ± .0728**	.9144 ± .0846	.9252 ± .0849	.8976 ± .0819	.9063 ± .0908	.9048 ± .0896	.8992 ± .0924
glass06vs5	11.00	.9825 ± .0188	**.9922 ± .0136**	.9950 ± .0100	.9890 ± .0169	.9321 ± .0629	.9486 ± .0403	.9319 ± .0447
glass0146vs2	11.06	.7046 ± .1519	**.7558 ± .1386**	.7260 ± .1601	.7329 ± .1245	.7334 ± .1419	.7366 ± .1350	.7144 ± .1427
ecoli0147vs56	12.28	.8847 ± .0868	.8950 ± .0697	.8977 ± .0658	**.9121 ± .0663**	.8780 ± .0668	.8837 ± .0737	.8918 ± .0665
cleveland0vs4	12.62	.8364 ± .1298	**.8673 ± .1061**	.8446 ± .1261	.7744 ± .1384	.8282 ± .1325	.8340 ± .1276	.8349 ± .1385
ecoli0146vs5	13.00	.9022 ± .0738	**.9089 ± .0712**	.9248 ± .0705	.8976 ± .0779	.8912 ± .0702	.8842 ± .0738	.9027 ± .0726
ecoli4	13.84	.9337 ± .0660	.9308 ± .0705	.9325 ± .0730	.9133 ± .1018	.9349 ± .0583	**.9387 ± .0550**	.9345 ± .0612
shuttle0vs4	13.87	.9998 ± .0007	**1.000 ± .0000**	**1.000 ± .0000**	.9999 ± .0004	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**
yeast1vs7	13.87	.7482 ± .0921	**.7741 ± .0759**	.7816 ± .0711	.7305 ± .0765	.7438 ± .0724	.7490 ± .0633	.7644 ± .0750
glass4	15.47	**.9283 ± .0728**	.9115 ± .1092	.8901 ± .0995	.8404 ± .1472	.8835 ± .0850	.8867 ± .0743	.8866 ± .0746
pageblocks13vs4	15.85	.9919 ± .0074	**.9941 ± .0047**	.9945 ± .0115	.9928 ± .0186	.9846 ± .0182	.9882 ± .0113	.9694 ± .0196
abalone9vs18	16.68	.7192 ± .0839	.7440 ± .0734	.7418 ± .0673	.6946 ± .0761	.7405 ± .0637	.7337 ± .0577	**.7452 ± .0631**
glass016vs5	19.44	**.9779 ± .0341**	.9765 ± .0202	.9730 ± .0237	.9142 ± .1258	.9417 ± .0250	.9677 ± .0245	.9441 ± .0114
shuttle2vs4	20.50	.9990 ± .0051	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	.9998 ± .0019
yeast1458vs7	22.10	.6434 ± .0975	**.6464 ± .0821**	.6407 ± .0909	.5653 ± .0714	.6208 ± .0970	.6158 ± .0973	.6172 ± .1010
glass5	22.81	**.9697 ± .0575**	.9667 ± .0221	.9665 ± .0236	.9321 ± .1097	.9526 ± .0199	.9648 ± .0238	.9494 ± .0141
yeast2vs8	23.10	**.8001 ± .0856**	.7726 ± .0609	.7540 ± .0737	.7706 ± .0938	.7643 ± .0782	.7759 ± .0808	.7492 ± .0674
yeast4	28.41	.8334 ± .0558	**.8518 ± .0490**	.8506 ± .0461	.7405 ± .0705	.8465 ± .0446	.8487 ± .0432	.8409 ± .0412
yeast1289vs7	30.56	.7059 ± .0860	**.7295 ± .0742**	.7365 ± .0743	.6574 ± .0798	.7149 ± .0757	.7203 ± .0748	.7184 ± .0839
yeast5	32.78	.9630 ± .0322	.9639 ± .0323	.9654 ± .0311	.9001 ± .0575	**.9678 ± .0270**	.9678 ± .0276	.9633 ± .0238
yeast6	39.15	.8420 ± .0525	**.8715 ± .0422**	.8589 ± .0451	.7823 ± .0740	.8487 ± .0341	.8469 ± .0375	.8682 ± .0398
ecoli0137vs26	39.15	.8401 ± .1674	.8492 ± .1620	.7920 ± .1455	**.8575 ± .1739**	.8377 ± .1026	.8226 ± .1523	.8347 ± .1557
abalone19	128.87	.5570 ± .0669	.7082 ± .0967	.7244 ± .0849	.5387 ± .0400	**.7208 ± .0757**	.7206 ± .0877	.7202 ± .0812
Average	–	.8681 ± .0578	**.8752 ± .0545**	.8737 ± .0546	.8525 ± .0642	.8679 ± .0540	.8697 ± .0544	.8678 ± .0549

Table 9
Test Results and standard deviation for the selected ordering-based pruning schemes using the GM metric. From the leftmost to the rightmost column the name of the dataset, IR and ensemble classifiers are shown, i.e. SMOTE-Bagging (SMT-Bag), Under-Bagging (U-Bag), Roughly-Based Bagging (RB-Bag), SMOTE-Boost (SMT-B), RUS-Boost (RUS-B), EUS-Boost (EUS-B), and Easy-Ensemble (Easy).

