Information Sciences 325 (2015) 98–117

Contents lists available at ScienceDirect

Information Sciences

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

Diversity techniques improve the performance of the best imbalance learning ensembles

José F. Díez-Pastor[a,*], Juan J. Rodríguez[a], César I. García-Osorio[a], Ludmila I. Kuncheva[b]

[a] University of Burgos, Spain
[b] University of Bangor, UK

ARTICLE INFO

Article history:
Received 6 October 2014
Revised 15 May 2015
Accepted 4 July 2015
Available online 11 July 2015

Keywords:
Classifier ensembles
Imbalanced data sets
SMOTE
Undersampling
Rotation forest
Diversity

ABSTRACT

Many real-life problems can be described as unbalanced, where the number of instances belonging to one of the classes is much larger than the numbers in other classes. Examples are spam detection, credit card fraud detection or medical diagnosis. Ensembles of classifiers have acquired popularity in this kind of problems for their ability to obtain better results than individual classifiers. The most commonly used techniques by those ensembles especially designed to deal with imbalanced problems are for example Re-weighting, Oversampling and Undersampling. Other techniques, originally intended to increase the ensemble diversity, have not been systematically studied for their effect on imbalanced problems. Among these are Random Oracles, Disturbing Neighbors, Random Feature Weights or Rotation Forest. This paper presents an overview and an experimental study of various ensemble-based methods for imbalanced problems, the methods have been tested in its original form and in conjunction with several diversity-increasing techniques, using 84 imbalanced data sets from two well known repositories. This paper shows that these diversity-increasing techniques significantly improve the performance of ensemble methods for imbalanced problems and provides some ideas about when it is more convenient to use these diversifying techniques.

1. Introduction

The class imbalance problem[1] arises when one class has much more examples than the others [11].

Imbalance learning has attracted much attention because imbalanced data sets are common in real world problems like those related to security: spam detection [29], fraud detection [17], software defect detection [65]; biomedical data: finding the transition between coding and non-coding DNA in genes [28], mining cancer gene expression [70]; or financial data, for example, risk predictions in credit data [25].

Classification of imbalanced data is difficult because standard classifiers are driven by accuracy, hence the minority class may simply be ignored [58], besides generally all classifiers present some performance loss when the data is unbalanced [50]. In addition, many imbalanced datasets suffer problems related to its intrinsic characteristic. According to [44] there are at least six

* Corresponding author. Tel.: +34653030301.
E-mail address: jfdpastor@ubu.es (J.F. Díez-Pastor).

[1] Being aware of the terminological debate about "*unbalanced*" vs "*imbalanced*", we will use both words interchangeably. Our reason is that a keyword look-up should be able to retrieve this study, whichever word has been picked.

http://dx.doi.org/10.1016/j.ins.2015.07.025
0020-0255/

of these problems: overlapping [59], lack of density and information [66], noisy examples [8], small disjuncts [68], the significance of borderline instances to discriminate between positive and negative classes [47] and differences in the data distributions between training and test stages [54].

In [23], the approaches to dealing with unbalanced datasets are sorted into four categories[2]:

- **The algorithm-level** category encompasses modifications of existing general learning algorithms which bias the learning toward the minority class. Examples of this category are Hellinger Distance Decision Trees (HDDT) [14], Class Confidence Proportion Decision Tree (CCPDT) [42] and Significant, Positively Associated and Relatively Class Correlated Classification Trees (SPARCCC) [63], as well as other class-size insensitive decision trees. In other occasions misclassification costs are different for different examples, [71] presents decision tree and Naïve Bayesian learning methods that learns with unknown costs.
- **The data-level** category includes pre-processing algorithms that change the prior distribution of the classes either by increasing the number of minority class examples or by reducing the size of the majority class. In the first category of algorithms the simplest technique is to randomly add examples, without caring about neighbors from other class or the overlap between classes, some examples are Oversampling [4], SMOTE [10]. Other methods creates artificial instances taking into account these issues: Borderline-SMOTE [31], Safe-level SMOTE [9], ADASYN [32] or Cluster Based Oversampling [38]. In the second category Random Undersampling [3] removes random examples from the majority class and other methods like Edited Nearest Neighbor (ENN) [69] and Tomek Links [61] are based on data cleaning techniques.
- **The cost-sensitive** category contains methods that assign different costs for each class. Examples include AdaCost [20], AdaC1, AdaC2, and AdaC3 [60].
- **Classifier ensembles** [40,49] are combinations of several classifiers which are called base classifiers or member classifiers. Ensembles often give better results than individual classifiers. Although ensembles were not designed to work with imbalanced data, they have been successfully applied to this task through combination with processing techniques from the data-level category.

According to [23], the algorithm level and cost-sensitive approaches are more problem-dependent, whereas data level and ensemble learning approaches based on data processing are more versatile.

Ensemble methods for imbalanced learning tackle the imbalance problem using techniques like re-weighting, Oversampling and Undersampling. These preprocessing techniques attempt to train base classifiers with a less unbalanced dataset. These preprocessing techniques not only address the problem of imbalance, but add diversity, since each base classifier is trained on a different version of the data set. Diversity is one of the cornerstones of ensembles. An ideal ensemble system should have accurate individual classifiers and at the same time their errors should be in different instances. Several techniques have been developed to increase the diversity of an ensemble (see Section 2.4). In this paper we argue that techniques especially designed to increase diversity impact the performance of imbalance learning, significantly improving even the specific techniques.

To prove this claim, we conducted an experimental study where both classifiers especially designed for unbalanced sets and standard classifiers were tested in its original form and in conjunction with several diversity-increasing techniques. Finding that ensembles combined with diversity-increasing techniques ranked better than their original counterpart, even though the original version was specifically designed to work for imbalanced data. We also try to provide some clues when it is more appropriate to use diversity techniques using meta-learning, and evaluate the performance of the techniques in the presence of noisy and borderline examples.

The rest of the paper is structured as follows: Section 2 presents some background of ensemble learning, state-of-the-art techniques for imbalanced data, and our research hypothesis concerning diversity enhancing techniques. Section 3 shows the experimental study and results. Section 4 enumerates the findings extracted in the experimental study. And finally, in Sections 5 and 6 the conclusions and several future lines of research are presented.

2. Ensemble learning for imbalanced problems

In this section, the concept of ensemble and the importance of diversity will be introduced, then the preprocessing techniques and the ensembles methods for imbalanced problems used in this paper will be described. Finally, several techniques to increase the diversity in ensembles will be explained

2.1. Ensembles of classifiers

Ensemble of classifiers is combinations of multiple classifiers, referred as base classifiers. Ensembles usually achieve better performance than any of the single classifiers [40]. In order to build a good ensemble, it is necessary not only to build good base classifiers, also the base classifiers must be diverse, this means that for the same instance, the base classifiers return different outputs and their errors should be in different instances. Ensemble methods differ in the way they induce diversity between the base classifiers. The most common approach is modifying the training set for each member of the ensemble. In Bagging [6], each base classifier is obtained from a random sample of the training data. In the resampling version of AdaBoost [22], the data set for each subsequent ensemble member is drawn according to a distribution of weights over the data. The weights are modified

[2] Notice that these categories are not mutually exclusive, for example some cost-sensitive methods can be included in the classifier ensembles category.

depending on the correctness of the prediction given to the example by the previous classifier. In this way, the next classifier will give more importance to the difficult examples. MultiBoost [67] combines AdaBoost with Wagging (weighted bagging) [5], a variant of bagging which, instead of creating samples from the original dataset, it randomly modifies the weight associated to each instance.

2.2. Preprocessing techniques for imbalance learning

Preprocessing techniques aiming at balancing the class proportions can be easily embedded into an ensemble. The strategies are usually to increase the size of the minority class, to reduce the size of the majority class, or do both at the same time. While there are many variants of such preprocessing techniques, we will focus on:

