Pattern Recognition 46 (2013) 3460–3471

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

EUSBoost: Enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling

Mikel Galar [a,*], Alberto Fernández [b], Edurne Barrenechea [a], Francisco Herrera [c]

[a] Departamento de Automática y Computación, Universidad Pública de Navarra, Campus Arrosadía s/n, 31006 Pamplona, Spain
[b] Department of Computer Science, University of Jaén, 23071 Jaén, Spain
[c] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain

ARTICLE INFO

Article history:
Received 14 December 2012
Received in revised form
21 March 2013
Accepted 9 May 2013
Available online 15 May 2013

Keywords:
Classification
Imbalanced data-sets
Ensembles
Class distribution
Kappa-error diagrams
Boosting

ABSTRACT

Classification with imbalanced data-sets has become one of the most challenging problems in Data Mining. Being one class much more represented than the other produces undesirable effects in both the learning and classification processes, mainly regarding the minority class. Such a problem needs accurate tools to be undertaken; lately, ensembles of classifiers have emerged as a possible solution. Among ensemble proposals, the combination of Bagging and Boosting with preprocessing techniques has proved its ability to enhance the classification of the minority class.

In this paper, we develop a new ensemble construction algorithm (EUSBoost) based on RUSBoost, one of the simplest and most accurate ensemble, which combines random undersampling with Boosting algorithm. Our methodology aims to improve the existing proposals enhancing the performance of the base classifiers by the usage of the evolutionary undersampling approach. Besides, we promote diversity favoring the usage of different subsets of majority class instances to train each base classifier. Centered on two-class highly imbalanced problems, we will prove, supported by the proper statistical analysis, that EUSBoost is able to outperform the state-of-the-art methods based on ensembles. We will also analyze its advantages using kappa-error diagrams, which we adapt to the imbalanced scenario.

1. Introduction

Skewed class distributions hinder the classifier learning task, since the generalization capabilities of the classifiers are compromised [24,53,29]. Class imbalance [50,25] has been appointed as one of the most challenging problems in Data Mining [57]; it refers to data-sets having a very different number of instances from each one of the two classes (focusing on binary problems), that is, their presence is not balanced in the data-set as expected by the classifier. Even though the uneven class distribution need not be a problem itself, this situation usually produces overlapping or small disjuncts, added to the frequently small sample size of the minority class [50,25]. Besides these problems, another important point is that the evaluation criterion used to guide the learning process and evaluate the classifiers can lead to misunderstandings. The minority (positive) class can be ignored and treated as noise, losing the generalization ability of the classifiers in favor of the majority (negative) class. The great attention that this problem has attracted in the literature is understandable due to the large amount of real-world classification problems suffering from these conditions, such as anomaly detection [30], image annotation [59] or facial age estimation [9], among many others.

Having such a large number of potential applications, many techniques have been developed aiming to address the problem. These techniques can be categorized into three groups plus other one which makes use of the methods from the former groups to construct ensembles of classifiers. The first group addresses the modification of existing algorithms [55]. A different approach considering preprocessing techniques [5,11] aims to balance the skewed class distribution of the examples prior to the training task. In third place, cost-sensitive approaches [10,17] combine the mentioned techniques in such a way that they incorporate different misclassification costs for the instances in the learning algorithm. Finally, ensemble solutions [12,49,47] try to combine one of the previous approaches with an ensemble learning algorithm to form ensembles of accurate[1] and diverse base classifiers. These solutions have been proved to be more accurate than previous non-ensemble approaches [19].

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

[1] The accuracy of the base classifiers in the imbalance framework is measured by the proper performance measures, we refer to accurate and accuracy of the base classifiers in the sense of accurate classifiers that must correctly distinguish both classes (see Section 2.2).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.05.006

Ensembles of classifiers [46] attempt to increase the accuracy of individual classifiers by their combination. In the specialized literature, there are several practical and theoretical reasons for the usage of ensemble approaches instead of non-ensemble ones (in depth discussions can be found in [33,41,46]). Among the different types of ensembles, class imbalance has been mainly overcome by the combination of minor variants of the same classifier, which are built changing the class distribution of the instances. The way in which these distributions are altered is the key factor to construct diverse ensembles obtaining higher accuracies than those obtained by their individual base classifiers [33]. The most common approaches to construct ensembles by data-variation are AdaBoost [18] and Bagging [7]. Nevertheless, on their own, ensembles are not able to deal with imbalanced problems, since they are inherently designed to maximize accuracy measure. For this reason, they are combined with techniques to address the problem of uneven class distributions; among these proposals, those combining preprocessing algorithms and ensembles [12,4,47] are the most common. In particular, the simplest combinations have stood out with respect to more complex proposals. Any combination between random undersampling or SMOTE [11] with AdaBoost (RUSBoost [47], SMOTEBoost [12]) or Bagging (UnderBagging [4], SMOTEBagging [51]) excels in comparison with more elaborated methods [19]. In the empirical study carried out in [19], RUSBoost was the ensemble with the best trade-off, being one of the best performers, but also the least complex; it is simple and fast, besides it was found to be equivalent to SMOTEBagging in terms of performance.

Analyzing these results, it is interesting to observe that ensembles based on random sampling techniques are highly competitive despite its randomness (due to their good accuracy-diversity relation); however, in the imbalance framework, we believe that such a randomness might be improved enhancing the accuracy of the base classifiers in a supervised manner.

To do so, instead of random undersampling, our proposal considers the usage of Evolutionary Undersampling (EUS) [23], in which we also stress the diversity between classifiers favoring the most diverse chromosomes in comparison to the previously used ones. This diversity mechanism is a key component in our model, because we mainly aim at increasing the accuracy of the base classifiers but not being at the expense of losing much diversity with respect to that of RUSBoost. We will show that the proposed EUSBoost ensemble outperforms RUSBoost and the other best state-of-the-art ensembles in highly imbalanced data-sets, where this strategy can be exploited. In addition, we explain the behavior of our proposal adapting the kappa-error diagrams [38] to the imbalance framework.

In order to carry out the empirical comparison, we establish a similar experimental framework to that in [19]; we test 33 two-class highly imbalanced real-word data-sets from KEEL data-set repository [1,2]. We consider C4.5 [43] as base classifier for our experiments since it has been the most commonly used in imbalanced domains [5,14,19] and the Area Under the ROC Curve (AUC) [6,28] as evaluation criterion. The results obtained in the comparisons are contrasted by the proper non-parametric statistical tests, as suggested in the literature [13,21,22].

The main contributions of this paper can be summarized as follows:

- We present a novel ensemble method based on Boosting in combination with EUS (instead of random undersampling [47] or SMOTE [12]).
- This combination allows us to introduce a modification in the fitness function of EUS to promote diversity of the undersampled data-sets, which is translated into more accurate ensembles when dealing with highly imbalanced data-sets [52].
- We introduce kappa-AUC error diagrams to explain the behavior of EUSBoost in the imbalance framework.

The rest of this paper is organized as follows: In Section 2, we recall the problem of imbalanced data-sets, discussing the evaluation metrics and describing the best state-of-the-art ensemble techniques. In Section 3, we put forward our proposal. Next, in Section 4, we introduce the experimental framework used in the experiments. In Section 5, we carry out the comparison of our proposal with the state-of-the-art methods, whereas in Section 6, we explain the reason of the better behavior of EUSBoost using kappa-AUC error diagrams. Finally, in Section 7, we make our concluding remarks.

2. The problem of skewed class distributions in classification

In this section, we first recall the problems that may arise due to the imbalanced class distribution of the instances. Then, we show the evaluation criteria that are commonly used in imbalanced scenarios. Finally, we review the best ensemble approaches found in [19] that will be the base for the comparison.

2.1. The class imbalance problem

A data-set is said to be imbalanced whenever the number of instances from the different classes is not nearly the same. Focusing on a two-class imbalanced scenario, the problem is that one class is under-represented in the data-set. Moreover, from the point of view of the learning task, it is usually the class of interest [25].

In an imbalanced scenario, standard classifier learning algorithms usually fail due to their accuracy-oriented design. They are biased toward the majority class, because it is easier to learn. The correct classification of negative instances favors accuracy metric more than the correct prediction of the minority class instances. Hence, positive instances might be ignored (treated as noise), because general rules predicting the negative class produce better accuracy rates. Although the mere fact that the data-set is imbalanced need not mean an added difficulty to the learning task [50,25] (since the classes could be perfectly separable), in real-world problems it usually brings along a series of difficulties that hinder the classifier learning: such as small sample size [29], overlapping [24] or small disjuncts [53].

In this paper, in order to organize the different data-sets in some way, we consider the Imbalance Ratio (IR) defined as nthe number of negative class examples divided by the number of positive class examples. Using the IR, we refer to a data-set as highly imbalanced if its IR is greater than 9 [23].

2.2. Performance evaluation in imbalanced domains

How to evaluate a classifier is of great importance to properly assess its classification performance and guide its modeling. Focusing on two-class problems, the results of the correctly and incorrectly recognized examples of each class can be recorded in a confusion matrix (Table 1).

Table 1
Confusion matrix for a two-class problem.