Dataset	IR	SMT-Bag_Kappa	U-Bag_RE-GM	RB-Bag_BB-Imb	SMT-B_RE-GM	RUS-B_BB-Imb	EUS-B_BB-Imb	Easy_BB-Imb
glass1	1.82	.7814 ± .0484	.7939 ± .0500	.7845 ± .0471	.7920 ± .0697	**.8123 ± .0504**	.8011 ± .0525	.8080 ± .0525
ecoli0vs1	1.86	.7700 ± .3090	.7897 ± .2926	**.9750 ± .0201**	.7867 ± .3174	.8177 ± .1503	.7705 ± .2704	.7703 ± .2945
wisconsin	1.86	**.9721 ± .0116**	.9700 ± .0116	.9701 ± .0124	.9666 ± .0128	.9712 ± .0115	.9698 ± .0121	.9713 ± .0103
pima	1.90	.7498 ± .0275	.7513 ± .0265	**.7519 ± .0285**	.7295 ± .0280	.7304 ± .0275	.7353 ± .0294	.7489 ± .0270
iris0	2.00	.9877 ± .0219	**.9897 ± .0205**	**.9897 ± .0205**	**.9897 ± .0205**	**.9897 ± .0205**	**.9897 ± .0205**	**.9897 ± .0205**
glass0	2.06	**.9469 ± .0303**	.9366 ± .0338	.8439 ± .0395	.9142 ± .0582	.9458 ± .0378	.9454 ± .0397	.9434 ± .0352
yeast1	2.46	.6591 ± .1426	.7076 ± .0941	**.7299 ± .0211**	.5433 ± .1823	.7088 ± .0748	.7142 ± .0750	.7057 ± .0970
vehicle2	2.52	.9706 ± .0134	.9734 ± .0111	.9749 ± .0127	**.9830 ± .0116**	.9813 ± .0095	.9812 ± .0105	.9796 ± .0103
vehicle1	2.52	.7612 ± .0323	.7941 ± .0287	.7923 ± .0253	.7611 ± .0386	.7961 ± .0288	**.8013 ± .0248**	.7955 ± .0268
vehicle3	2.52	.7574 ± .0349	.7953 ± .0214	.7966 ± .0241	.7471 ± .0287	.7923 ± .0283	**.8008 ± .0236**	.7987 ± .0218
haberman	2.68	**.6585 ± .0404**	.6528 ± .0404	.6517 ± .0376	.6391 ± .0513	.6288 ± .0558	.6270 ± .0543	.6371 ± .0523
glass0123vs456	3.19	.6536 ± .2321	.7363 ± .1770	**.9410 ± .0316**	.6760 ± .2010	.7076 ± .1781	.7114 ± .1694	.6921 ± .1760
vehicle0	3.23	.9601 ± .0136	.9547 ± .0151	.9584 ± .0128	**.9685 ± .0162**	.9658 ± .0118	.9656 ± .0130	.9618 ± .0120
ecoli1	3.36	.9002 ± .0250	.9035 ± .0286	.9086 ± .0283	.8680 ± .0419	.9055 ± .0323	.8984 ± .0338	**.9093 ± .0318**
newthyroid2	4.92	.9661 ± .0323	.9618 ± .0404	.9664 ± .0361	**.9830 ± .0240**	.9807 ± .0264	.9702 ± .0358	.9671 ± .0353
newthyroid1	5.14	.9650 ± .0400	.9471 ± .0628	.9480 ± .0583	**.9815 ± .0295**	.9670 ± .0456	.9600 ± .0538	.9563 ± .0591
ecoli2	5.46	.9095 ± .0497	.9044 ± .0518	.8956 ± .0532	**.9116 ± .0499**	.8923 ± .0491	.8995 ± .0450	.8972 ± .0490
segment0	6.01	.9904 ± .0087	.9898 ± .0080	.9889 ± .0083	.9937 ± .0080	**.9961 ± .0048**	.9959 ± .0051	.9944 ± .0064
glass6	6.38	**.9256 ± .0502**	.9245 ± .0499	.9228 ± .0490	.9191 ± .0683	.9235 ± .0498	.9239 ± .0459	.9212 ± .0501
yeast3	8.11	.9314 ± .0238	**.9340 ± .0228**	.9296 ± .0245	.8824 ± .0347	.9296 ± .0219	.9294 ± .0214	.9318 ± .0231
ecoli3	8.19	.8592 ± .0757	.8682 ± .0717	.8629 ± .0737	.8283 ± .1084	.8647 ± .0672	.8672 ± .0691	**.8779 ± .0553**
pageblocks0	8.77	.9919 ± .0075	**.9941 ± .0047**	.9628 ± .0087	.9926 ± .0193	.9843 ± .0188	.9880 ± .0116	.9687 ± .0204