- **Random undersampling**, that is done by eliminating random examples from the majority class. A drawback of this method is that, potentially, it can discard useful data. However, this adverse effect is minimized when using ensembles, since instances discarded in one iteration can remain in others. The most common implementation is to remove as much majority instances as necessary to match the size of the minority class. When random undersampling is used in this way in the experimental study, we name it RUS in the abbreviated names.
- **Random oversampling**, that creates copies of minority class instances randomly chosen. This method can lead to overfitting, since it creates copies of existing instances.
- **SMOTE**, that creates artificial instances for the minority class. To create an instance from an existing one, it randomly generates a synthetic example along the imaginary line that connects the instance with one of its k nearest neighbors from the same class. The amount of SMOTE, the number of generated synthetic instances, is a parameter of the method, whose value differs from one study to another. In this paper, we have used the two most commonly used configurations. These are to create as much artificial instances of the minority class as needed to double the minority class size (we call it 100% of SMOTE and we use the abbreviation SM100 in figures and tables), and to create as much artificial instances of the minority class as necessary to match the size of the majority class (we simply call this SMOTE, shortened as SM). SM100 only improves the balance ratio while SM equalizes the class proportions.
- **Random balance**, that takes into account the fact that the optimal amount of Undersampling and Oversampling/SMOTE is problem-specific and has considerable influence on the performance of the classifier. Random Balance [19] is designed to be used within an ensemble, to solve the above problem, it relies on the randomness and repetition. Random balance conserves the size of the dataset but varies the class proportions in the training sample of each base classifier using a random ratio. This includes the case where the minority class is over-represented and the imbalance ratio is inverted. SMOTE and random undersampling (resampling without replacement) are used to respectively increase or reduce the size of the classes to achieve the desired ratios. The procedure is simple, having a dataset S, with minority class S_P (subset of positive instances) and majority class S_N (subset of negative instances), it can be described as follows:
  1. A random number between 2 and $|S|-2$ is obtained. This number is going to be the new size of the majority class, *newMajSize*, and accordingly the new size of the minority class, *newMinSize*, will be $|S|-$*newMajSize*, so that the size of the new set, S', will be identical to the initial set ($|S|=|S'|$).
  2. If *newMajSize* $<|S_N|$, the new majority class, S'_N, is created by random sampling without replacement the original S_N so that its final size is $|S'_N|=$*newMajSize*, and the new minority class, S'_P, is obtained from S_P using SMOTE to get *newMinSize* $-|S_P|$ new artificial instances.
  3. Otherwise, S'_P is the class created as a random sample of S_P, and S'_N is the class grown by SMOTE from S_N, so that the final sizes are $|S'_P|=$ *newMinSize* and $|S'_N|=$ *newMajSize*.

2.3. Ensemble methods especially designed for imbalance

Sometimes ensembles not designed specifically for imbalance are used in unbalanced problems. A very common way of doing this is to combine them with one of the previous preprocessing techniques, for example, combining Bagging with Undersampling as it is done in [3]. Other times the method is especially designed to deal with unbalanced datasets. This section lists some of the most prominent methods specifically designed to deal with unbalanced.

- **SMOTEBagging** [64] is similar to Bagging except that each classifier is built with a data set with classes of equal size. The data set in each iteration is composed as follows. A bootstrap sample is taken from the majority class, keeping the original size, say N. The sample of size N for the minority class is created through a combination of Oversampling and SMOTE. The Oversampling percentage varies in each iteration, ranging from 10% in the first iteration to 100% in the last, always being multiple of ten. The rest of the positive instances are generated by SMOTE.
- **SMOTEBoost** [12] is a modification of the re-weighting version of AdaBoost.M2. After each boosting round, SMOTE is applied in order to create new synthetic examples from the minority class. The synthetic instances always have the same weight, the weight of the instances in the original dataset, while the originals have weights that are updated according to a pseudo-loss function. Those instances that are difficult for the previous classifiers have bigger weights.
- **RAMOBoost** [13] is inspired by SMOTEBoost and ADASYN [32] algorithms. The main difference with SMOTEBoost is the way it creates positive instances. While SMOTE, used within SMOTEBoost, creates instances uniformly, in RAMOBoost there exists a

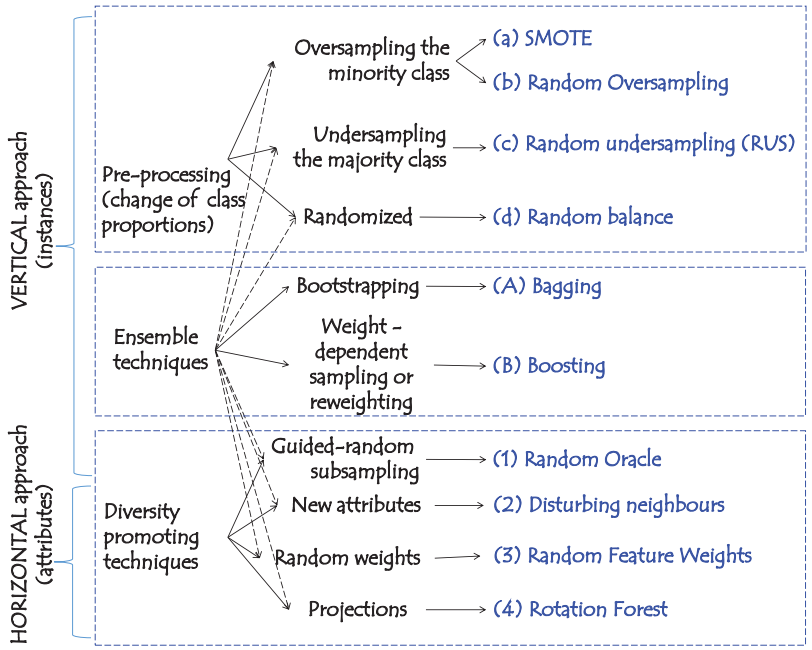

Fig. 1. Ensemble diversifying heuristics based on data manipulation.

sampling distribution based in the underlying data distribution. As a result, the artificial instances are created on the difficult regions of the decision boundary.

- **RUSBoost** [57] works similarly to SMOTEBoost. This time a Random undersampling is applied after each boosting round, removing instances from the majority class. In this case, there are no new instances. It is only necessary to normalize the weights of the instances in the processed dataset with regard to the total sum of weights in the original dataset.
- **EUSBoost** [24] is based on RUSBoost and it aims to improve the original method by using evolutionary Undersampling. EUSBoost also tries to promote diversity using different subsets of majority class instances to train each base classifier in each iteration.
- **EasyEnsemble** [43] samples several subsets from the majority class, trains an AdaBoost ensemble using repeatedly each of these subsets, and combines the outputs of those classifiers.
- **Random balance-boost** [19] follows the same philosophy as SMOTEBoost and RUSBoost. Each base classifier is trained with a dataset obtained through Random balance. The number of instances removed by Undersampling is equal to the number of instances introduced by SMOTE. As in SMOTEBoost, synthetic instances are generated with a weight proportional to the total number of instances. The combination of SMOTE, UnderSampling and Re-weighting provides more diversity which generally leads to better performance in ensemble learning.

2.4. Diversity-enhancing techniques

Diversity is essential in order to build an accurate ensemble of classifiers.

Fig. 1 summarizes some widely used diversifying heuristics for building classifier ensembles based on manipulating of the training data.

Diversity is naturally promoted by Oversampling, Undersampling or Re-weighting. Approaches which do not specifically target imbalance are often overlooked in imbalanced learning, in spite of their marked success in general multi-class classification or regression. Among these, the vertical approach methods (A) and (B) are commonly used to introduce diversity in ensembles. Here we propose that ensemble creation methods may offer more to imbalanced learning than what has already been achieved. To this end, we examine four further techniques illustrated in the lower part of Fig. 1, called diversity-enhancing techniques; these are detailed below.

We decided on the following approaches:

- **Guided random sampling.** In the Random Oracle ensemble [41], for each iteration, the instances are divided into two groups using a random hyperplane, and then a classifier is built for each group. Ensemble methods based on random partitioning the training set into several samples are often used for scaling up classifiers in large databases. [52].
- **New attributes.** Some methods, like Disturbing neighbors [45], expand the feature space with attributes that are not originally present in the dataset. For each base classifier in the ensemble, disturbing neighbors uses N randomly selected instances, the disturbing neighbors, to train a 1-NN (Nearest Neighbors) classifier, then it creates N binary attributes for each instance

(with value 1 if the corresponding disturbing neighbor is the closest to the instance, 0 otherwise) and an additional attribute whose value is the class predicted by the 1-NN classifier, hence, the feature space is expanded with N+1 new attributes.

- **Random weights.** It is possible to introduce diversity by giving a different importance/weight to every attribute for each member of the ensemble. Random Subspaces [33] can be viewed as a special case of this approach. A set of binary random weights is used, where a weight of value 0 means that the attribute is not included in the subset for the respective ensemble member. The result is that different classifiers are constructed using different subsets of the attributes. The Random Forest ensemble [7] is a bagging ensemble with random trees. A random tree differs from the standard tree only by its training. In random trees, a random subset of attributes is considered for the splitting of each node. This can be seen as a variation of Random Subspaces but instead of using the same subset for the whole tree the set is different in each node. Proposed more recently, the Random Feature Weights ensemble [46] associates a vector of weights with each tree of the ensemble. This vector is used to modify the way in which the merit function of the attributes is calculated. For a training set D and a weight vector $\mathbf{w}$, the new merit function for attribute a_i is defined as $f_{\mathbf{w}}(a_i, D) = w_i f(a_i, D)$, where $f(a_i, D)$ denotes the original merit function of a_i for D. Thus, the method introduces a bias which favors the selection and use of attributes with higher associated weight. The vector of weights is randomly drawn for each tree in the ensemble, thereby introducing diversity. These weights are real-valued (not just 0 and 1, as in Random subspaces and Random forest) so it is possible to draw a parallel with Wagging, but using features instead of instances.
- **Projections.** These are frequently used to reduce the dimensionality of the data. In the context of ensemble learning, projecting can be construed as a diversifying heuristic. A well known example is Rotation Forest [55], an ensemble method for decision trees which uses principal component analysis (PCA) to project different groups of attributes for each base classifier. A similar approach but using supervised projections is Boosting Projections [26,27]. Random projections have been used to provide an extra diversity and embed the original set into a space of lower dimension [56]. Random subspaces can be also seen as a special case of Random projections.