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

From this matrix, different measures can be deduced to perform the evaluation in the imbalance framework:

- *True positive rate*, $TP_{rate} = TP/(TP + FN)$.
- *True negative rate*, $TN_{rate} = TN/(FP + TN)$.
- *False positive rate*, $FP_{rate} = FP/(FP + TN)$.
- *False negative rate*, $FN_{rate} = FN/(TP + FN)$.

The quality of the results obtained by a classification algorithm should be assessed by its performance on both classes at the same time, hence, these measures on their own are still inadequate. Receiver Operating Characteristic (ROC) graphic [6] combines these measures to produce a valid evaluation criterion. ROC graphics allow one to visualize the trade-off between TP_{rate} (benefits) and FP_{rate} (costs), evidencing that increasing the number of true positives without also increasing the number of false positives is not possible for any classifier. Area Under the ROC Curve (AUC) [28] corresponds to the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. AUC provides a scalar measure of the performance of a classifier and it has been widely used in imbalanced domains [24,23,48]. AUC measure is computed as the area of the ROC curve:

$$AUC = \frac{1 + TP_{rate} - FP_{rate}}{2} \quad (1)$$

2.3. Solutions for the class imbalance problem

As mentioned in the Introduction, the approaches to deal with class imbalance were initially divided into three groups depending on how they overcome the problem: algorithm level modifications [3], data-level treatments [5,16] and cost-sensitive approaches [10,17]. However, in recent years, techniques based on ensembles of classifiers have come up showing their usefulness [19].

Ensemble techniques for the class imbalance problem can be categorized into two main classes: cost-sensitive approaches and ensemble learning algorithms with embedded data preprocessing techniques. Cost-sensitive approaches [49] have the same disadvantage as non-ensemble cost-sensitive methods: the costs must be defined, and these costs are not usually found in standard classification data-sets; besides, it can be difficult to set them appropriately, even subjective. Otherwise, methods combining ensemble learning algorithms with data-level techniques are more general, and have attracted more attention. They can also be divided into three subclasses: Bagging-, Boosting-, and Hybrid-based ensembles, depending on the ensemble learning algorithm in which they are based. An extensive empirical analysis of ensemble solutions for class imbalance was carried out in [19], where both Boosting [18] and Bagging [7] in combination with preprocessing techniques achieved the best results. Hereafter, we recall these ensemble learning algorithms:

- *Bagging* [7]. It consists of training different classifiers with bootstrapped replicas of the original training data-set. Hence, diversity is obtained by resampling different data subsets.
- *Boosting* [18]. It uses the whole data-set to train each classifier serially, but after each round, it gives more focus to difficult instances, with the goal of correctly classifying in the following iteration those examples that were incorrectly classified during the current one. In this work, we use AdaBoost.M2 [18], which has been widely employed in imbalanced domains in combination with data level techniques. The main advantage of AdaBoost.M2 with respect to other Boosting approaches is that it considers the confidences given by the base classifiers in the weight update, taking advantage of these estimations.

Different preprocessing techniques have been embedded in these ensemble methods. Among these combinations, those considering the random undersampling method and the Synthetic Minority Oversampling Technique (SMOTE) [11] have been the most successful ones [19]. Afterwards, we briefly recall the operating procedure of these data-level methods:

- *Random Undersampling*. It balances the class distribution by randomly eliminating majority class examples. Its major drawback is that potentially useful data can be discarded.
- *Synthetic Minority Oversampling Technique* (SMOTE) [11]. It is an oversampling method that creates minority class examples by interpolating several minority class instances that lie together (the k Nearest Neighbors, k NN, are considered).

After recalling both base techniques, we briefly survey the best state-of-the-art ensembles for class imbalance problem, the first two algorithms are Bagging-based, the next two are Boosting-based ensembles, whereas the last one is a hybrid approach:

- *UnderBagging* [4] randomly undersamples the data-set in each Bagging iteration (all minority class instances are kept in every iteration).
- *SMOTEBagging* [51] uses SMOTE in each iteration. The new data-set contains two times the number of majority class instances. The first half is a bootstrapped replica of the majority class instances, whereas the second half is obtained via SMOTE and random oversampling depending on a resampling rate.
- *RUSBoost* [47] removes instances from the majority class in each iteration of AdaBoost.M2 using the random undersampling technique. The weights of the instances in the new undersampled data-set are normalized to form a distribution.
- *SMOTEBoost* [12] introduces synthetic minority class instances using SMOTE algorithm. Since new instances are created, new weights must be assigned, which are proportional to the total number of instances in the new data-set. The weights of the instances from the original data-set are normalized to form a distribution with the new instances.
- *EasyEnsemble* [37] performs similarly to UnderBagging, but in spite of training a classifier for each new bag, they train each bag using AdaBoost. Hence, the final classifier looks like an ensemble of ensembles, despite it is a single ensemble.

3. EUSBoost: evolutionary undersampling guided boosting

In this section, we explain our proposal to overcome the class imbalance problem with ensembles of classifiers. As we have explained, our aim is to improve the performance of previous approaches, and more specifically that of RUSBoost, enhancing the accuracy of the base classifiers while maintaining their diversity. Whereas the original EUS aims to increase the accuracy, its modification to account for diversity is a very important component of our model, since it allows us to improve the global performance.

First in Section 3.1, we recall EUS algorithm. Afterwards in Section 3.2, we show our initial model which uses EUS inside Boosting. Then, in Section 3.3, we present how we enhance the diversity of the base classifiers with a modified fitness function in EUS model. Finally, we discuss the computational complexity of EUSBoost in Section 3.4.

3.1. Evolutionary undersampling

EUS [23] arises from the application of evolutionary prototype selection algorithms to imbalanced domains, where some of their features can be better fitted, e.g., the fitness function. Prototype selection [20] is a sampling process which, instead of aiming to

balance the class distributions (as the ones shown in Section 2.3), reduces the reference set for the nearest neighbor (1NN) classifier in order to improve its accuracy and reduce the storage necessity. Within the imbalance framework these objectives change, since the balance of the data distribution (aiming to prevent bad behaviors in the classification process) gains importance. Therefore, EUS attempts to obtain a useful subset of the original data-set. In order to do so, it starts randomly undersampling several data subsets, which are evolved until the currently best undersampled data-set cannot be further improved (in terms of the fitness function). EUS has shown its usefulness in real-world applications [14].

In every evolutionary method, the representation of the solution is a key issue, that is, how is a chromosome that codifies the real solution modeled. In EUS, each chromosome is a binary vector representing the presence or absence of instances in the data-set. Even though the whole data-set can be codified within the chromosome, we reduce the search space by only considering the majority class instances; hence, all the minority class instances are always introduced in the new data-set. Accordingly, a chromosome in EUS is represented as

$$V = (v_{x_1}, v_{x_2}, v_{x_3}, v_{x_4}, \ldots, v_{x_{n^-}}), \tag{2}$$

where v_{x_i} takes the values 0 or 1, indicating whether instance x_i is included or not in the data-set (n^- stands for the number of majority class instances).

In order to rank the chromosomes, we use a fitness function that considers the balancing between the minority class and majority class instances aside from taking into account the expected performance of the selected data subset [23]. The performance is estimated by hold-one-out technique using 1NN classifier, and is measured by the geometric mean (GM) [3] (Eq. (3)), which allows one to maximize the accuracy of both classes at the same time,

$$\mathrm{GM} = \sqrt{\mathrm{TP}_{rate} \cdot \mathrm{TN}_{rate}}. \tag{3}$$

The fitness function of EUS is as follows:

$$\mathrm{fitness}_{\mathrm{EUS}} = \begin{cases} \mathrm{GM} - |1 - \frac{n^+}{N^-} \cdot P| & \text{if } N^- > 0 \\ \mathrm{GM} - P & \text{if} N^- = 0, \end{cases} \tag{4}$$

where n^+ is the number of minority (positive) class instances, N^- is the number of majority (negative) class instances selected and P is the penalization factor accounting for the importance given to the balance between both classes. The recommended value for P is 0.2, providing a good trade-off between accuracy and balancing. Notice that the AUC (Eq. (1)) could also be used as a performance measure, but in our case, following the empirical analysis carried out in [23], we use the GM since it has less over-fitting problems and performs better in EUS (even though AUC is then used as evaluation criterion).

EUS makes use of the well-known CHC algorithm [15], holding a good balance between exploration and exploitation, to evolve the initially random population. Recall that CHC is an elitist genetic algorithm that uses the heterogeneous uniform cross-over (HUX) to combine two chromosomes, which interchanges exactly half of their different genes. The recombination strategy is only carried out if the Hamming distance between two chromosomes is greater than the threshold (initially $L/4$, being L the length of the chromosome); if no parents are recombined the threshold is reduced by one (incest prevention mechanism). Besides, no mutation is applied, instead, when there is no more progress (recombined chromosomes do not improve their parents and the threshold is zero), the chromosomes are reinitialized. To do so, the best chromosome is used as a template and the chromosomes are generated by randomly changing the 35% of the genes. Then, the evolution is resumed.

In the case of EUS, the HUX of the original CHC algorithm is modified, decreasing the probability of including instances in the data-set, so that a good reduction rate is obtained. In this manner, each time HUX switches a gene on, the gene is switched off with a certain probability (which recommended value is 0.25).