ecoli034vs5	9.00	.8618 ± .0973	.8923 ± .0810	**.9114 ± .0905**	.8937 ± .0810	.8687 ± .0822	.8747 ± .0869	.8823 ± .0854
yeast2vs4	9.08	.7761 ± .1108	.7439 ± .0747	**.9476 ± .0328**	.7284 ± .1298	.7574 ± .0825	.7669 ± .0879	.7317 ± .0777
ecoli067vs35	9.09	.8857 ± .0537	.8847 ± .0859	.8484 ± .0804	**.8906 ± .0816**	.8622 ± .0782	.8603 ± .0756	.8852 ± .0809
ecoli0234vs5	9.10	.8444 ± .0941	.8545 ± .0975	**.8966 ± .0938**	.8548 ± .0976	.8666 ± .0996	.8580 ± .1038	.8611 ± .1003
glass015vs2	9.12	.6452 ± .2936	.7024 ± .2172	**.7294 ± .1923**	.6771 ± .2449	.6930 ± .1850	.7082 ± .1682	.6769 ± .1895
yeast0359vs78	9.12	.7133 ± .0758	.7217 ± .0737	**.7350 ± .0578**	.5942 ± .0968	.7332 ± .0474	.7342 ± .0563	.7217 ± .0563
yeast02579vs368	9.14	.7947 ± .0413	**.8126 ± .0318**	.8039 ± .0259	.7658 ± .0488	.8013 ± .0405	.7891 ± .0392	.8042 ± .0334
yeast0256vs3789	9.14	**.9143 ± .0243**	.9116 ± .0162	.9099 ± .0168	.8923 ± .0322	.9034 ± .0225	.9049 ± .0225	.9119 ± .0156
ecoli046vs5	9.15	.8517 ± .0881	.8565 ± .0845	**.9126 ± .0795**	.8503 ± .0965	.8549 ± .0825	.8564 ± .0868	.8522 ± .0873
ecoli01vs235	9.17	.8959 ± .0819	**.9046 ± .0769**	.9039 ± .0711	.8895 ± .0898	.8870 ± .0742	.8792 ± .0789	.8987 ± .0774
ecoli0267vs35	9.18	**.9074 ± .1035**	.8856 ± .0998	.8485 ± .0984	.8728 ± .1398	.8631 ± .0987	.8747 ± .0921	.8684 ± .1089
glass04vs5	9.22	.9822 ± .0193	.9921 ± .0139	**.9939 ± .0123**	.9888 ± .0173	.9290 ± .0654	.9463 ± .0428	.9284 ± .0483
ecoli0346vs5	9.25	**.9210 ± .0860**	.9047 ± .0850	.8959 ± .1104	.8890 ± .1056	.8957 ± .0928	.8966 ± .1004	.8942 ± .0890
ecoli0347vs56	9.28	**.9130 ± .0827**	.9059 ± .0745	.8878 ± .0868	.8871 ± .0837	.8903 ± .0894	.8867 ± .1088	.8824 ± .0818
yeast05679vs4	9.35	.8255 ± .0529	.8210 ± .0557	.8200 ± .0484	.7708 ± .0840	**.8269 ± .0501**	.8258 ± .0475	.8159 ± .0480
ecoli067vs5	10.00	**.9780 ± .0205**	.9772 ± .0202	.8854 ± .0675	.9758 ± .0215	.9739 ± .0221	.9746 ± .0201	.9739 ± .0213
vowel0	10.10	.9850 ± .0122	.9684 ± .0217	.9680 ± .0210	**.9899 ± .0157**	.9822 ± .0147	.9853 ± .0129	.9794 ± .0171
glass016vs2	10.29	**.9769 ± .0389**	.9759 ± .0209	.6820 ± .2269	.8918 ± .1871	.9395 ± .0272	.9668 ± .0254	.9424 ± .0121
glass2	10.39	.7422 ± .1404	.7468 ± .1270	.7585 ± .1319	.6569 ± .1955	**.7592 ± .1044**	.7556 ± .1085	.7165 ± .1322
ecoli0147vs2356	10.59	**.9136 ± .0788**	.9102 ± .0912	.8528 ± .0872	.8903 ± .0909	.9019 ± .0972	.9001 ± .0965	.8936 ± .1020
led7digit02456789vs1	10.97	**.8450 ± .0995**	.8209 ± .0844	.8194 ± .0850	.7600 ± .2204	.8255 ± .0800	.8191 ± .0705	.8261 ± .0903
ecoli01vs5	11.00	.8490 ± .0851	.8632 ± .0871	**.9212 ± .0919**	.8686 ± .0910	.8223 ± .0814	.8408 ± .0801	.8476 ± .0939
glass06vs5	11.00	.8349 ± .0447	.8437 ± .0416	**.9949 ± .0101**	.8437 ± .0630	.8565 ± .0420	.8476 ± .0409	.8458 ± .0407