The preprocessing techniques surrounded by a dashed line in the upper part of Fig. 1 are specifically designed for dealing with imbalanced data. Each of these techniques alone can be used for creating an ensemble (indicated by the dashed arrows).

Here we are interested in finding out whether the preprocessing techniques (a–d) can be enhanced by (A–B), as well as other diversity heuristics (1–4), to produce better ensembles. Many of the techniques can be used in combination, both within their own group (for example, technique (d) described in 2.2 combines (a) with (c)) and with techniques from different groups (Section 2.3 describes some ensemble learning methods that combine preprocessing techniques (a–d) with (A–B) sampling and re-weighting techniques). In the experimental study we have used a representative from each diversity heuristic: from (1), Random Oracle, from (2), Disturbing neighbors, from (3), Random Feature Weights, and from (4), Rotation Forest.

3. Experimental set-up and results

We intend to demonstrate that techniques for promoting diversity in classifier ensembles enhance the performance of bespoke state-ot-the-art ensembles for imbalance learning. The structure of this section is: firstly, the ensemble methods used in the experiments, along with their basic parameters and some clarifications of its operation are listed (Section 3.1). Secondly, the datasets used, along with their basic characteristics are shown (Section 3.2). Then (in Section 3.3), will be carried out a comparison between the ensembles and the same group of ensembles combined with the technique used to increase the diversity listed in Section 2.4. The effect of the size ensemble is studied in Section 3.4.

Some ideas about when it is more appropriate to use diversity techniques taking into account complexity measures of the datasets are provided in Section 3.5, and finally Section 3.6 explores the effect of Disturbing Neighbors in presence of noisy and borderline instances.

3.1. Ensemble methods tested in the experimental set-up

Ensembles which only use techniques like Reweighing, Resampling, Oversampling and Undersampling will be called *basic ensembles*. Fig. 2 displays the collection of basic ensemble methods examined in this study. Each of the following methods will be combined with each of the diversity-enhancing techniques described in Section 2.4, these methods will be called *enhanced ensembles* (for example, RUSBoost (RUSBo) is considered a *basic ensemble*, while RUSBoost combined with Disturbing Neighbors (DN+RUSBo) is considered an *enhanced ensemble*). Finally ensembles using only the diversity-enhancing techniques (no resampling or other preprocessing) were also tested.

These *basic* methods can be grouped into three different categories: x) baseline methods which are not modified in any way to cope with imbalanced data, y) ensemble methods especially designed to cope with imbalanced data, and z) baseline methods combined with preprocessing techniques to improve its performance in unbalanced datasets. In these methods, the abbreviation used contains the name of the family of ensembles (e.g. E: Simple Ensemble, Ba: Bagging and Bo: Boosting),[3] and

[3] In a method belonging to the *Simple Ensemble family*, the dataset used to build each member of the ensemble is built using only the preprocessing technique (the source of diversity in these ensembles is the preprocessing method, since SMOTE, RUS and Random Balance are randomized methods that produce a different dataset at each iteration), while in a method that is a member of the *Bagging family* or the *Boosting family* the dataset used to build each base classifier comes from using resampling and then the preprocessing technique.

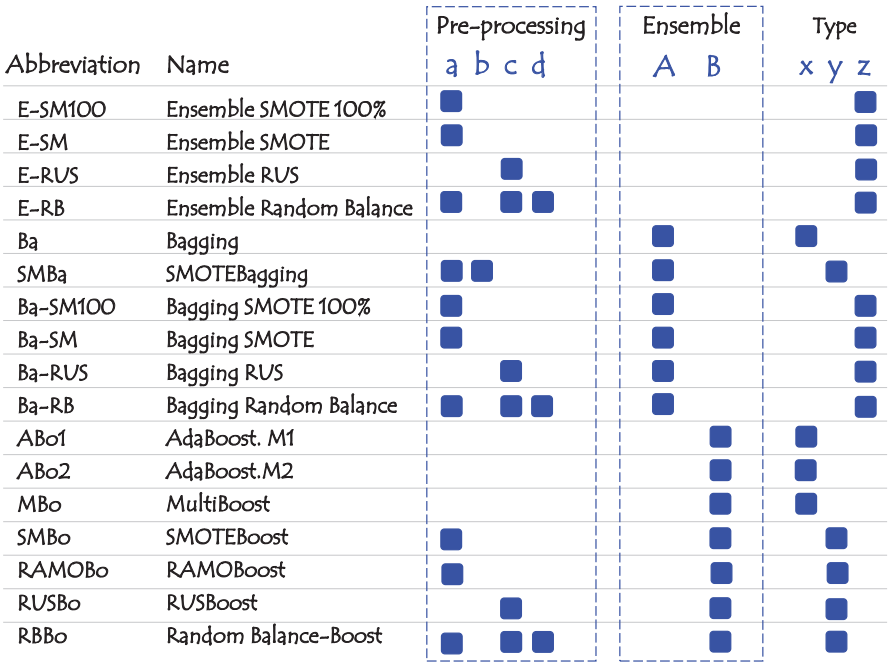

Fig. 2. Ensemble methods compared in this study.

the preprocessing techniques (SM/SM100: SMOTE, RUS: Random Undersampling, RB: Random Balance). These methods are listed below:

1. E-SM100: SMOTE is used, in each iteration, to double the size of the minority class. The number of neighbors is 5 for all methods that use SMOTE.
2. E-SM: SMOTE is used, in each iteration, to get a minority class of the same size as the majority class.
3. E-RUS: Random Undersampling is used, in each iteration, to reduce the majority class so that equal in size to that of the minority class.
4. E-RB: Radom Balance is used in each iteration.
5. Ba: Bagging.
6. SMBa: SMOTEBagging.
7. Ba-SM100: Bagging in which in each iteration SMOTE is used to double the size of the minority class.
8. Ba-SM: Bagging in which in each iteration SMOTE is used to get a minority class of the same size as the majority class.
9. Ba-RUS: Bagging in which in each iteration Random Undersampling is used to reduce the size of the majority class to the size of the minority class. This method is called UnderBagging in [3,23].
10. Ba-RB: Bagging with Random Balance in each iteration.
11. ABo1: AdaBoost.M1.
12. ABo2: AdaBoost.M2.
13. MBo: MultiBoost.
14. SMBo: SMOTEBoost with the following settings: SMOTE 100% in each iteration.
15. RAMOBo: RAMOBoost with the following settings: number of synthetic instances equal to the size of the minority class, number of neighbors used to create synthetic instances equal to 5, number of neighbors used to compute probabilities equal to 10.
16. RUSBo: RUSBoost.
17. RBBo: Random Balance Boost.

The size of the ensembles was set to 100. Default Weka parameters were used in all the ensemble methods provided by the library, except the number of sub committees in multiboost that was set to 10. In a further section (Section 3.4) the effect of the size of the ensemble is studied.

The classifier used as a base classifier in all ensembles was J48, the Java implementation of Quinlan's C4.5 [53]. As recommended for imbalanced data [15], it was used with Laplace smoothing at the leaves, but without pruning and collapsing. When C4.5 is used with this configuration, it is called C4.4 [51].

Table 1
Characteristics of the 20 data sets from the HDDT collection. Column #E shows the number of examples in the dataset, column #A the number of attributes, both numeric and nominal in the format (numeric/nominal), and column IR the imbalance ratio (the number of instances of the majority class per instance of the minority class).

Data set	#E	#A	IR	Data set	#E	#A	IR
Boundary	3505	(0/175)	27.50	Ism	11180	(6/0)	42.00
Breast-y	286	(0/9)	2.36	Letter	20000	(16/0)	24.35
Cam	18916	(0/132)	19.08	Oil	937	(49/0)	21.85
Compustat	13657	(20/0)	25.26	Optdigits	5620	(64/0)	9.14
Covtype	38500	(10/0)	13.02	Page	5473	(10/0)	8.77
Credit-g	1000	(7/13)	2.33	Pendigits	10992	(16/0)	8.63
Estate	5322	(12/0)	7.37	Phoneme	5404	(5/0)	2.41
German-numer	1000	(24/0)	2.33	PhosS	11411	(480/0)	17.62
Heart-v	200	(5/8)	2.92	Satimage	6430	(36/0)	9.29
Hypo	3163	(7/18)	19.95	Segment	2310	(19/0)	6.00

3.2. Datasets and tools

Two collections of data sets were used. The HDDT collection[4] contains 20 binary imbalanced data sets used in [15]. Table 1 shows the characteristics of this dataset collection.