3.2. Combining boosting and evolutionary undersampling

Random techniques are powerful to construct ensembles, since they provide much diversity, which combined with accurate base classifiers is translated into high performance ensembles [33]. However, from our point of view, such an uncontrolled randomness could be better managed when dealing with highly imbalanced data-sets, leading to an improved performance. Even though random undersampling is often an appropriate technique assessing good results, it could discard potentially useful instances of the majority class, which might be important for the learning process. Besides, as the IR of the data-set increases, so does the probability of ignoring useful majority class examples. For this reason, we focus on highly imbalanced data-sets aiming to undertake this problem by the usage of EUS, which has proved its validity in this framework [23].

Due to the initial randomness of the solutions of EUS, the resulting data subset usually differs from one execution to another (evolutionary algorithms are stochastic search procedures). We benefit from this instability, since it helps maintaining the diversity (classifiers trained with identical data-sets are not useful to construct ensembles); in addition, we seek for diversity modifying the evolutionary process (as we will explain in Section 3.3).

The inclusion of EUS within Boosting algorithm is simple and easy to implement, yet effective. We follow the idea of RUSBoost and other Boosting-based algorithms [19] introducing the undersampling process inside the loop of AdaBoost.M2. In this case, EUS is used instead of random undersampling or SMOTE, and then, only the weights of the instances in the undersampled data-set are used in the induction process (as in RUSBoost). The whole procedure of EUSBoost is outlined in Algorithm 1. The new steps introduced in our proposal are the 7th and 8th, while the 9th is modified:

- *Step* 7—EUS is introduced, which returns a new data-set (S') considering all the minority class instances and the majority class instances selected.
- *Step* 8—The new weight distribution is computed.
- *Step* 9—The classifier is trained, in this step the original data-set is maintained, but the instances that are not in the undersampled data-set have no weight, and hence, they are ignored.

Algorithm 1. EUSBoost, EUS embedded in AdaBoost.M2.

Input: Training set $S = \{\mathbf{x}_i, y_i\},\ i = 1, \ldots, N$; and $y_i \in \{c_1, c_2\}$; T: Number of iterations; I: Weak learner

Output: Boosted classifier: $H(x) = \arg\max_{y \in \mathbb{C}} \sum_{t=1}^{T} \ln\left(\frac{1}{\beta_t}\right) h_t(x, y)$

where h_t, β_t (with $h_t(x, y) \in [0, 1]$) are the classifiers and their assigned weights, respectively

1: $D_1(i) \leftarrow 1/N$ for $i = 1, \ldots, N$

2: $w^1_{i,y} \leftarrow D_1(i)$ for $i = 1, \ldots, N,\ y \neq y_i$

3: **for** $t = 1$ to T **do**

4: $W^t_i \leftarrow \sum_{y \neq y_i} w^t_{i,y}$

5: $q_t(i, y) \leftarrow \frac{w^t_{i,y}}{W^t_i}$ for $y \neq y_i$

6: $D_t(i) \leftarrow \frac{W^t_i}{\sum_{i=i}^{N} W^t_i}$

7: $S' = \text{EvolutionaryUndersampling}(S)$;

8: $$D'_t(k) \leftarrow \begin{cases} \dfrac{W_i^t}{\sum_{x_i \in S'} W_i^t} & \text{if } x_i \in S' \\ 0 & \text{otherwise} \end{cases}$$

9: $h_t \leftarrow I(S, D'_t)$

10: $$\epsilon_t \leftarrow \frac{1}{2} \sum_{i=1}^{N} D_t(i) \left(1 - h_t(\mathbf{x}_i, y_i) + \sum_{i, y \neq y_i} q_t(i,y) h_t(\mathbf{x}_i, y) \right)$$

11: $$\beta_t = \frac{\epsilon_t}{1-\epsilon_t}$$

12: $$w_{i,y}^{t+1} = w_{i,y}^t \cdot \beta_t^{(1/2)(1+h_t(\mathbf{x}_i,y_i)-h_t(\mathbf{x}_i,y))} \text{ for } i = 1, \ldots, N, \ y \neq y_i$$

13: **end for**

3.3. Promoting diversity: adjusting the fitness function of evolutionary undersampling

Diversity in classifier ensembles, that is, being composed of different classifiers giving different outputs, is crucial and has been widely studied in the literature [8,52]. However, no direct relation has been found between diversity and accuracy despite several different measures have been tested [35]. However, dealing with imbalanced data-sets, the importance of diversity is even clearer, being usually easier to produce diverse classifiers by data-variation due to the problems mentioned in Section 2.1. This fact is in accordance with the findings in [52], where the authors show that there exists a relation between diversity and single-class performance measures in this scenario, having a positive impact on the minority class classification, but also on global performance measures such as AUC.

One of the problems of EUSBoost is that seeking for accurate base classifiers causes a loss of diversity in the resulting ensemble. Moreover, more accurate base classifiers need not produce a more accurate ensemble. In this work, we aim to take advantage of the class imbalance problem to boost the diversity of the base classifiers in EUSBoost. On this account, we modify the objective function used in the undersampling process in order to favor the chromosomes in which we are more interested. In our case, these chromosomes are those better combining diversity and performance, in terms of the original fitness function. Nevertheless, notice that we measure the diversity of the chromosomes instead of the diversity of the outputs of the classifiers as it is usually done [35], because we need to measure it *a priori* before learning the classifier. Therefore, our assumption is that base classifiers learned from data-sets with much more different instance sets are more diverse (and we force it in a supervised manner, which is not usually done). Moreover, in order to further increase diversity, we change the weight of each factor (accounting for diversity/accuracy) in each iteration. In order to introduce both ideas within EUSBoost, we develop a new fitness function modifying the original evaluation procedure of EUS (Eq. (4)). Afterwards, we explain this procedure in detail.

There exist several measures to compute the diversity [35]. One of the most common is the Q-statistic [58], which has been widely applied in classifier ensembles [32,52]. In our case, we apply this measure to compute the diversity between two solutions (Eq. (2)), since each instance has an associated value indicating whether it is included or not in the new data-set (1 and 0, respectively). Therefore, having two binary vectors (V_i, V_j), the Q-statistic is computed as follows:

$$Q_{i,j} = \frac{N^{11}N^{00} - N^{01}N^{10}}{N^{11}N^{00} + N^{01}N^{10}}, \tag{5}$$

where N^{ab} stands for number of elements (instances) with value a in the first vector and with value b in the second (if $a=b$ both data-sets agree including or not the instance). The value of the statistic ranges from −1 to 1. Lower values of Q indicate greater diversity ($Q_{i,j}=0$ means that both vectors are statistically independent).

The problem of the Q-statistic and other diversity measures is that they are pairwise measures [35]. In order to evaluate each possible solution, we need to compare it with all the previous vectors used to construct the base classifiers, aggregating all the pairwise diversity values. In this case, we propose to use the maximum of all the pairwise diversities $Q_{i,j}$, different from the usual way where the arithmetic mean is used [35]. In this manner, we promote the candidate instance subset that is the most dissimilar with respect to all the previous data-sets considered. Therefore, being V_j the candidate solution to be evaluated, and $V_i,\ i=1,\ldots,t$ (recall that t is the current iteration) all the previously used solutions, we compute the global diversity Q as

$$Q = \max_{i=1,\ldots,t} Q_{i,j}. \tag{6}$$

After defining the evaluation of the diversity, we carry out a modification of the fitness function of EUS:

$$\text{fitness}_{\text{EUS}_Q} = \text{fitness}_{\text{EUS}} \cdot \frac{1.0}{\beta} \cdot \frac{10.0}{\text{IR}} - Q \cdot \beta, \tag{7}$$

where $\text{fitness}_{\text{EUS}}$ is the original fitness function (Eq. (2)), IR is the imbalance ratio, Q is the global Q-statistic and β is a weight factor changing in each iteration:

$$\beta = \frac{N-t-1}{N}. \tag{8}$$

We should notice that in Eq. (7) the Q term is subtracted to maximize the diversity. Moreover, in the first iteration of the ensemble ($t=1$), EUS is executed in its original form (using Eq. (4)), since we have no vectors to compare with the actual candidate solution.

Remark 1. In the new fitness function, as long as the iterations progress, more importance is given to the precision (the original EUS fitness function) and less to the diversity. This procedure is in accordance with the philosophy of Boosting, since in later iterations the correct classification of the most difficult instances gains importance. Furthermore, the larger the IR is, the less importance the original fitness has and the greater the influence of the diversity is. When the IR is moderately high (around 10), it is important to have accurate base classifiers. However, diversity cannot be forced to the same extent, since having less instances to decide upon (that is, less majority instances to be removed, and more to be maintained in the data-set to balance the class distribution), forcing the diversity could imply a severe decrease of the performance. The contrary occurs when the IR increases (there are more majority class instances to be left out to balance the data-set). In case of data-sets with extreme imbalance, where it is easier to reach high diversity, notice that all the importance is not given to the diversity, since all the chromosomes will share the same weight for the accuracy part, and hence, despite its absolute value is low, this part will also be taken into account.

In order to understand how the fitness function works along the iterations of EUSBoost, in Fig. 1, we provide an example of its behavior in a data-set used in the experiments. It can be observed that initially a high performance can be obtained with the greatest diversity; as iterations progress, the performance needs to be decreased in order to maintain diversity. Finally, at last iterations, it is more difficult to maintain the diversity (there are more previous subsets), but also the weight of the performance is increased, and hence, a less diverse but more accurate subset is selected.