glass0146vs2	11.06	.6646 ± .2468	.6889 ± .1986	.6870 ± .2392	.6717 ± .1925	.7001 ± .1717	**.7344 ± .1787**	.6676 ± .1757
ecoli0147vs56	12.28	**.8942 ± .1101**	.8876 ± .0967	.8941 ± .0700	.8825 ± .1070	.8751 ± .1043	.8825 ± .1071	.8817 ± .1059
cleveland0vs4	12.62	.8029 ± .1980	**.8583 ± .1183**	.8168 ± .2001	.7155 ± .2213	.8094 ± .1773	.8189 ± .1623	.8195 ± .1752
ecoli0146vs5	13.00	.8919 ± .0813	.9025 ± .0704	**.9212 ± .0764**	.8611 ± .0925	.8722 ± .0735	.8816 ± .0724	.8774 ± .0758
ecoli4	13.84	.9302 ± .0714	.9271 ± .0768	.9286 ± .0797	.9020 ± .1276	.9323 ± .0623	**.9366 ± .0577**	.9320 ± .0650
shuttle0vs4	13.87	.9998 ± .0007	**1.000 ± .0000**	**1.000 ± .0000**	.9999 ± .0004	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**
yeast1vs7	13.87	.5839 ± .1457	.6225 ± .0994	**.7762 ± .0742**	.3232 ± .2327	.6103 ± .0968	.6059 ± .0987	.6056 ± .1060
glass4	15.47	**.9241 ± .0809**	.8915 ± .1822	.8771 ± .1306	.7925 ± .2582	.8735 ± .1134	.8800 ± .0799	.8811 ± .0778
pageblocks13vs4	15.85	.9573 ± .0117	.9633 ± .0093	**.9944 ± .0119**	.9510 ± .0110	.9640 ± .0093	.9632 ± .0091	.9638 ± .0096
abalone9vs18	16.68	.6931 ± .1096	.7320 ± .0859	.7357 ± .0765	.6390 ± .1095	.7366 ± .0668	.7302 ± .0588	**.7410 ± .0680**
glass016vs5	19.44	.9863 ± .0175	**.9939 ± .0123**	.9723 ± .0246	.9839 ± .0388	.9939 ± .0123	.9939 ± .0123	.9939 ± .0123
shuttle2vs4	20.50	.9990 ± .0052	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	**1.000 ± .0000**	.9998 ± .0019
yeast1458vs7	22.10	.9303 ± .0347	.9396 ± .0387	.6295 ± .0997	.8737 ± .0731	**.9448 ± .0357**	.9441 ± .0350	.9443 ± .0313
glass5	22.81	**.9673 ± .0664**	.9659 ± .0230	.9656 ± .0245	.9221 ± .1270	.9512 ± .0209	.9638 ± .0249	.9479 ± .0148
yeast2vs8	23.10	.7151 ± .0227	.7306 ± .0226	**.7389 ± .0838**	.7160 ± .0250	.7124 ± .0245	.7125 ± .0251	.7290 ± .0218
yeast4	28.41	.8278 ± .0619	**.8502 ± .0498**	.8485 ± .0448	.6982 ± .1003	.8440 ± .0431	.8465 ± .0418	.8375 ± .0395
yeast1289vs7	30.56	.7332 ± .1105	**.7647 ± .0870**	.7270 ± .0808	.6883 ± .1114	.7365 ± .0715	.7429 ± .0619	.7584 ± .0781
yeast5	32.78	.9623 ± .0331	.9632 ± .0335	.9648 ± .0321	.8936 ± .0654	**.9673 ± .0279**	.9672 ± .0285	.9628 ± .0241
yeast6	39.15	.8313 ± .0617	**.8688 ± .0462**	.8570 ± .0472	.7496 ± .0995	.8476 ± .0353	.8456 ± .0386	.8671 ± .0411
ecoli0137vs26	39.15	.8765 ± .0957	.8907 ± .0748	.7209 ± .2861	**.9069 ± .0728**	.8747 ± .0706	.8796 ± .0796	.8888 ± .0698
abalone19	128.87	.3139 ± .2260	.6956 ± .1137	**.7154 ± .0976**	.2177 ± .1972	.7116 ± .0759	.7107 ± .0984	.7120 ± .0810
Average	–	.8547 ± .0741	**.8683 ± .0646**	.8673 ± .0647	.8268 ± .0909	.8628 ± .0599	.8642 ± .0612	.8618 ± .0626