The KEEL collection[5] contains 66 binary imbalanced data sets (we have used 64 of them in the experiments, because two of them were almost identical to two in the other repository) from the repository of KEEL [1]. Datasets in the KEEL collection are not completely independent of each other. Several of them are variants of the same original dataset. Starting with a single multiclass dataset, several binary datasets are created by grouping their classes in different ways. Table 2 shows the characteristics of this dataset collection.

Many data sets in these two collections are available or are modifications of data sets in the UCI Repository [2].

Weka 3.7.10 [30] was used for the experiments. The results were obtained with a 5×2-fold cross validation [18].

Three criteria were used for evaluating the ensemble performance: The *F*-measure [62], the Geometric Mean [39] and the Area Under the ROC Curve (AUC) [21].

Given a test dataset, containing P examples of the positive class and N examples of the negative class. The confusion matrix is shown in Table 3.

The True Positive Rate (TPR) also named Sensitivity or Recall in some fields, is defined as TP/P, and False Positive Rate (FPR) is defined as FP/N. The precision is defined as $TP/(TP + FP)$.

Using these previous measures it is possible to define the F-measure as

$$FMeasure = 2 \times \frac{precision \times recall}{precision + recall}$$

The Geometric Mean is defined as

$$GMean = \sqrt{TP/P \times TN/N}$$

In this work the ROC curve is obtained from the probabilities assigned to the instances by the classifier, each probability threshold gives a TPR and FPR that defines a point in the curve. The AUC is computed from Wilcoxon rank s um test statistic.

3.3. Comparison between basic and enhanced ensembles

This section will show a summary of the results for each of the basic ensembles and their improved versions and a comparison between these methods is performed. The summary is done by averaging, for each method the score across the 84 data sets.

Averaging the results is not the best way to compare multiple methods, since a big difference in a dataset can mask a general trend in the rest. To compare multiple methods, we used average ranks [16]. Each method was assigned a rank for each data set based on its performance, separately for each criterion. The best method obtained rank 1, the second best obtained rank 2, etc. When there was a tie, the ranks were shared out. For example, if the top three methods for a given data set tied, each one of them would receive rank $(1 + 2 + 3)/3 = 2$ for this data set. The ranks are averaged across all datasets. The methods were arranged by their ranks, where the best methods (the ones with the lowest average ranks) were at the top of the list. Iman and Davenport's [37] test was applied to check whether there are any significant differences between the ranks of the compared methods. Subsequently, Hochberg's test [36] was carried out next to identify all methods which were not significantly different from the winner.

This section of the experiments consisted of two parts:

[4] Available at http://www.nd.edu/~dial/hddt/.
[5] Available at http://sci2s.ugr.es/keel/imbalanced.php.

Table 2
Characteristics of the data sets from the KEEL collection. Column #E shows the number of examples in the dataset, column #A the number of attributes, both numeric and nominal in the format (numeric/nominal), and column IR the imbalance ratio (the number of instances of the majority class for each instance of the minority class).

Data set	#E	#A	IR	Data set	#E	#A	IR
Abalone19	4174	(7/1)	129.44	Glass4	214	(9/0)	15.46
Abalone9-18	731	(7/1)	16.40	Glass5	214	(9/0)	22.78
Cleveland-0_vs_4	177	(13/0)	12.62	Glass6	214	(9/0)	6.38
Ecoli-0-1-3-7_vs_2-6	281	(7/0)	39.14	Haberman	306	(3/0)	2.78
Ecoli-0-1-4-6_vs_5	280	(6/0)	13.00	Iris0	150	(4/0)	2.00
Ecoli-0-1-4-7_vs_2-3-5-6	336	(7/0)	10.59	Led7digit-0-2-4-5-6-7-8-9_vs_1	443	(7/0)	10.97
Ecoli-0-1-4-7_vs_5-6	332	(6/0)	12.28	New-thyroid1	215	(5/0)	5.14
Ecoli-0-1_vs_2-3-5	244	(7/0)	9.17	New-thyroid2	215	(5/0)	5.14
Ecoli-0-1_vs_5	240	(6/0)	11.00	Page-blocks-1-3_vs_4	472	(10/0)	15.86
Ecoli-0-2-3-4_vs_5	202	(7/0)	9.10	Pima	768	(8/0)	1.87
Ecoli-0-2-6-7_vs_3-5	224	(7/0)	9.18	Shuttle-c0-vs-c4	1829	(9/0)	13.87
Ecoli-0-3-4-6_vs_5	205	(7/0)	9.25	Shuttle-c2-vs-c4	129	(9/0)	20.50
Ecoli-0-3-4-7_vs_5-6	257	(7/0)	9.28	Vehicle0	846	(18/0)	3.25
Ecoli-0-3-4_vs_5	200	(7/0)	9.00	Vehicle1	846	(18/0)	2.90
Ecoli-0-4-6_vs_5	203	(6/0)	9.15	Vehicle2	846	(18/0)	2.88
Ecoli-0-6-7_vs_3-5	222	(7/0)	9.09	Vehicle3	846	(18/0)	2.99
Ecoli-0-6-7_vs_5	220	(6/0)	10.00	Vowel0	988	(13/0)	9.98
Ecoli-0_vs_1	220	(7/0)	1.86	Wisconsin	683	(9/0)	1.86
Ecoli1	336	(7/0)	3.36	Yeast-0-2-5-6_vs_3-7-8-9	1004	(8/0)	9.14
Ecoli2	336	(7/0)	5.46	Yeast-0-2-5-7-9_vs_3-6-8	1004	(8/0)	9.14
Ecoli3	336	(7/0)	8.60	Yeast-0-3-5-9_vs_7-8	506	(8/0)	9.12
Ecoli4	336	(7/0)	15.80	Yeast-0-5-6-7-9_vs_4	528	(8/0)	9.35
Glass-0-1-2-3_vs_4-5-6	214	(9/0)	3.20	Yeast-1-2-8-9_vs_7	947	(8/0)	30.57
Glass-0-1-4-6_vs_2	205	(9/0)	11.06	Yeast-1-4-5-8_vs_7	693	(8/0)	22.10
Glass-0-1-5_vs_2	172	(9/0)	9.12	Yeast-1_vs_7	459	(7/0)	14.30
Glass-0-1-6_vs_2	192	(9/0)	10.29	Yeast-2_vs_4	514	(8/0)	9.08
Glass-0-1-6_vs_5	184	(9/0)	19.44	Yeast-2_vs_8	482	(8/0)	23.10
Glass-0-4_vs_5	92	(9/0)	9.22	Yeast1	1484	(8/0)	2.46
Glass-0-6_vs_5	108	(9/0)	11.00	Yeast3	1484	(8/0)	8.10
Glass0	214	(9/0)	2.06	Yeast4	1484	(8/0)	28.10
Glass1	214	(9/0)	1.82	Yeast5	1484	(8/0)	32.73
Glass2	214	(9/0)	11.59	Yeast6	1484	(8/0)	41.40

Table 3
Confusion matrix in binary problems.

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

1. **Basic ensembles vs enhanced variants and enhanced variants among themselves.** Each basic ensemble was compared with its 4 enhanced variants, where each variant was obtained by applying the respective diversity-enhancing method. The average ranks were calculated using five methods (four when comparing diversity techniques among themselves).
   Moreover, due to most of the enhanced versions make use of some type of preprocessing, average ranks were calculated using all of the methods that use the same diversity technique. The 17 ensemble variants and a basic ensemble that only use the diversity technique.
2. **The overall winner.** In one final comparison, we select the method with best average rank for each row in Figs. 5(a), 6(a) and 7(a), and calculated the ranks using these best methods.

3.3.1. Basic ensembles versus enhanced variants and enhanced variants among themselves

The way in which the combination of the ensemble methods and the diversity technique is performed is always the same. In each iteration a modified training set resulting from the application of Sampling, Oversampling, Undersampling, Reweighting or the combination of two of these techniques (depending on the ensemble type) is created. This modified training set is modified again using a diversity technique for Random Linear Oracles, Disturbing neighbors and Rotation Forest. The method Random Feature Weights does not modify the dataset, it uses a modified decision tree.

Some of the diversity techniques have parameters. Rotation Forest has been used with the default parameters found in Weka package. In the case of Disturbing Neighbors, as the authors of the method proposed, the dimensions used to compute the Nearest Neighbor are randomly selected, choosing 50% of the attributes. The number of "Disturbing Neighbors" selected in each iteration is $N = 10$. The Random Feature Weights implementation is based on J48 and is used with the C4.4 configuration. The exponent

value, which control the level of randomness, was set to 1 (the value used by the authors when combined with another methods). Random Linear Oracles do not have any parameter.