In addition to the Q-statistic, we have considered another way to measure the diversity of the solutions, which is not usually used to measure the diversity between the outputs of the classifiers. We can simply consider that each solution is a code-word (i.e.,

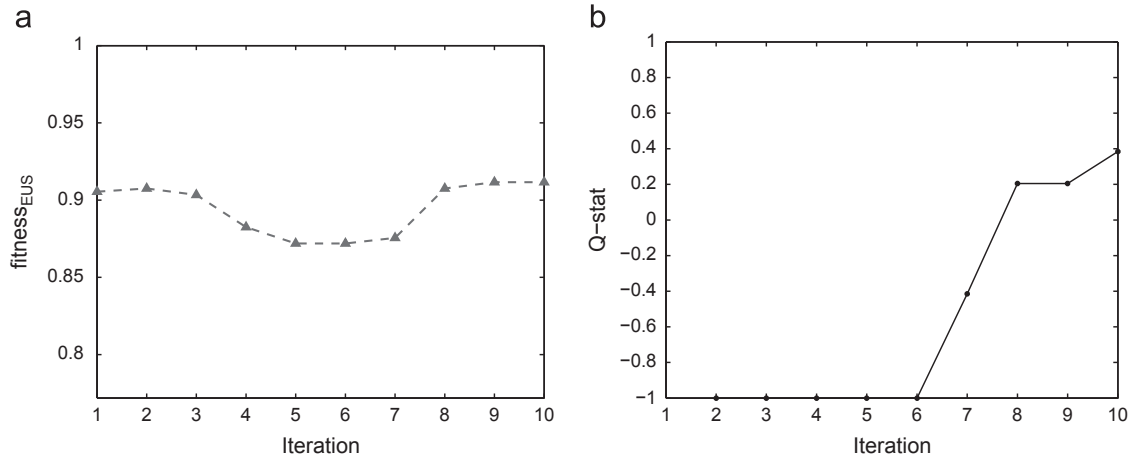

Fig. 1. An example of the behavior of both components of the fitness function in each iteration of EUSBoost (the values of the chromosome/subset selected are shown): (a) fitness function of EUS (performance); and (b) global Q-statistic (diversity).

a chromosome is a code-word), which codifies the instances to be used in the training phase. Hence, we can measure how different two code-words are by the well-known Hamming distance, that is, the number of bits that differ between two code-words. In this case, the global procedure is the same, but the fitness function changes slightly, since our aim is to maximize the Hamming distance between all solutions (the greater the distance is, the most dissimilar the code-words are):

$$\text{fitness}_{\text{EUS}_H} = \text{fitness}_{\text{EUS}} \cdot \frac{1.0}{\beta} \cdot \frac{10.0}{IR} + H \cdot \beta, \tag{9}$$

where H is the minimum Hamming distance between the candidate chromosome and all the previous chromosomes used to construct the existing classifiers. The Hamming distance in Eq. (9) is normalized by dividing its result by the number of bits in the code-word (genes in the chromosome, i.e., instances codified).

3.4. On the computational complexity of EUSBoost

We acknowledge that EUSBoost training phase is computationally more expensive than that of the other methods. This is due to EUS, which is executed in all iterations of Boosting. Focusing on the comparison with RUSBoost, which simply needs to randomly select the instances in each iteration, the training procedure requires more computations. Note that in EUS, 1NN classifier is executed using each chromosome as a reference set (being undersampled data-sets, the higher the IR is, the faster the method runs) for the whole data-set.

In our algorithm there is an added cost with respect to RUSBoost, since EUS is executed in each iteration of Boosting (instead of random undersampling). The complexity of EUS comes from the application of 1NN for the evaluation of each chromosome. 1NN complexity is linear with respect to the number of examples and attributes, i.e., its complexity is of $\mathcal{O}(n_{ref} \cdot m)$, where n_{ref} is the number of examples on the reference set (in our case the undersampled data-set) and m the number of attributes. In order to estimate the hold-one out performance, 1NN needs to be executed n times (being n the number of examples in the data-set). Since EUS is executed $T-1$ times (being T the number of Boosting iterations, 10 in our experiments), and max_{eval} is the number of allowed evaluations for EUS (10 000 in our experiments), the added complexity of our algorithm is of $\mathcal{O}(n \cdot n_{ref} \cdot m \cdot max_{eval} \cdot (T-1))$. We should notice that in this case, n_{ref} is always near $2 \cdot n^+$ (two times the number of minority class examples), since the data-sets evaluated are undersampled, and hence the complexity decreases with greater IR.

Anyway, as well as in any other preprocessing technique, the training time is just taken into account once per data-set, being the testing time of EUSBoost equivalent to that of RUSBoost, since the same number of classifiers and instances are considered. Hence, its application in classification problems that do not require on-line training might not be compromised.

4. Experimental framework

In this section, we present the set-up used to develop the empirical comparison in Section 5. The configuration is the same as the one used in [19] to carry out the empirical study of the state-of-the-art ensemble methods, but in this case focusing on highly imbalanced data-set for which EUSBoost has been designed. First, we show the ensemble methods that were found to be the best performers in [19], which are the base for the comparison in Section 4.1. Afterwards, we provide details of the real-world highly imbalanced problems that we have used to carry out the comparison in Section 4.2. Finally, we recall the statistical tests that we have applied to properly compare the classifiers performance in Section 4.3.

4.1. Algorithms and parameters

First of all, we must set the baseline classifier used by all the ensembles, which needs to be a weak learner [40]. We use C4.5 decision tree generating algorithm [43] considering that the previous approaches were proposed in combination with C4.5 [19] (carrying out a fair comparison) and its usage in the imbalance framework is common [5,52,19]. Moreover, it has been appointed as one of the top-10 Data Mining algorithms [56] and it is considered as a standard to build accurate ensembles [34,45,40].

Regarding ensemble learning algorithms, we compare our approach with those that were proved to be the best performers (explained in Section 2.3). Table 2 summarizes their operating procedure grouped by families and shows the abbreviations that we will use through the experimental study.

In our experiments, all the methods must have the same opportunities to achieve their best performance, but constraining the fine-tuning of their parameters depending on the data-set. In classifier ensembles, usually the higher the number of classifiers in the ensemble, the better the assessed performance is; however, this need not occur in all methods (i.e., more non-diverse classifiers could worsen the results or produce over-fitting problems). In [19], it was shown that Boosting-based ensembles work better

Table 2
State-of-the-art ensembles considered.

Abbr.	*Method*	*Short description*
RUS	RUSBoost [47]	AdaBoost.M2 with random undersampling
SBO	SMOTEBoost [12]	AdaBoost.M2 with SMOTE
UB	UnderBagging [4]	Bagging with undersampling of the majority class
SBAB	SMOTEBagging [51]	Bagging where each bag's SMOTE quantity varies
EASY	EasyEnsemble [37]	Bagging with undersampling of the majority class and AdaBoost

Table 3
Parameters for C4.5 and the preprocessing algorithms.

Algorithm	*Parameters*
C4.5	Prune=True, Confidence level=0.25 Minimum number of item-sets per leaf=2 Confidence=Laplace Smoothing [42]
SMOTE	Number of Neighbors k=5, Quantity=Balance, Distance=Heterogeneous Value Difference Metric (HVDM)
EUS	Population Size=50, Number of Evaluations=10 000, Probability of inclusion HUX=0.25, Evaluation Measure=GM, Selection Type=Majority, Distance Function=Euclidean, Balancing=True, P=0.2

with only 10 classifiers, whereas Bagging-based ones need 40 to reach their maximum potential. For this reason, aiming to find the best performer algorithm, we consider 10 classifiers for the former ensembles and 40 for the latter and hybrid ones. Table 3 shows the configuration parameters used to run C4.5 and the preprocessing algorithms (SMOTE and EUS). All experiments have been developed under KEEL software [1,2]. We must stress that the implementations of the algorithms in Table 2 are publicly available in KEEL source code, whereas the proposed method is available from the authors upon request.

4.2. Data-sets

We have considered the 33 most imbalanced binary data-sets from KEEL data-set repository [1], which are publicly available on the corresponding web-page.[2] In order to obtain these two-class imbalanced problems, original multi-class data-sets were modified in such a way that the union of one or more classes was labeled as the positive class and the same was done to obtain the negative class. Table 4 summarizes the properties of the data-sets: for each data-set, the number of examples (#Ex.), number of attributes (#Atts.), the percentage of examples of each class and the IR. This table is ordered according to the IR (last column).

We have obtained AUC metric estimates using a 5-fold stratified cross-validation. This process was carried out three times with different seeds. The data partitions used in this paper can be found in KEEL-dataset repository [1], so that any interested researcher can reproduce the experimental study.

4.3. Statistical tests

Statistical analysis needs to be carried out in order to compare the different algorithms appropriately. We use non-parametric tests as suggested in the literature [13,22,21]. The parametric statistical analysis loses its credibility because the initial conditions guaranteeing its reliability may not be satisfied [13]. Any interested reader can find additional information on the Website http://sci2s.ugr.es/sicidm/. In this paper, we consider two different tests to perform two types of comparisons:

Table 4
Summary description of the imbalanced data-sets.