Table 10
Average results (AUC), Ranks (Friedman Aligned test) and APVs (Holm test) for **the state-of-the-art in ensemble learning for imbalanced classification**. Control method is pointed out with asterisks. Symbol ∗ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the control algorithm.

Ensemble Model	AUC Test	Ranking (Friedman Aligned)	APV (Holm test)	W/T/L
SMOTE-Bagging-Kappa	.8681 ± .0578	231.02 (3)	0.01015∗	25/0/41
Under-Bagging-RE-GM	**.8752 ± .0545**	165.89 (1)	********	-/-/-
Roughly-Bagging-BB-Imb	.8737 ± .0546	183.4 (2)	0.45108	28/4/34
SMOTE-Boost-RE-GM	.8525 ± .0642	311.16 (7)	0.00000∗	16/2/48
RUS-Boost-BB-Imb	.8679 ± .0540	246.84 (6)	0.00248∗	23/4/39
EUS-Boost-BB-Imb	.8697 ± .0544	236.78 (4)	0.00686∗	21/4/41
Easy-Ensemble-BB-Imb	.8678 ± .0549	245.42 (5)	0.00249∗	21/3/42

Table 11
Average results (GM), Ranks (Friedman Aligned test) and APVs (Holm test) for **the state-of-the-art in ensemble learning for imbalanced classification**. Control method is pointed out with asterisks. Symbol ∗ implies significant differences at 95%, whereas symbol + sets the confidence degree at 90%. Wins/Ties/Loses values are computed with respect to the control algorithm.

Ensemble Model	GM Test	Ranking (Friedman Aligned)	APV (Holm test)	W/T/L
SMOTE-Bagging-Kappa	.8547 ± .0741	246.97(6)	0.00862∗	25/0/41
Under-Bagging-RE-GM	**.8683 ± .0646**	174.13 (1)	********	-/-/-
Roughly-Bagging-BB-Imb	.8673 ± .0647	185.43 (2)	0.62673	30/3/33
SMOTE-Boost-RE-GM	.8268 ± .0909	337.57 (7)	0.00000∗	13/2/51
RUS-Boost-BB-Imb	.8628 ± .0599	224.25 (4)	0.09312+	27/4/35
EUS-Boost-BB-Imb	.8642 ± .0612	218.27(3)	0.11512	23/4/39
Easy-Ensemble-BB-Imb	.8618 ± .0626	233.89 (5)	0.04054∗	23/3/40

Another straightforward step is to analyze the synergy of ensembles of classifiers with novel approaches on instance generation for the rebalancing of the training set [81]. Additionally, there are new imbalanced classification problems that must be addressed, such as those based on Multi-instance [70] and Multi-label learning [11].

Furthermore, ensembles of classifiers have been traditionally used for online learning from Data Streams [53,54]. The problem of imbalanced data is also present in this framework, and novel approaches have been developed to address the former issue with ensemble learning [27,71]. Since our proposed approach is based on improving the efficiency of the system, we consider that it must be a good point of reference for further work on this topic.

Finally, we cannot forget about Big Data applications [15,20], being maybe the hottest topic in the research community. As in the standard case study, ensembles of classifiers are a very valuable tool for addressing the imbalanced classification problem in a Big Data scenario [16,80]. Focusing on developing novel algorithms based on the MapReduce scheme must be regarded as a profitable research line.

7. Concluding remarks

Ensembles of classifiers have shown very good properties for addressing the problem of imbalanced classification. They work in line with baseline solutions for this task such as preprocessing or cost-sensitive learning. However, we must face the problem of setting the optimal number of classifiers that better suits the problem. A low number may be insufficient to reach a quality solution, and a high number may involve conflicts in the final decision process or over-fitting, leading to erroneous outputs.