Fig. 3 shows a comparison of the basic ensemble methods with the same methods augmented with Random Oracles (1st row), Disturbing Neighbors (2nd row), Random Feature Weights (3rd row) and Rotation Forest (4th row). There are three columns of graphs, one for each performance measure. Each point in the graph corresponds to a pair of basic ensemble method and dataset. The x coordinate is the value of the performance measure for the basic ensemble while the y coordinate is the value for that method combined with the diversity technique. Points above the diagonal are the cases when the combination is better than the basic ensemble. The graphs also show the percentage of points above and below the diagonal. In the graphs the majority of points are above the diagonal. Nevertheless, the performance measures show different behaviors, especially for Rotation Forest: for AUC the advantage of Rotation Forest is clear. For F-measure and G-mean, although the majority of the points are above the diagonal, there are some points were Rotation Forest is clearly worse. These points are relatively few, if we consider that each graph has 1428 points (17 basic ensemble methods multiplied by 84 datasets).

Fig. 4 shows the average scores across the 84 datasets for the AUC, F-Measure and G-Mean. The way to interpret the tables is as follows: each line from the second onwards contains the values of a *basic ensemble* and the enhanced versions of this ensemble. The column indicates the diversity-enhancing strategy used. So, for example, the intersection of column RFW and row Ba contains the scores of Bagging combined with Random Feature Weights. Ensembles using only the diversity-enhancing techniques (no resampling or other preprocessing) are shown in the first row. The full table of results can be consulted in the supplementary material.[6] For the AUC it is clear that enhanced methods, especially those that use Rotation Forest obtain better average results. One trend that also happens when considering the F-Measure. For the G-Mean, the best average results are obtained by methods that use Random Undersampling as a preprocessing technique (Ba-RUS and E-RUS). For this measure, although the basic methods are improved with the diversity techniques, the selection of the basic ensemble method (row) has more influence than the diversity-enhancing technique (column).

Figs. 5, 6 and 7 show the average ranks calculated from the area under the curve, the F-Measure and the G-Mean, respectively.

The structure of the three figures is the same, on the left, the ranks are calculated by rows. The possible ranks were from 1 to 5 (the basic ensemble method and its four combinations with diversity techniques) or 1 to 4 when the diversity techniques are not combined with any other ensemble or preprocessing technique. Again, rank 1 would correspond to the best alternative, and rank 5, to the worst. The intensity of the cell reflects the average rank. Cells with lower ranks (better) are filled in light gray and those with higher ranks (worse) are filled in dark gray. It is easy to see that, in general, the methods that use a diversity enhancing techniques perform better than those without.

The best method in each row has its rank in brackets. Those methods that are equivalent to the best one at significance 0.05 (Hochberg's test [36]) are delimited in parentheses.

Note that there are no entries in parentheses in the first column in Fig. 5(a). This means that, for the AUC, all *basic ensembles* perform significantly worse than the best enhanced variant. For the F-Measure and G-Mean criteria, improvement does not happen for all the methods but it is clearly noticeable for the methods of interest, the best methods according to this measure.

The way to know which are the best basic ensembles is through the average ranks calculated column-wise instead of row-wise, on the first column in the right table (Figs 5(b), 6(b) and 7(b)). Again the best method in each column has its rank in brackets.

The best basic methods according the F-Measure are RBBo, Ba-RB, RUSBo, and Ba-SM and these methods are clearly improved when combined with Rotation Forest strategy, as seen in Fig. 6(a).

For average ranks computed using G-Mean, the top methods are Ba-RUS, E-RB and RUSBo and these methods are improved when combined with Random Oracles and Disturbing Neighbors.

3.3.2. The overall winner

A new average rank has been calculated from the best combinations in each row in Figs. 5(a), 6(a) and 7(a).

The results are shown in Table 4. The classic Rotation Forest on its own, alongside several methods combined with Rotation Forest monopolize the best positions in the AUC table.

For the F-Measure, the basic Rotation Forest does not achieve a good position, although various combinations of Rotation Forest with other ensembles still occupy the top positions. The best combinations use Oversampling strategies trying to obtain more balance.

This difference may be due to the fact that these two measures consider different aspects: the AUC only takes into account the probabilities given by the classifier for each instance, while the F-Measure takes into account whether the instances are correctly labeled or not. One classifier could achieve the maximum possible AUC and simultaneously the minimum F-Measure, which happens when none of the instances of the positive class are correctly labeled, but the probabilities assigned to the instances of the positive class are higher than the probabilities of instances belonging to the negative class. This could be the reason why balancing strategies have less impact when considering the AUC as an evaluation measure.

For the G-Mean, there is no clear trend. Although it seems that this time ensembles enhanced with Random Linear Oracles and Disturbing Neighbors outperform those enhanced with Rotation Forest.

[6] https://github.com/joseFranciscoDiez/research/wiki/Supplementary-Material–Diversity-imbalanced-learning.

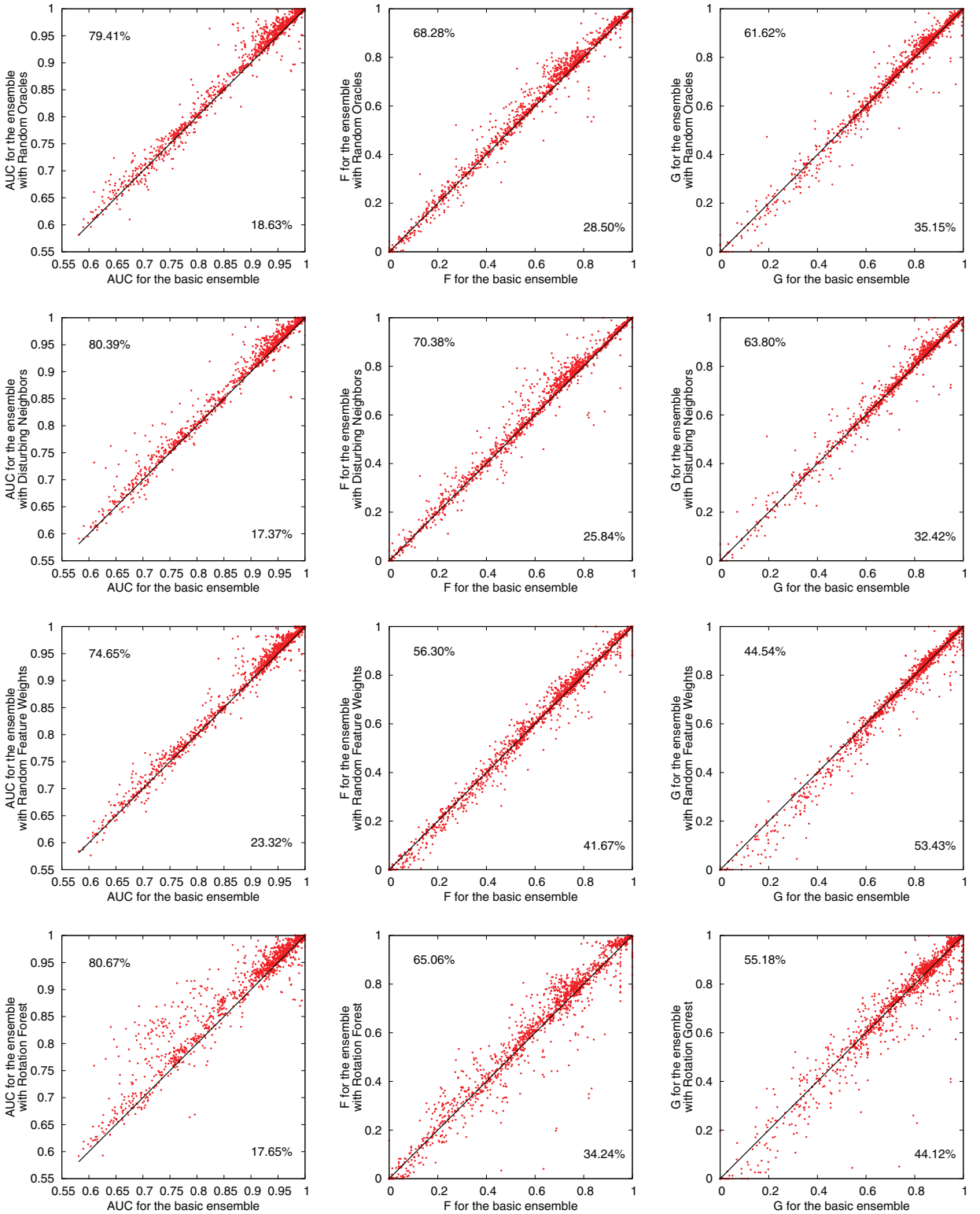

Fig. 3. Comparison of the basic ensemble methods with their combinations with Random Oracles, Disturbing Neighbors, Random Feature Weights and Rotation Forest. The numbers in the corners indicate the percentage of points above or below the diagonal.