No.	Data-sets	#Ex.	#Atts.	(%min;%maj)	IR
1	Glass04vs5	92	9	(9.78, 90.22)	9.22
2	Ecoli0346vs5	205	7	(9.76, 90.24)	9.25
3	Ecoli0347vs56	257	7	(9.73, 90.27)	9.28
4	Yeast05679vs4	528	8	(9.66, 90.34)	9.35
5	Ecoli067vs5	220	6	(9.09, 90.91)	10.00
6	Vowel0	988	13	(9.01, 90.99)	10.10
7	Glass016vs2	192	9	(8.89, 91.11)	10.29
8	Glass2	214	9	(8.78, 91.22)	10.39
9	Ecoli	336	7	(8.63, 91.37)	10.59
10	Led7digit	443	7	(8.35, 91.65)	10.97
11	Glass06vs5	108	9	(8.33, 91.67)	11.00
12	Ecoli01vs5	240	6	(8.33, 91.67)	11.00
13	Glass0146vs2	205	9	(8.29, 91.71)	11.06
14	Ecoli0147vs56	332	6	(7.53, 92.47)	12.28
15	Cleveland0vs4	177	13	(7.34, 92.66)	12.62
16	Ecoli0146vs5	280	6	(7.14, 92.86)	13.00
17	Ecoli4	336	7	(6.74, 93.26)	13.84
18	Yeast1vs7	459	8	(6.72, 93.28)	13.87
19	Shuttle0vs4	1829	9	(6.72, 93.28)	13.87
20	Glass4	214	9	(6.07, 93.93)	15.47
21	Page-blocks	472	10	(5.93, 94.07)	15.85
22	Abalone9vs18	731	8	(5.65, 94.25)	16.68
23	Glass016vs5	184	9	(4.89, 95.11)	19.44
24	Shuttle2vs4	129	9	(4.65, 95.35)	20.5
25	Yeast1458vs7	693	8	(4.33, 95.67)	22.10
26	Glass5	214	9	(4.20, 95.80)	22.81
27	Yeast2vs8	482	8	(4.15, 95.85)	23.10
28	Yeast4	1484	8	(3.43, 96.57)	28.41
29	Yeast1289vs7	947	8	(3.17, 96.83)	30.56
30	Yeast5	1484	8	(2.96, 97.04)	32.78
31	Ecoli0137vs26	281	7	(2.49, 97.51)	39.15
32	Yeast6	1484	8	(2.49, 97.51)	39.15
33	Abalone19	4174	8	(0.77, 99.23)	128.87

- *Pairwise comparisons.* We use Wilcoxon paired signed-rank test [54] to find out whether significant differences exist between a pair of algorithms.
- *Multiple comparisons.* We first use Friedman aligned-ranks test [26] to detect statistical differences among a set of algorithms. Then, if significant differences are found, we check if the control algorithm (the best one) is significantly better than the others (that is, $1 \times n$ comparison) using Holm post hoc test [27].

Moreover, we show the p-value for each comparison, which represents the lowest level of significance of a hypothesis resulting in a rejection. In such a manner, we are able to know how different two algorithms are.

As a complementary visualization tool, we consider the average aligned-ranks of each algorithm (used in the Friedman aligned-ranks test) in order to compare at first glance the behavior of each algorithm with respect to the others. These rankings are obtained computing the difference between the performance obtained by the algorithm and the mean performance of all algorithms in the corresponding data-set. These differences are ranked from 1 to $k \cdot n$ (being k the number of data-sets and n the number of methods), assigning the corresponding rank to the method from which the

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

difference has been computed. Hence, the lower the rank is, the better the method is. At last, the average ranking of each algorithm in all data-sets can be computed to show their global performance.

5. Empirical comparison: EUSBoost vs. state-of-the-art

In this section, we will compare our proposal to address the class imbalance problem using ensembles with the best performers state-of-the-art techniques. First in Section 5.1, we will investigate whether the modification of the fitness function to promote diversity (Section 3.3) is worth it or not, and we will check which of both modifications proposed works better. Then, we will compare the best proposal against the previously presented state-of-the-art methods in Section 5.2.

Before going through both analyses, we show the test AUC results of all the methods in each data-set in Table 5. In this table, and in the following analysis, EUSB corresponds to EUSBoost without diversity promotion, whereas $EUSB_Q$ and $EUSB_H$ stand for EUSBoost with the diversity mechanism with Q-statistic and the Hamming distance, respectively. Moreover, we append a "1" or a "4" after each abbreviation whenever it uses 10 or 40 classifiers, respectively.

5.1. Does promoting diversity help?

In this subsection, we analyze whether the factor accounting for diversity helps improving the performance of the ensemble or not. In addition, we analyze which of the proposed methods is the one with the best behavior.

In order to compare the three proposals, we use the Friedman aligned-ranks test, which allow us to confront all the methods by a unique test. Before carrying out the test, we show the average aligned-ranks in Fig. 2.

We observe that the average aligned-ranks of $EUSB1_Q$ are lower than the ranks of the others, which means that it is the best performer method. In this case, the promotion of the diversity seems to work as expected; however, the Hamming distance does not show the same behavior, performing similar to the original EUSBoost. The differences in terms of average aligned-ranks are contrasted by the corresponding Friedman aligned-ranks test, which outputs a p-value of 0.000003, and therefore the null hypothesis of equivalence is rejected. Thereby, we proceed with the Holm post hoc test to compare the control method ($EUSB1_Q$) with the other two methods (the results are shown in Table 6).

The Holm test verifies that $EUSB1_Q$ is significantly better than EUSB1 and $EUSB1_H$, which is supported by the low p-values obtained. Hence, we continue in the following subsection with EUSBoost using the diversity mechanism considering the Q-statistic. We can conclude

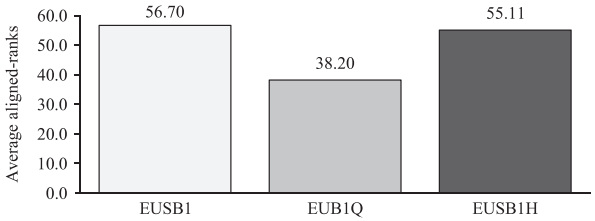

Fig. 2. Average aligned-ranks of EUSBoost ensembles.

Table 5
Detailed test results table of the tested algorithms. The best result in each data-set is stressed in bold-face.

Data-set	IR	Boosting-based		Bagging-based		Hybrid	EUSBoost-based		
		RUS1	SBO1	UB4	SBAG4	EASY	EUB1	$EUB1_Q$	$EUB1_H$
Glass04vs5	9.22	**0.9941**	0.9858	**0.9941**	0.9817	**0.9941**	**0.9941**	**0.9941**	**0.9941**
Ecoli0346vs5	9.25	0.9086	0.9005	0.9065	**0.9275**	0.8678	0.8955	0.9047	0.9065
Ecoli0347vs56	9.28	**0.8928**	0.8784	0.8803	0.8738	0.8659	0.8791	0.8902	0.8649
Yeast05679vs4	9.35	0.8032	0.7867	0.7949	**0.8144**	0.7702	0.7964	0.8067	0.7901
Ecoli067vs5	10.00	0.8792	0.8658	0.8858	0.8450	0.8567	0.8842	0.8825	**0.8900**
Vowel0	10.10	0.9509	**0.9887**	0.9470	0.9876	0.9449	0.9644	0.9537	0.9616
Glass016vs2	10.29	0.6179	0.5820	0.7331	0.6700	0.6595	0.6807	**0.7488**	0.7267
Glass2	10.39	0.6877	0.7501	0.7753	**0.8045**	0.7247	0.7068	0.7138	0.6949
Ecoli0147vs2356	10.59	0.8628	0.8883	0.8472	0.8895	0.8622	**0.8943**	0.8902	0.8749
Led7Digit02456789vs1	10.97	0.8763	0.7037	**0.8880**	0.8830	0.8734	0.8573	0.8552	0.8753
Ecoli01vs5	11.00	0.8705	0.8727	0.8265	0.8523	0.8910	0.8788	0.8932	**0.9235**
Glass06vs5	11.00	0.9916	0.9732	0.9139	0.9800	0.8470	0.9932	**0.9950**	0.9915
Glass0146vs2	11.06	0.7197	0.6780	0.7707	0.7501	0.7533	0.7603	**0.7736**	0.7589
Ecoli0147vs56	12.28	0.8642	0.8716	0.8565	0.8407	0.8411	0.8546	0.8858	**0.8924**
Cleveland0vs4	12.62	0.8238	0.8073	0.7753	0.7929	0.8022	0.8164	**0.8280**	0.7842
Ecoli0146vs5	13.00	**0.9295**	0.9045	0.8865	0.9147	0.8282	0.8891	0.8923	0.8872
Ecoli4	13.84	**0.9309**	0.9107	0.8899	0.9232	0.8782	0.8751	0.9273	0.9017
Shuttlec0vsc4	13.87	**1.0000**	**1.0000**	**1.0000**	0.9998	**1.0000**	**1.0000**	**1.0000**	**1.0000**
Yeastbc1vsc7	13.87	0.7552	0.7210	0.7580	0.6896	0.7167	0.7669	0.7651	**0.7771**
Glass4	15.47	0.8806	**0.9192**	0.8728	0.8862	0.8813	0.8700	0.9073	0.9042
Pageblocks13vs4	15.85	0.9684	0.9877	0.9790	0.9884	0.9711	0.9869	**0.9906**	0.9810
Abalone918	16.68	0.7276	0.7339	0.7302	0.7294	0.7223	0.7107	**0.7424**	0.6993
Glass016vs5	19.44	0.9743	0.9181	0.9429	0.8800	0.9524	0.9876	**0.9886**	0.9681
Shuttlec2vsc4	20.50	**1.0000**	**1.0000**	**1.0000**	**1.0000**	0.9905	**1.0000**	**1.0000**	**1.0000**
Yeast1458vs7	22.10	**0.6343**	0.5637	0.6135	0.6243	0.5800	0.6200	0.6282	0.5950
Glass5	22.81	0.9837	0.9789	0.9488	0.8915	0.9520	0.9870	**0.9878**	**0.9878**
Yeast2vs8	23.10	0.7995	0.7750	0.7651	**0.8019**	0.7320	0.7727	0.7614	0.7637
Yeast4	28.41	0.8377	0.7150	0.8478	0.7730	0.8317	0.8255	0.8311	**0.8489**
Yeast1289vs7	30.56	0.6907	0.6448	0.7407	0.6432	0.6798	**0.7491**	0.7132	0.7144
Yeast5	32.78	0.9492	0.9104	0.9546	**0.9661**	0.9502	0.9539	0.9382	0.9554
Ecoli0137vs26	39.15	0.8254	**0.8360**	0.7451	0.8306	0.7281	0.8099	0.8230	0.8039
Yeast6	39.15	0.8555	0.7966	**0.8678**	0.8387	0.8573	0.8601	0.8661	0.8623
Abalone19	128.87	0.6629	0.5252	**0.7081**	0.5602	0.6980	0.6730	0.6878	0.6663
Average		0.8530	0.8295	0.8499	0.8434	0.8334	0.8544	**0.8626**	0.8559