The answer to the former problem consists of carrying out a pruning (actually a selection) from a pool with a high number of classifiers. More specifically, classifiers can be added or not to the final set according to the optimization of a given quality measure, being the final aim the improvement of the global behavior of the system. In this sense, this technique receive the name of ordering-based pruning.

In this work, we have proposed several novel ordering-based ensemble pruning metrics to work in the context of imbalanced classification. To check the validity of our approach, we have developed an exhaustive experimental analysis over a large number of benchmark datasets considering the best ensemble methods in this framework. From this study, we have emphasized the goodness of the new proposed pruning heuristics, as they have allowed us to improve the performance of all baseline ensemble classifiers.

As future research we plan to apply a genetic search optimization procedure allowing for a more exhaustive search of the best cooperative classifiers within the ensemble. Also, to avoid the threshold parameter for the pruning approach, we should analyze the computation of the former regarding different factors. Another topic for future study is to study the behavior of the novel pruning metrics under different case studies, i.e., regarding data intrinsic characteristics. We also seek to

confirm that this good behavior achieved by the application of the ordering pruning approach is maintained independently of the baseline classifier selected. Finally, we have stressed the significance of this work by providing a wide number of prospects for different related topics, including novel approaches for pruning and enhancing diversity, testing recent preprocessing approaches in synergy with this methodology, the application of our proposal for data streams and online learning in imbalanced domains, and to address Big Data problems.

Acknowledgement

This work was supported by the Spanish Ministry of Science and Technology under projects TIN-2011-28488, TIN2013-40765-P, TIN2014-57251-P; Andalusian Research Plans P11-TIC-7765 and P10-TIC-6858; and both the University of Jaén and Caja Rural Provincial de Jaén under project UJA2014/06/15.