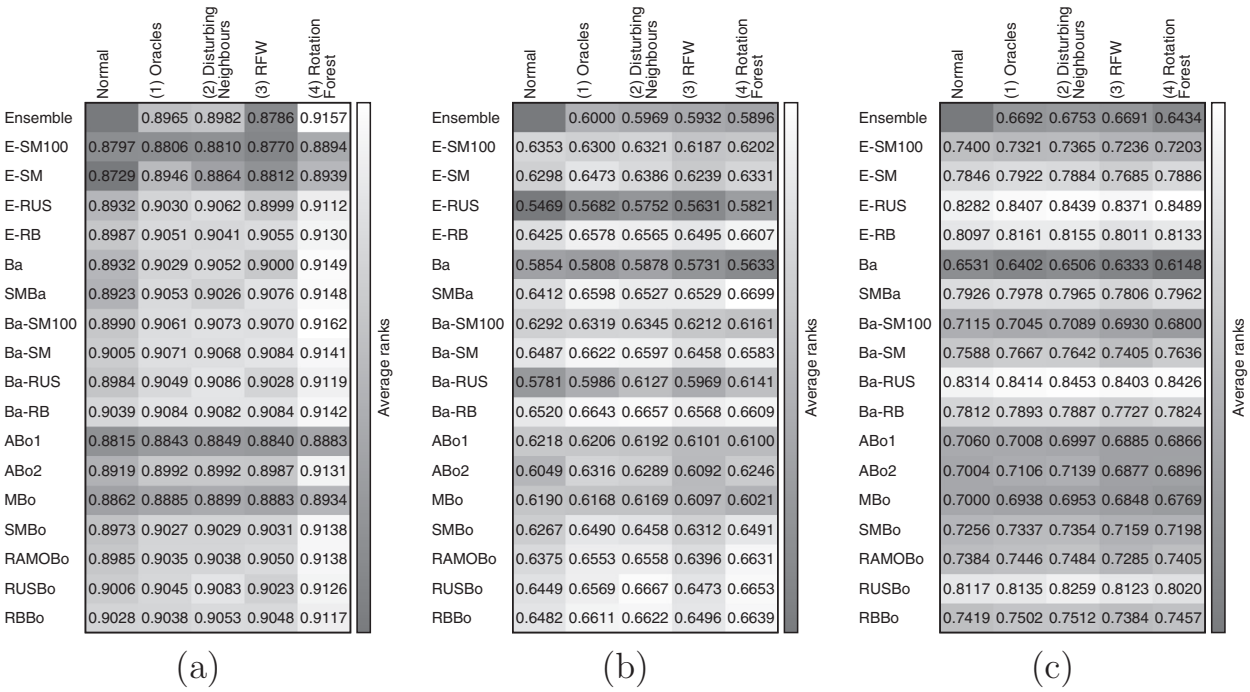

Fig. 4. Average scores in terms of the AUC (a) F-Measure (b) and G-Mean (c). The intensity of the cell reflects the score. Cells with higher values (better) are filled in light gray and those with lower values (worse) are filled in dark gray.

Table 4
Average ranks for best methods. a) According to the AUC b) according to the F-Measure c) according to the G-Mean. The combination of diversity techniques with other ensemble methods will be named using the prefix O in the case of Random Linear Oracles, DN for Disturbing neighbors and RF for Rotation Forest.

(a) AUC		(b) F-measure		(c) G-Mean	
Method	Rank	Method	Rank	Method	Rank
Rotation forest	6.774	RF+RAMOBo	6.250	O+Ba-RUS	4.607
RF+Ba-SM100	6.929	RF+RBBo	6.375	DN+RUSBo	4.750
RF+Ba	7.155	RF+SMBa	6.946	RF+E-RUS	4.768
RF+RAMOBo	7.256	RF+Ba-SM	7.375	O+E-RB	4.798
RF+ABo2	7.506	DN+Ba-RB	7.512	O+Ba-RB	5.982
RF+SMBa	7.542	RF+RUSBo	7.577	O+SMBa	6.417
RF+SMBo	7.679	RF+SMBo	7.744	O+Ba-SM	8.214
RF+E-RB	7.780	DN+Ba-SM100	9.012	DN+RBBo	8.768
RF+Ba-SM	7.940	O+E-RB	9.107	DN+RAMOBo	8.833
RF+RBBo	8.304	O+ABo2	10.679	O+E-SM	8.923
RF+Ba-RB	8.458	RF+Ba-RUS	10.762	DN+SMBo	10.560
DN+RUSBo	9.536	O+E-SM	10.857	DN+Ba-SM100	11.232
RF+Ba-RUS	10.125	DN+ABo1	11.214	DN+E-SM100	12.244
RF+E-RUS	10.280	DN+E-SM100	11.518	DN+ABo2	13.107
RF+MBo	13.482	DN+MBo	11.738	DN+ABo1	13.530
O+E-SM	13.780	Rotation forest	11.857	DN+MBo	14.399
RF+ABo1	14.780	DN+Ba	11.935	DN+Ba	14.464
RF+E-SM100	15.696	RF+E-RUS	12.542	Random oracles	15.405

3.4. Ensemble size

In the results presented until now, the ensemble size was 100. Nevertheless, different sizes can be more adequate for different ensemble methods. In [23] two ensemble sizes were considered, 10 and 40, and some ensemble methods have better results with a smaller size.

In order to study the behavior of this size, the five non-enhanced methods with best ranks in Figs. 5(b), 6(b) and 7(b) were selected for each performance measure. For the selected methods, experiments were carried out using the values 10, 20, 30,..., 100 as for the ensemble size. For each ensemble method, the ten considered sizes were compared using average ranks. Figs. 8(a–

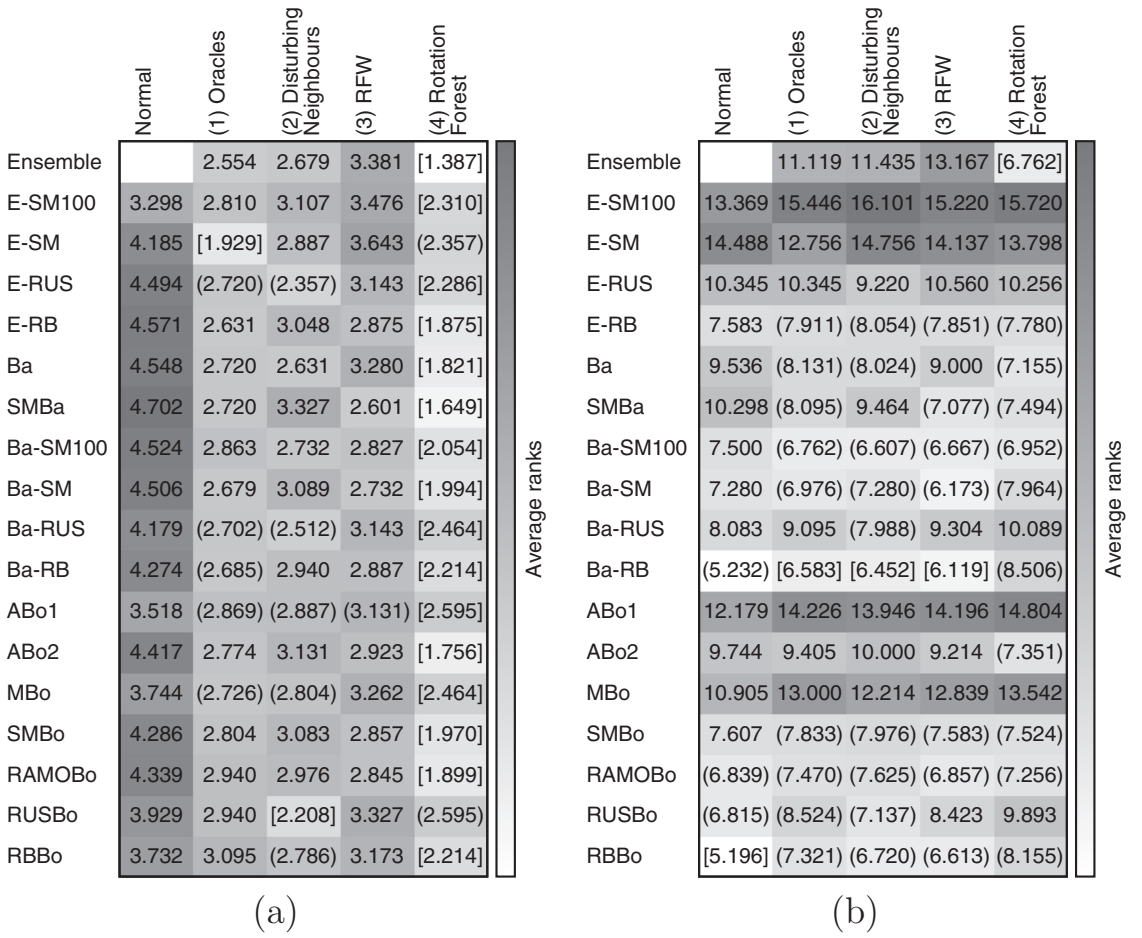

Fig. 5. Average Ranks per row (a) and column (b) (AUC). The best method in each row (a) or in each column (b) has it rank in brackets. Those methods that are equivalent to the best one at significance 0.05 are delimited in parentheses.

c) show these average ranks for the three measures. Each graph has five lines, one for each considered ensemble method. The values in the lines are in the range [1–10], as they are average ranks from 10 configurations.