Table 6
Holm test results for the comparison among EUSBoost-based methods.

Control method: $EUSB_Q$ (38.20)					
i	Algorithm (Rank)	Z	p-Value	Holm	Hypothesis ($\alpha=0.05$)
2	EUSB1 (56.70)	2.616295	0.008889	0.05	**Rejected for $EUSB_Q$**
1	$EUSB1_H$ (55.11)	2.391307	0.016789	0.025	**Rejected for $EUSB_Q$**

from these tests that promoting diversity can enhance the results of the ensemble when the appropriate diversity measure is used, otherwise, the methodology might not be so effective.

5.2. EUSBoost vs. state-of-the-art

Hereafter, once we have shown and selected our best proposal, we analyze whether the proposed methodology provides more accurate ensembles than previous approaches, and more specifically, if the usage of a supervised approach (EUS) is able to outperform more random approaches based on random undersampling. To do so, we start showing the average aligned-ranks computed for Friedman aligned-ranks test in Fig. 3.

Looking at Fig. 3, we can observe that $EUSB1_Q$ excels, followed by the combinations of random undersampling with Boosting and Bagging, respectively. The worst performers are those approaches using SMOTE, which are not so accurate in highly imbalanced data-sets as well as the hybrid-based ensemble (EASY). In spite of the ranks, we must contrast these statements with the proper statistical test. The Friedman aligned-ranks test outputs a p-value of 0.000028, which indicates that the hypothesis of equivalence can be rejected with high confidence (significant differences exist). Hence, we continue with the Holm post hoc test (Table 7).

The Holm test brings out the good performance of $EUSB1_Q$ with highly imbalanced data-sets. $EUSB1_Q$ is able to statistically outperform all except RUS1 method, despite the low p-value obtained. For this reason, we get them into a pairwise comparison using the Wilcoxon test; in such a way, we can obtain a better insight of their performance when compared one versus the other (Table 8). Following its result, we can state that $EU1B_Q$ outstands in this framework; the test rejects the hypothesis of equivalence with a low p-value, and hence, being the ranks in favor of $EUSB1_Q$, its superiority is demonstrated. Therefore, the application of both the guided undersampling process (EUS) instead of random undersampling and the promotion of the diversity allows EUSBoost to statistically outperform the previous approaches in the framework of highly imbalanced data-sets.

In order to graphically show the advantage of $EUSB1_Q$ with respect to the others, we present in Fig. 4 a scatter plot where each point compares $EUSB1_Q$ with one of the other algorithms in a data-set. The x-axis position of the point is the AUC measure of $EUSB1_Q$, whereas the y-axis position is that of the other algorithm. Therefore, points that appear below the $y=x$ line correspond to data-sets where $EUSB1_Q$ performs better.

In Fig. 4, we can observe that when $EUSB1_Q$ performs better, it is usually much better than the other methods (besides, most of the points lie under the $y=x$ line), whereas when it performs worse, its loss is not large. Moreover, the greatest advantage seems to occur when the other algorithm in the comparison obtains a low AUC performance, which usually happens when the IR is higher. In these cases, focusing on obtaining more accurate base classifiers yet diverse makes a difference with respect to random techniques.

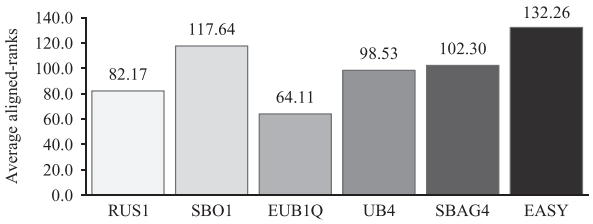

Fig. 3. Average aligned-ranks of the comparison between EUSBoost and the state-of-the-art ensemble methods.

Table 7
Holm test results for the comparison between $EUSB1_Q$ and the state-of-the-art ensemble methods.

Control method: $EUSB_Q$ (64.11)					
i	Algorithm (Rank)	Z	p-Value	Holm	Hypothesis ($\alpha=0.05$)
5	EASY (132.26)	4.83113	0.00000	0.01	**Rejected for $EUSB_Q$**
4	SBO1 (117.64)	3.79466	0.00015	0.0125	**Rejected for $EUSB_Q$**
3	SBAG4 (102.30)	2.70771	0.00678	0.01667	**Rejected for $EUSB_Q$**
2	UB4 (98.53)	2.44027	0.01468	0.025	**Rejected for $EUSB_Q$**
1	RUS1 (82.17)	1.28028	0.20045	0.05	Not rejected

Table 8
Wilcoxon tests to compare our proposal $EUSB1_Q$ with RUS1. R^+ corresponds to $EUSB1_Q$ and R^- to RUS1.

Comparison	R^+	R^-	Hypothesis ($\alpha=0.05$)	p-Value
$EUSB1_Q$ vs. RUS1	399.0	162.0	**Rejected for $EUSB1_Q$**	0.03327

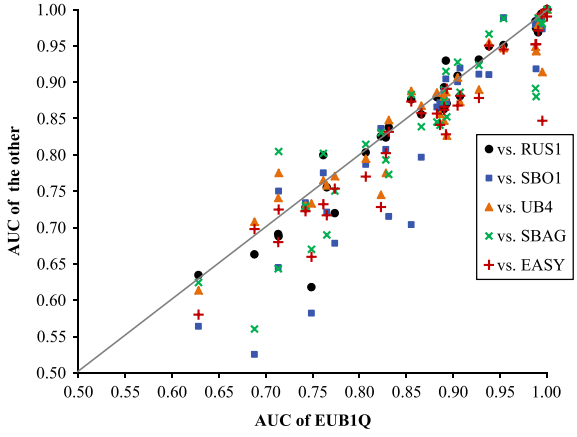

Fig. 4. Scatterplot for the comparison of the results of $EUSB1_Q$ and the other methods.

6. Kappa-AUC error diagrams: analyzing the behavior of EUSBoost

Kappa-error diagrams were suggested by Margineantu and Dietterich [38] to visualize the relation existing between the accuracy and diversity of the base classifiers of an ensembles. The error term is estimated using the accuracy rate, but dealing with imbalanced data-sets, accuracy is no longer meaningful and neither is the error (1−Acc). For this reason, we adapt kappa-error diagrams to the imbalance framework, which has not been

previously done, neither previous approaches have been analyzed with these diagrams [12,47,51].

These plots have been employed to analyze the behavior of different ensemble approaches (which assume balanced data distributions) to explain the differences in their performance [34,39,44]. A kappa-error diagram is a scatter plot where a point is plotted for each pair of base classifiers in the ensemble. The relation is studied in a pairwise manner since pairwise diversity measures are clear and intuitive, whereas their extension is not easy to establish [33,35]. An ensemble is depicted as a cloud of points in the scatter plot. The x-axis of the plot represents the kappa (κ) pairwise diversity measure, whereas the y-axis represents the mean error of both classifiers. In a similar way to that of the Q-statistic, κ can be easily computed for two classes; having two binary vectors (V_i, V_j) representing the outputs of the classifiers (0 or 1) for each instance, κ is computed as follows:

$$\kappa_{i,j} = \frac{2(N^{00}N^{11} - N^{01}N^{10})}{(N^{00}+N^{01})(N^{00}+N^{10}) + (N^{01}+N^{11})(N^{10}+N^{11})} \tag{10}$$

κ ranges from −1 to 1. Low values of κ mean high disagreement (diversity). Classifiers outputting the same class labels produce $\kappa = 1$, whereas statistically independent classifiers obtain $\kappa = 0$. Negative values accounts for negative correlation, which can be even better than independence for ensembles [36].