References

[1] J. Alcala-Fdez, L. Sanchez, S. Garcia, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernandez, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Comput. 13 (2009) 307–318.
[2] R. Barandela, J.S. Sanchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognit. 36 (3) (2003) 849–851.
[3] R. Barandela, R. Valdovinos, J. Sanchez, New applications of ensembles of classifiers, Pattern Anal. Appl. 6 (3) (2003) 245–256.
[4] 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. 6 (1) (2004) 20–29.
[5] M. Bhardwaj, V. Bhatnagar, Towards an optimally pruned classifier ensemble, Int. J. Mach. Learn. Cybern. 6 (5) (2015) 699–718.
[6] J. Blaszczynski, J. Stefanowski, Neighbourhood sampling in bagging for imbalanced data, Neurocomputing 150 (2015) 529–542.
[7] V. Bolon-Canedo, N.S.-M. no, A. Alonso-Betanzos, J.M. Benitez, F. Herrera, A review of microarray datasets and applied feature selection methods, Inf. Sci. 282 (2014) 111–135.
[8] L. Breiman, Bagging predictors, Mach. Learn. 24 (2) (1996) 123–140.
[9] A. Bria, N. Karssemeijer, F. Tortorella, Learning from unbalanced data: a cascade-based approach for detecting clustered microcalcifications, Med. Image Anal. 18 (2) (2014) 241–252.
[10] I. Brown, C. Mues, An experimental comparison of classification algorithms for imbalanced credit scoring data sets, Expert Syst. Appl. 39 (3) (2012) 3446–3453.
[11] F. Charte, A.J. Rivera, M.J. del Jesus, F. Herrera, Mlsmote: approaching imbalanced multilabel learning through synthetic instance generation, Knowl. Based Syst. 89 (2015) 385–397.
[12] 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.
[13] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explor. 6 (1) (2004) 1–6.
[14] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03), 2003, pp. 107–119.
[15] C.P. Chen, C.-Y. Zhang, Data-intensive applications, challenges, techniques and technologies: a survey on big data, Inf. Sci. 275 (2014) 314–347.
[16] S. del Rio, V. Lopez, J.M. Benitez, F. Herrera, On the use of mapreduce for imbalanced big data using random forest, Inf. Sci. 285 (2014) 112–137.
[17] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[18] 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.
[19] P. Domingos, Metacost: A general method for making classifiers cost–sensitive, in: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99), 1999, pp. 155–164.
[20] A. Fernandez, S. Rio, V. Lopez, A. Bawakid, M. del Jesus, J. Benitez, F. Herrera, Big data with cloud computing: an insight on the computing environment, mapreduce and programming framework, WIREs Data Min. Knowl. Discov. 4 (5) (2014) 380–409.
[21] Y. Freund, R. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, J. Comput. Syst. Sci. 55 (1) (1997) 119–139.
[22] M. Galar, A. Fernandez, E. Barrenechea, F. Herrera, Eusboost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognit. 46 (12) (2013) 3460–3471.
[23] M. Galar, A. Fernandez, 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. Part C: Appl. Rev. 42 (4) (2012) 463–484.
[24] S. García, A. Fernandez, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.
[25] S. Garcia, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2607–2624.
[26] S. Garcia, F. Herrera, Evolutionary under-sampling for classification with imbalanced data sets: proposals and taxonomy, Evol. Comput. 17 (3) (2009) 275–306.
[27] A. Ghazikhani, R. Monsefi, H.S. Yazdi, Online neural network model for non-stationary and imbalanced data stream classification, Int. J. Mach. Learn. Cybern. 5 (1) (2014) 51–62.
[28] L. Guo, S. Boukir, Margin-based ordered aggregation for ensemble pruning, Pattern Recognit. Lett. 34 (6) (2013) 603–609.
[29] M.M. Haque, M.K. Skinner, L.B. Holder, Imbalanced class learning in epigenetics, J. Comput. Biol. 21 (7) (2014) 492–507.
[30] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[31] D. Hernandez-Lobato, G.M.-M. noz, A. Suarez, Statistical instance-based pruning in ensembles of independent classifiers, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 364–369.
[32] S. Hido, H. Kashima, Y. Takahashi, Roughly balanced bagging for imbalanced data, Stat. Anal. Data Min. 2 (2009) 412–426.
[33] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.
[34] X. Hu, Using rough sets theory and database operations to construct a good ensemble of classifiers for data mining applications, in: N. Cercone, T.Y. Lin, X. Wu (Eds.), IEEE International Conference in Data Mining (ICDM), IEEE Computer Society, 2001, pp. 233–240.
[35] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Trans. Knowl. Data Eng. 17 (3) (2005) 299–310.
[36] S. Karakatic, V. Podgorelec, Improved classification with allocation method and multiple classifiers, Inf. Fusion 31 (2016) 26–42.
[37] M.J. Kim, D.K. Kang, H.B. Kim, Geometric mean based boosting algorithm with over-sampling to resolve data imbalance problem for bankruptcy prediction, Expert Syst. Appl. 42 (3) (2015) 1074–1082.
[38] B. Krawczyk, G. Schaefer, A hybrid classifier committee for analysing asymmetry features in breast thermograms, Appl. Soft Comput. J. 20 (2014) 112–118.
[39] B. Krawczyk, M. Wozniak, G. Schaefer, Cost-sensitive decision tree ensembles for effective imbalanced classification, Appl. Soft Comput. 14 (2014) 554–562.
[40] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Anal. Appl. 6 (1) (2003) 22–31.
[41] L.I. Kuncheva, Diversity in multiple classifier systems, Inf. Fusion 6 (1) (2005) 3–4.