In general, bigger ensemble sizes give better average ranks. Then, using 100 for the ensemble size instead of a smaller value is justified. There is one clear exception, the behavior of RUSBo for the G-mean, for this method the best sizes are 20 and 30. Interestingly, this method does not have this behavior for AUC and F-measure.

Given the unsual behavior of RUSBo with G-mean, its performance with the diversity-enhancing methods was analyzed. Fig. 8(d) shows, for the considered ensemble sizes, the average ranks of the five RUSBo ensemble configurations. As there are five configurations, the average ranks are in [1–5]. The behavior is rather uniform: for instance, for all ensemble sizes, DN+RUSBo has better rank than RUSBo and RUSBo is better than RF+RUSBo. Hence, the possibility of improving an ensemble method with a diversity-enhancing method is not restricted to a particular ensemble size.

3.5. Trying to predict when to apply diversity techniques

In the majority of the cases the diversity-enhancing technique improves the basic ensemble method, but as this improvement is not guaranteed, in this section a study is performed that attempts to relate the meta-feature of dataset with the convenience of whether to apply or not the techniques to increase diversity.

The meta-features were obtained using *the data complexity library*[7] (DCoL) [48]. This software computes the list of fourteen features shown in Table 5, which are designed to characterize the complexity of data sets for supervised learning and that were first defined in [34,35].

In order to learn the relationship between meta-features and the best combination of ensemble and diversity technique, three datasets, one per performance measure, were built with the following attributes:

[7] This software is available at http://dcol.sourceforge.net.

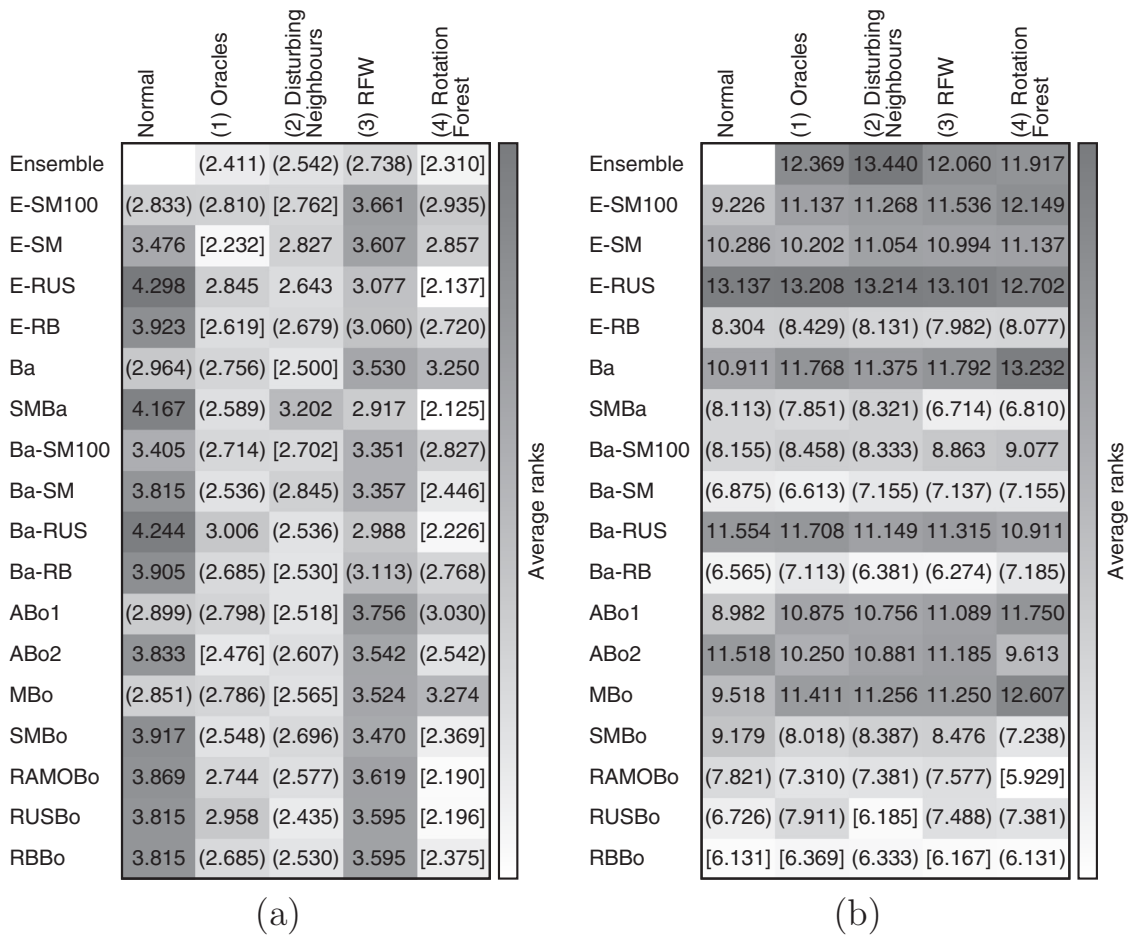

Fig. 6. Average Ranks per row (a) and column (b) (F-Measure). The best method in each row (a) or in each column (b) has it rank in brackets. Those methods that are equivalent to the best one at significance 0.05 are delimited in parentheses.

Table 5
Meta-features.

ID	Measure	ID	Measure
*F*1	Maximum Fisher's discriminant ratio	*L*3	Nonlinearity of a linear classifier
*F*1*v*	Directional-vector maximumline Fisher's discriminant ratio	*N*1	Fraction of points on the class boundary
*F*2	Overlap of the per-classline bounding boxes	*N*2	Ratio of average intra/inter class nearest neighbor distance
*F*3	Maximum (individual)line feature efficiency	*N*3	Leave-one-out error rate of the one-nearest neighbor classifier
*F*4	Collective feature efficiency	*N*4	Nonlinearity of the one-nearest neighbor classifier
*L*1	Minimized sum of the error distance of a linear classifier	*T*1	Fraction of maximum covering spheres
*L*2	Training error of a linear classifier	*T*2	Average number of points per dimension

1. The fourteen features that characterize the dataset.
2. The name of the basic ensemble.
3. The name of the technique used to increase the diversity.
4. And the class, which encodes if, for the given dataset (the one from which the first fourteen features were obtained), the combination of the ensemble and the diversity technique gives better results than the ensemble alone. The value of this attribute depends on the performance measure, and hence is usually different in each of the three datasets. If it is '*yes*', the combination of diversity technique and ensemble give better results (measured in terms of the AUC, the G-mean or the F-measure) than using the ensemble alone, on the contrary, its value is '*no*'.

First we want to know if it is possible to establish any relationship between the meta-features and the fact that the diversity technique improves the ensemble. To do this, we compared the performance of a weak classifier, that just predicts the mode of the class, with others much stronger, as J48 and Rotation Forest. Results are listed in Table 6.