In other respects, the estimation of the pairwise error is computed as the mean error of both classifiers. However, dealing with imbalanced data-sets we have considered AUC measure. In order to maintain the similarity with kappa-error diagrams we have designed kappa-AUC error diagrams, where the AUC error is computed by 1−AUC, in such a way that we do not change the meaning of the diagrams, but only the measure used. Hence, the pairwise AUC error is computed as

$$\text{AUC error}_{ij} = 1 - \frac{\text{AUC}_i + \text{AUC}_j}{2} \tag{11}$$

where AUC_i and AUC_j are the individual AUC values of the classifiers. Ideally, points in the kappa-error diagram should reside in the bottom left, i.e., accurate (low error) and also diverse (low kappa value) pairs of classifiers. However, very accurate classifiers cannot be very diverse [31].

We have obtained the values for kappa-AUC error diagrams from the three runs of 5-fold cross-validation used in the experiments; hence, each ensemble cloud consists of 675 points (5 executions repeated 3 times and 45 pairs of classifiers in each one). Hereafter, we compare our proposed method EUSB1_Q against RUSBoost to understand its better behavior. Fig. 5 shows the kappa-AUC error diagrams for two of the 33 data-sets used in the experimental analysis.

In these specific data-sets, we can observe that EUSBoost is achieving what we have asked for; in both cases, the mean AUC error of the classifiers has decreased, while the diversity has been slightly reduced (the gravity centers of the clouds can be used as a global view). Nevertheless, only showing two diagrams might not be representative enough to reach meaningful conclusions. For this reason, we have also considered kappa-error movement and relative movement diagrams [39,44], once again using the AUC error, which serve as a summarization of all kappa-AUC error diagrams, since all the data-sets are presented in a unique plot.

The kappa-AUC error movement diagram in Fig. 6(a) is obtained by plotting the centers of all data-sets for each one of both methods; then, an arrow is depicted to connect the center points of both methods in the same data-set (in this case, from RUSB1 to EUSB1_Q). This way, the behavior of the new ensemble method with respect to the other can be analyzed, showing how the base classifiers have been displaced in the diagram. An arrow going to the left indicates that the new method creates more diverse base classifiers, in the same manner as an arrow going down means that the AUC error decreases. The number near the head of each arrow refers to the data-set which produces it, in the numbering shown in Table 4.

In addition, kappa-AUC error relative movement diagram (Fig. 6(b)) depicts the same arrows as Fig. 6(a), but in this case, the initial points of the arrows have been moved to the origin, and the head of the arrows correspond to the difference between the centers of kappa-AUC error diagrams. This way, even though the arrows have the same meaning, its analysis is easier, since the results in all data-sets become comparable. The diagram shows that, in the majority of the data-sets, EUSBoost obtains an advantage on AUC in exchange for losing some diversity (bottom right direction), but in some cases, diversity is also boosted (bottom left direction). Hence, EUS is doing the work as expected, although there are few exceptions where neither diversity nor AUC have been improved (the undesired top right direction). Nonetheless, as we have shown in Section 5, EUSBoost is able to statistically outperform RUSBoost.

7. Concluding remarks

We have presented a novel approach to enhance ensembles of classifiers, mainly those based on Boosting techniques, when dealing with highly imbalanced data-sets. AUC improvement comes from the application of EUS instead of random undersampling. In such a way,

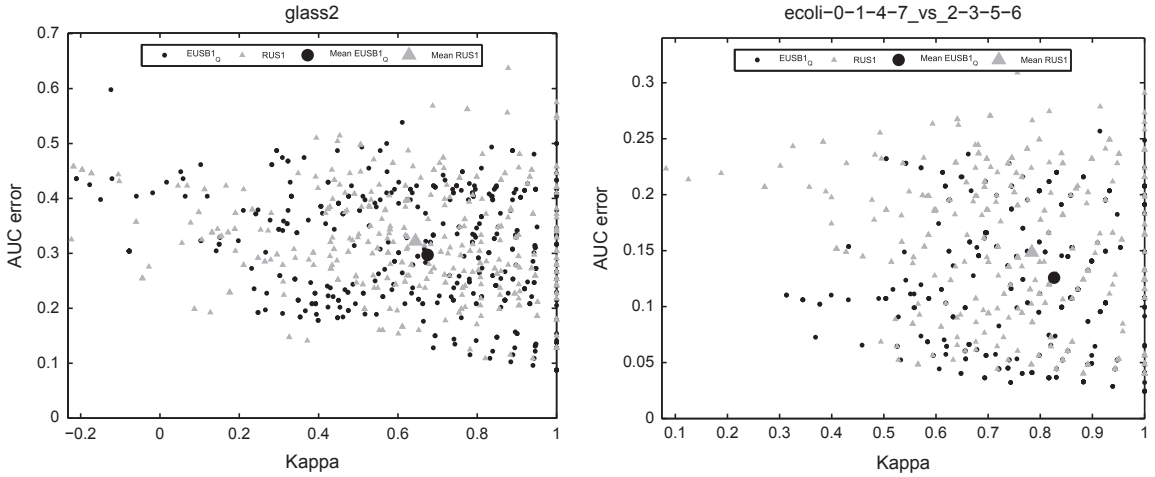

Fig. 5. Kappa-AUC error diagrams for two of the tested data-sets.

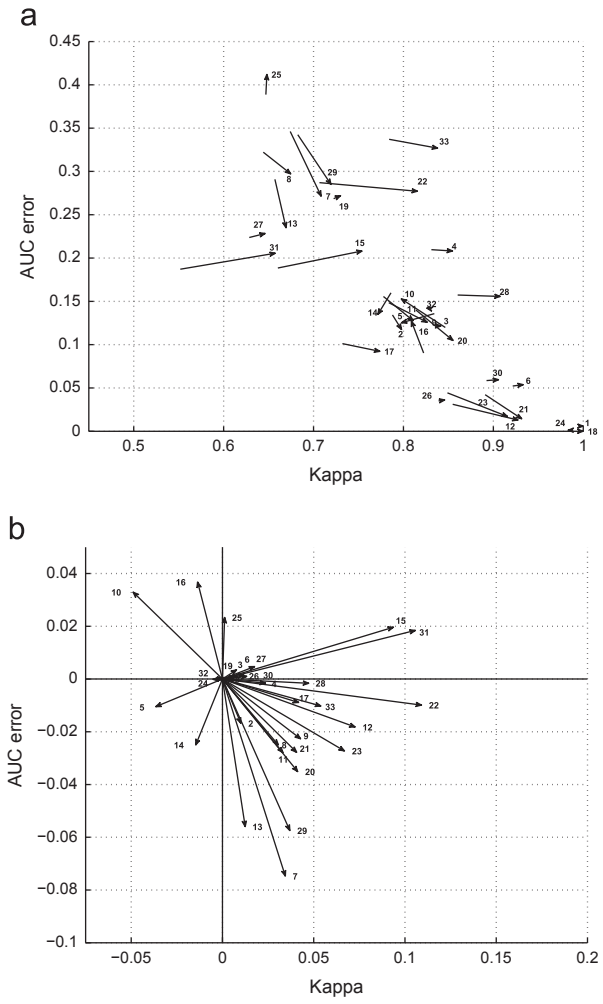

Fig. 6. Kappa-AUC error movement and relative movement diagrams comparing RUSBoost (the origin of the arrows) and EUSB1$_Q$ (the head of the arrows): (a) Kappa-AUC error movement diagram; and (b) Kappa-AUC error relative movement diagram.

we can obtain more accurate base classifiers than those obtained by RUSBoost. Moreover, the individual AUC gain is not sufficient itself, since diversity plays an important role in classifier ensembles. For this reason, we have proposed a mechanism embedded in EUS to promote diversity. We have shown that this mechanism using the appropriate diversity measure (e.g., the *Q*-statistic) is able to outperform the base EUSBoost (without enhancing diversity) and the state-of-the-art ensemble methods specifically designed for the class imbalance problem. For this reason, in future works we aim to test this approach with different diversity measures, including non-pairwise ones.

Finally, we have adapted kappa-error diagrams to the imbalance framework changing the computation of the error. This fact was not addressed yet, even though works on ensembles generally use these diagrams. In such a way, we have been able to analyze the advantages and disadvantages of EUSBoost in comparison to RUSBoost. The diagrams have shown, in accordance with previous studies [31], that the importance of the individual accuracy in ensembles might be greater than the influence of diversity, since EUSBoost has been able to outperform RUSBoost mainly exchanging not much diversity for accuracy, assessing more accurate ensembles.

Conflict of interest statement

None declared.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science under Projects TIN2010-15055 and TIN2011-28488 and the Andalusian Research Plan P10-TIC-6858.