[42] L.I. Kuncheva, J.J. Rodriguez, A weighted voting framework for classifiers ensembles, Knowl. Inf. Syst. 38 (2) (2014) 259–275.
[43] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles, Mach. Learn. 51 (2003) 181–207.
[44] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE 39 (2) (2009) 539–550.
[45] V. Lopez, A. Fernandez, M.D. Jesus, F. Herrera, A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets, Knowl. Based Syst. 38 (2013) 85–104.
[46] V. Lopez, A. Fernandez, S. Garcia, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (20) (2013) 113–141.
[47] V. Lopez, A. Fernandez, F. Herrera, On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed, Inf. Sci. 257 (2014) 1–13.
[48] V. Lopez, A. Fernandez, J.G. Moreno-Torres, F. Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics, Expert Syst. Appl. 39 (7) (2012) 6585–6608.
[49] Z. Lu, X. Wu, X. Zhu, J. Bongard, Ensemble pruning via individual contribution ordering, in: B. Rao, B. Krishnapuram, A. Tomkins, Y. Qiang (Eds.), KDD, ACM, 2010, pp. 871–880.
[50] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: D.H. Fisher (Ed.), 14th International Conference on Machine Learning (ICML), Morgan Kaufmann, 1997, pp. 211–218.
[51] G.M.-M. noz, D. Hernandez-Lobato, A. Suarez, An analysis of ensemble pruning techniques based on ordered aggregation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2) (2009) 245–259.
[52] G.M.-M. noz, A. Suarez, Using boosting to prune bagging ensembles, Pattern Recognit. Lett. 28 (1) (2007) 156–165.
[53] L.L. Minku, A.P. White, X. Yao, The impact of diversity on online ensemble learning in the presence of concept drift, IEEE Trans. Knowl. Data Eng. 22 (5) (2010) 730–742.
[54] L.L. Minku, X. Yao, Ddd: a new ensemble approach for dealing with concept drift, IEEE Trans. Knowl. Data Eng. 24 (4) (2012) 619–633.
[55] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodriguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognit. 45 (1) (2012) 521–530.
[56] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Trans. Neural Netw. Learn. Syst. 23 (8) (2012) 1304–1313.
[57] G.M.-M. noz, A. Suarez, Aggregation ordering in bagging, in: International Confernece on Artificial Intelligence and Applications (IASTED), 2004, pp. 258–263.
[58] Y. Park, J. Ghosh, Ensembles of α-trees for imbalanced classification problems, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 131–143.
[59] R. Polikar, Ensemble based systems in decision making, IEEE Circuits Syst. Mag. 6 (3) (2006) 21–45.
[60] R.C. Prati, G.E.A.P.A. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, Knowl. Inf. Syst. 45 (1) (2015) 247–270.
[61] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kauffman, 1993.
[62] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (1) (2010) 1–39.
[63] C. Seiffert, T. Khoshgoftaar, J.V. Hulse, A. Napolitano, RUSBoost: a hybrid approach to alleviating class imbalance, IEEE 40 (1) (2010) 185–197.
[64] J. Stefanowski, Dealing with data difficulty factors while learning from imbalanced data, in: S. Matwin, J. Mielniczuk (Eds.), Challenges in Computational Statistics and Data Mining, Studies in Computational Intelligence, vol. 605, Springer, 2016, pp. 333–363.
[65] Y. Sun, M.S. Kamel, A.K.C. Wong, Y. Wang, Cost–sensitive boosting for classification of imbalanced data, Pattern Recognit. 40 (12) (2007) 3358–3378.
[66] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, Int. J. Pattern Recognit. Artif. Intell. 23 (4) (2009) 687–719.
[67] C. Tamon, J. Xiang, On the boosting pruning problem, in: R.L. de Mantaras, E. Plaza (Eds.), 11th European Conference on Machine Learning (ECML), Lecture Notes in Computer Science, vol. 1810, Springer, 2000, pp. 404–412.
[68] K. Tumer, J. Ghosh, Error correlation and error reduction in ensemble classifiers, Connect. Sci. 8 (3-4) (1996) 385–403.
[69] I. Visentini, L. Snidaro, G.L. Foresti, Diversity-aware classifier ensemble selection via f-score, Inf. Fusion 28 (2016) 24–43.
[70] S. Vluymans, D.S. Tarrago, Y. Saeys, C. Cornelis, F. Herrera, Fuzzy rough classifiers for class imbalanced multi-instance data, Pattern Recognit. 53 (2016) 36–45.
[71] S. Wang, L.L. Minku, X. Yao, Resampling-based ensemble methods for online class imbalance learning, IEEE Trans. Knowl. Data Eng. 27 (5) (2015) 1356–1368.
[72] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09), 2009, pp. 324–331.
[73] S. Wang, X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, IEEE Trans. Knowl. Data Eng. 25 (1) (2013) 206–219.
[74] F. Wilcoxon, Individual comparisons by ranking methods, Biom. Bull. 1 (6) (1945) 80–83.
[75] M. Wozniak, M.G. na, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.
[76] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inf. Syst. 14 (1) (2007) 1–37.
[77] G. Yule, On the association of attributes in statistics, Philos. Trans. A 194 (1900) 257–319.
[78] 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, pp. 204–213.
[79] J.H. Zar, Biostatistical Analysis, Prentice Hall, Upper Saddle River, New Jersey, 1999.
[80] J. Zhai, S. Zhang, C. Wang, The classification of imbalanced large data sets based on mapreduce and ensemble of elm classifiers, in: International Journal of Machine Learning and Cybernetics, 2016, pp. 1–9, doi:10.1007/s13042-015-0478-7. in press.
[81] H. Zhang, M. Li, Rwo-sampling: A random walk over-sampling approach to imbalanced data classification, Inf. Fusion 20 (2014) 99–116.
[82] Y. Zhang, S. Burer, W.N. Street, Ensemble pruning via semi-definite programming, J. Mach. Learn. Res. 7 (2006) 1315–1338.
[83] Z.H. Zhou, J. Wu, W. Tang, Ensembling neural networks: Many could be better than all, Artif. Intell. 137 (2002) 239–263.(1–2)