References

[1] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.
[2] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2008) 307–318.
[3] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (3) (2003) 849–851.
[4] R. Barandela, R.M. Valdovinos, J.S. Sánchez, New applications of ensembles of classifiers, Pattern Analysis and Applications 6 (2003) 245–256.
[5] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behavior of several methods for balancing machine learning training data, SIGKDD Explorations Newsletter 6 (2004) 20–29.
[6] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.
[7] L. Breiman, Bagging predictors, Machine Learning 24 (1996) 123–140.
[8] G. Brown, J. Wyatt, R. Harris, X. Yao, Diversity creation methods: a survey and categorisation, Information Fusion 6 (1) (2005) 5–20, diversity in Multiple Classifier Systems.
[9] W.-L. Chao, J.-Z. Liu, J.-J. Ding, Facial age estimation based on label-sensitive learning and age-oriented regression, Pattern Recognition 46 (3) (2013) 628–641.
[10] N. Chawla, D. Cieslak, L. Hall, A. Joshi, Automatically countering imbalance and its empirical relationship to cost, Data Mining and Knowledge Discovery 17 (2008) 225–252.
[11] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligence Research 16 (2002) 321–357.
[12] N.V. Chawla, A. Lazarevic, L.O. Hall, K.W. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: Knowledge Discovery in Databases (PKDD'03), 2003, pp. 107–119.
[13] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[14] D. Drown, T. Khoshgoftaar, N. Seliya, Evolutionary sampling and software quality modeling of high-assurance systems, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 39 (5) (2009) 1097–1107.
[15] L.J. Eshelman, The CHC adaptive search algorithm: how to have safe search when engaging in nontraditional genetic recombination, in: G.J.E. Rawlins (Ed.), Foundations of Genetic Algorithms, Morgan Kaufmann, San Francisco, CA, 1991.
[16] A. Fernández, S. García, M.J. del Jesus, F. Herrera, A study of the behaviour of linguistic fuzzy rule based classification systems in the framework of imbalanced data-sets, Fuzzy Sets and Systems 159 (18) (2008) 2378–2398.
[17] A. Freitas, A. Costa-Pereira, P. Brazdil, Cost-sensitive decision trees applied to medical data, in: I. Song, J. Eder, T. Nguyen (Eds.), Data Warehousing and Knowledge Discovery, Lecture Notes in Computer Science, vol. 4654, 2007, pp. 303–312.
[18] Y. Freund, R.E. Schapire, A decision-theoretic generalization of on-line learning and an application to boosting, Journal of Computer and System Sciences 55 (1) (1997) 119–139.
[19] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 42 (4) (2012) 463–484.
[20] S. García, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 417–435.
[21] S. García, A. Fernández, 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, Information Sciences 180 (2010) 2044–2064.
[22] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[23] S. García, F. Herrera, Evolutionary undersampling for classification with imbalanced datasets: proposals and taxonomy, Evolutionary Computation 17 (2009) 275–306.
[24] V. García, R. Mollineda, J. Sánchez, On the *k*-nn performance in a challenging scenario of imbalance and overlapping, Pattern Analysis and Applications 11 (2008) 269–280.
[25] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.

[26] J.L. Hodges, E.L. Lehmann, Rank methods for combination of independent experiments in analysis of variance, Annals of Mathematical Statistics 33 (1962) 482–497.
[27] S. Holm, A simple sequentially rejective multiple test procedure, Scandinavian Journal of Statistics 6 (1979) 65–70.
[28] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.
[29] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis 6 (2002) 429–449.
[30] W. Khreich, E. Granger, A. Miri, R. Sabourin, Adaptive ROC-based ensembles of HMMs applied to anomaly detection, Pattern Recognition 45 (1) (2012) 208–230.
[31] L.Kuncheva, A bound on kappa-error diagrams for analysis of classifier ensembles, IEEE Transactions on Knowledge and Data Engineering 25 (3) (2013) 494-501.
[32] L. Kuncheva, C. Whitaker, C. Shipp, R. Duin, Limits on the majority vote accuracy in classifier fusion, Pattern Analysis and Applications 6 (2003) 22–31.
[33] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[34] L.I. Kuncheva, J.J. Rodriguez, Classifier ensembles with a random linear oracle, IEEE Transactions on Knowledge and Data Engineering 19 (4) (2007) 500–508.
[35] L.I. Kuncheva, C.J. Whitaker, Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy, Machine Learning 51 (2003) 181–207.
[36] L.I. Kuncheva, C.J. Whitaker, C.A. Shipp, R.P.W. Duin, Is independence good for combining classifiers? in: Proceedings of 15th International Conference on Pattern Recognition, 2000, vol. 2, 2000.
[37] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning, IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics 39 (2) (2009) 539–550.
[38] D.D. Margineantu, T.G. Dietterich, Pruning adaptive boosting, in: Proceedings of 14th International Conference on Machine Learning, ICML '97, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997.
[39] J. Maudes, J.J. Rodríguez, C. García-Osorio, C. Pardo, Random projections for linear SVM ensembles, Applied Intelligence 34 (2011) 347–359.
[40] I. Mukherjee, R.E. Schapire, A theory of multiclass boosting, Journal of Machine Learning Research 14 (2013) 437–497.
[41] R. Polikar, Ensemble based systems in decision making, IEEE Circuits and Systems Magazine 6 (3) (2006) 21–45.
[42] F. Provost, P. Domingos, Tree induction for probability-based ranking, Machine Learning 52 (2003) 199–215.
[43] J.R. Quinlan, C4.5: Programs for Machine Learning, 1st ed., Morgan Kaufmann Publishers, San Mateo-California, 1993.
[44] J.J. Rodríguez, C. García-Osorio, J. Maudes, Forests of nested dichotomies, Pattern Recognition Letters 31 (2) (2010) 125–132.
[45] J.J. Rodríguez, L.I. Kuncheva, C.J. Alonso, Rotation forest: a new classifier ensemble method, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (10) (2006) 1619–1630.
[46] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (2010) 1–39.
[47] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: a hybrid approach to alleviating class imbalance, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 40 (1) (2010) 185–197.
[48] J.A. Sáez, J. Luengo, F. Herrera, Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification, Pattern Recognition 46 (1) (2013) 355–364.
[49] Y. Sun, M.S. Kamel, A.K. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, Pattern Recognition 40 (12) (2007) 3358–3378.
[50] Y. Sun, A.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, International Journal of Pattern Recognition and Artificial Intelligence 23 (4) (2009) 687–719.
[51] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09), 2009.
[52] S. Wang, X. Yao, Relationships between diversity of classification ensembles and single-class performance measures, IEEE Transactions on Knowledge and Data Engineering 25 (1) (2013) 206–219.
[53] G.M. Weiss, F. Provost, Learning when training data are costly: the effect of class distribution on tree induction, Journal of Artificial Intelligence Research 19 (2003) 315–354.
[54] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.
[55] G. Wu, E. Chang, KBA: kernel boundary alignment considering imbalanced data distribution, IEEE Transactions on Knowledge and Data Engineering 17 (6) (2005) 786–795.
[56] 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, Knowledge and Information Systems 14 (2007) 1–37.
[57] Q. Yang, X. Wu, 10 challenging problems in data mining research, International Journal of Information Technology and Decision Making 5 (4) (2006) 597–604.
[58] G. Yule, On the association of attributes in statistics, Philosophical Transactions, A 194 (1900) 257–319.
[59] D. Zhang, M.M. Islam, G. Lu, A review on automatic image annotation techniques, Pattern Recognition 45 (1) (2012) 346–362.

Mikel Galar received the M.Sc. and Ph.D. degrees in Computer Science in 2009 and 2012, both from the Public University of Navarra, Pamplona, Spain. He is currently a Teaching Assistant in the Department of Automatics and Computation at the Public University of Navarra. His research interests are data-mining, classification, multi-classification, ensemble learning, evolutionary algorithms and fuzzy systems.

Alberto Fernández received the M.Sc. and Ph.D. degrees in computer science in 2005 and 2010, both from the University of Granada, Granada, Spain.

He is currently an Assistant Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. His research interests include data mining, classification in imbalanced domains, fuzzy rule learning, evolutionary algorithms, resolution of multi-classification problems with ensembles and decomposition techniques, and Business Intelligence in Cloud Computing.

He have been also awarded with the "Lofti A. Zadeh Prize" of the International Fuzzy Systems Association (IFSA) for the "Best paper on 2009-2010" for the work Hierarchical fuzzy rule based classification system with genetic rule selection for imbalanced data-sets.

Edurne Barrenechea is an Assistant Lecturer at the Department of Automatics and Computation, Public University of Navarra. She received an M.Sc. in Computer Science at the Pais Vasco University in 1990. She worked in a private company (Bombas Itur) as an Analyst Programmer from 1990 to 2001, and then she joined the Public University of Navarra as an Associate Lecturer. She obtained the Ph.D. in Computer Science in 2005 on the topic interval-valued fuzzy sets applied to image processing. Her publications comprise more than 20 papers in international journals and about 15 book chapters. Her research interests are fuzzy techniques for image processing, fuzzy sets theory, interval type-2 fuzzy sets theory and applications, decision making, and medical and industrial applications of soft computing techniques. She is a member of the board of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has been the supervisor of 28 Ph.D. students. He has published more than 240 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journal "Progress in Artificial Intelligence (Springer). He acts as an area editor of the International Journal of Computational Intelligence Systems and associated editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as a member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, IFSA 2013 Fellow, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008. Paper Award (bestowed in 2011), and 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association.

His current research interests include computing with words and decision making, bibliometrics, data mining, big data, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.