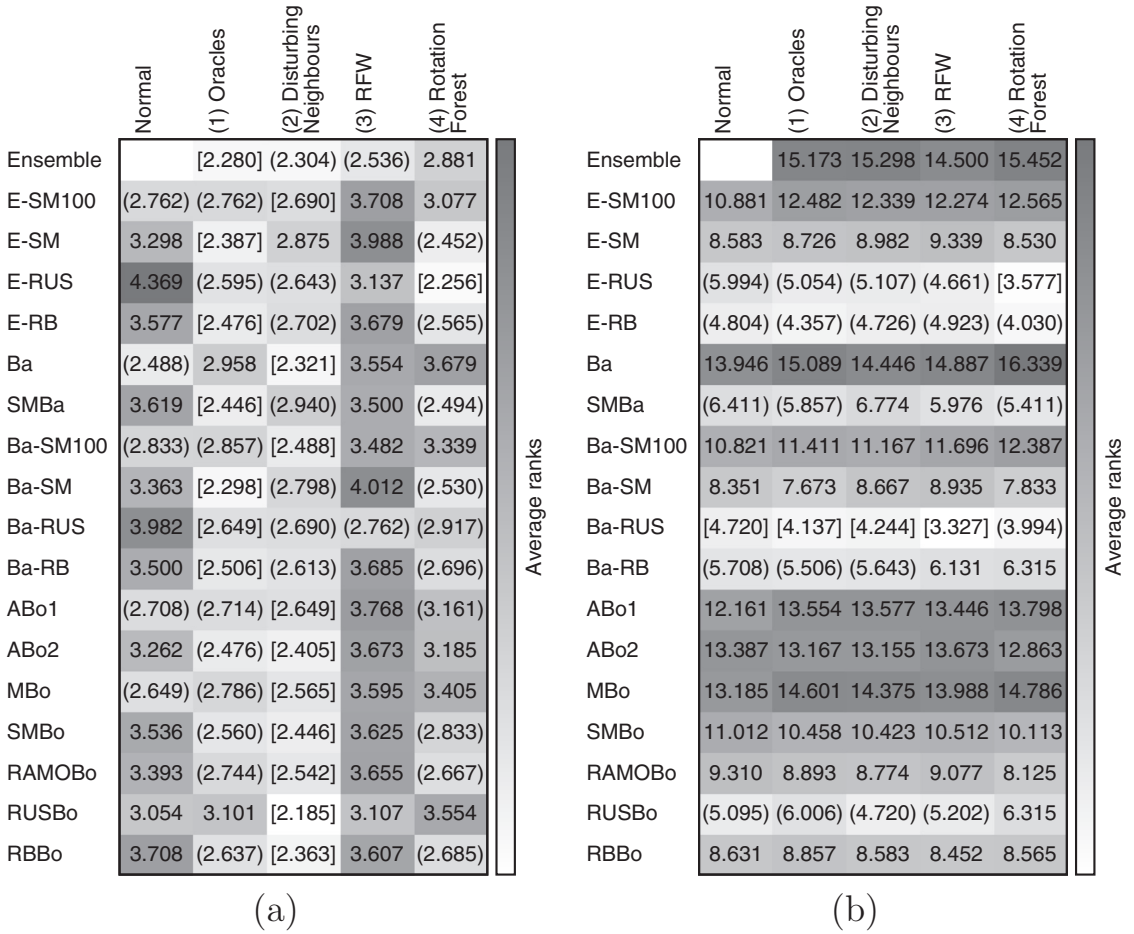

Fig. 7. Average Ranks per row (a) and column (b) (G-Mean). The best method in each row (a) or in each column (b) has it rank in brackets. Those methods that are equivalent to the best one at significance 0.05 are delimited in parentheses.

Table 6
Success percentage of the three classifiers evaluated on three meta-learning datasets (the symbol ◦ indicates the cases where there strong classifier is statistically better than the mode).

Dataset	Mode	J48	Rotation forest
Metadata for AUC	78.78	78.41	84.02 ◦
Metadata for F-measure	65.00	72.97 ◦	78.92 ◦
Metadata for G-mean	56.29	68.26 ◦	76.56 ◦

The statistical improvement obtained by using a classifier as J48 or Rotation Forest rather than simply predicting the mode is an indication that there is a relationship between the meta-features of a dataset and the fact that the combination of a diversity technique with an ensemble can give better results than the ensemble alone.

Next step is to try to learn this relationship and extract some general rules that could help us to find if a diversity technique will improve a basic ensemble method for a given dataset. We use HotSpot [30] to learn this set of rules in a tree structure, they identify for which meta-features there is a high probability that a certain diversity technique improves the ensemble. As well, this rule could help us to discard diversity techniques since they do not give any improvement to the ensemble used alone.

The following are the rules found for the dataset where the improvement is measure in terms of the AUC:

```
Class=yes (78.78% [4500/5712])
├─F2 > 0.0008 (84.09% [1601/1904])
│  ├─N1 <= 0.448 (86.65% [1532/1768])
│  └─N3 <= 0.304 (86.65% [1532/1768])
└─F1v <= 1.235 (82.46% [1514/1836])
   ├─N1 <= 0.448 (84.79% [1499/1768])
   └─N3 <= 0.304 (84.79% [1499/1768])
```

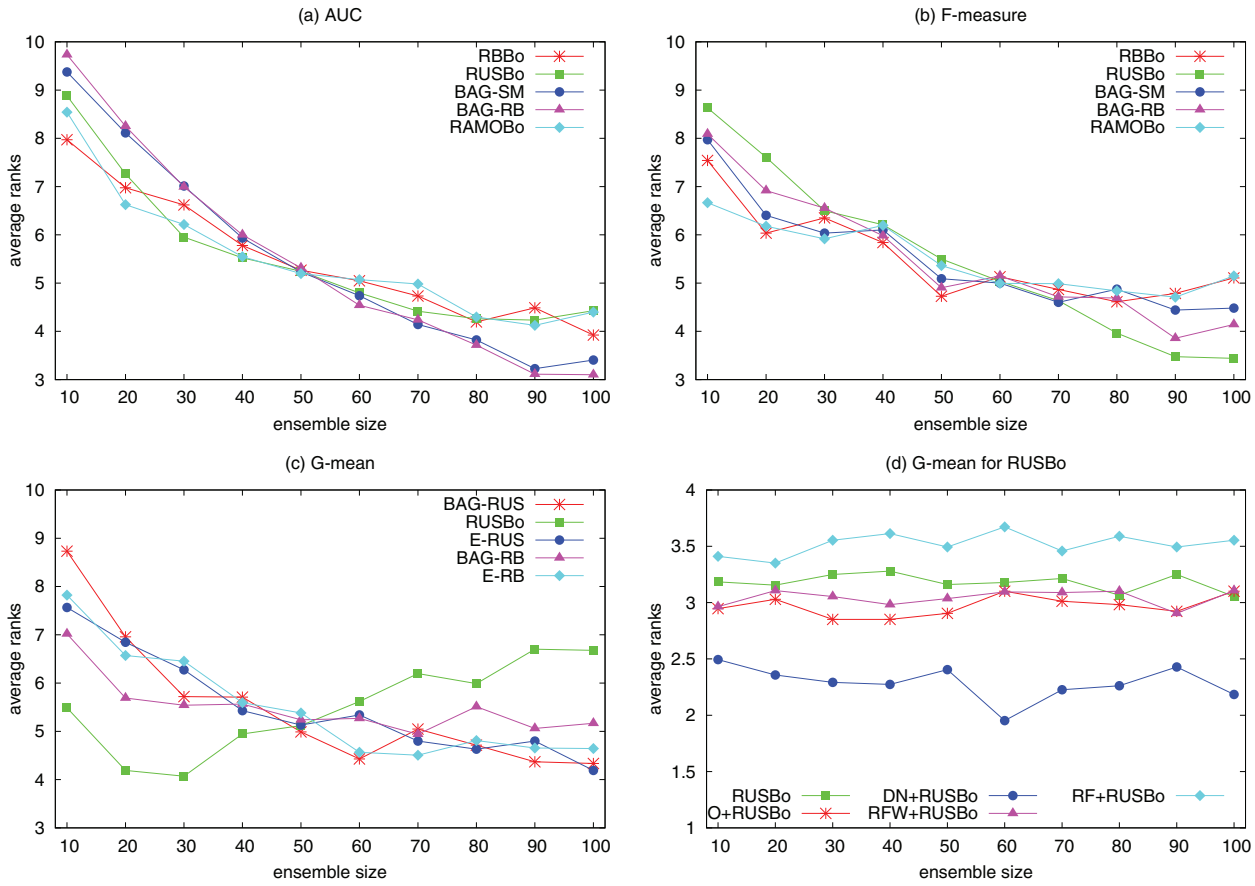

Fig. 8. Average ranks for different ensemble sizes.

The way of interpreting this rule is the following, there are 4500 instances out of 5712[8] for which the diversity technique improves the ensemble (that is in 78.78% of the instances). But the percentage is even bigger (86.65%) if we consider only those instances corresponding with datasets for which $F2$ is bigger than 0.0008 and $N1$ lower than 0.448. So the argument in favor of using diversity techniques together with ensembles is stronger for datasets with values of $F2$ and $N1$ in these ranges. With the dataset created using the F-Measure the following rules were obtained:

```
Class=yes (65% [3713/5712])
|   F1v <= 0.657 (78.98% [1289/1632])
|   |   F2 <= 0.271 (80.75% [1263/1564])
|   |   |   L1 > 0.143 (82.15% [1229/1496])
|   |   |   T2 <= 1250 (82.15% [1229/1496])
|   |   N2 <= 0.634 (80.75% [1263/1564])
|   |   |   L1 > 0.143 (82.15% [1229/1496])
|   |   |   T2 <= 1250 (82.15% [1229/1496])
|   F2 > 0.0008 (72.64% [1383/1904])
|   |   F1 > 0.185 (76.53% [1249/1632])
|   |   F2 <= 0.664 (76.23% [1244/1632])
```

That is, considering all datasets, the instances corresponding with configurations in which diversity improves the ensembles alone are 65% (improvement considering the F-measure). If we consider only datasets which $F1v$ is less than 0.657 the percentage increase to 78.98%. If the dataset has also a $F2$ value lower than 0.271, there is an extra increase to 80.75%. The percentage is 82.15 if we further restrict the dataset considering only those that have also a $L1$ greater than 0.143 or a $T2$ value lower or equal to 1250. So if we have a new dataset with values of $F1v$, $F2$, $L1$ and $T2$ verifying this inequalities, we better do not use the ensemble alone,

[8] 5712 instances: = 17 ensemble methods × 4 diversity techniques × 84 datasets.

but better combined with a diversity technique (at least is the performance measure we want to improve is the F-Measure). The rules for the dataset created using the G-Mean are:

```
Class=yes (56.29% [3215/5712])
|  F1v <= 0.657 (67.34% [1099/1632])
|  |  F2 <= 0.271 (68.41% [1070/1564])
|  |  N2 <= 0.634 (68.41% [1070/1564])
|  L1 > 0.287 (65.88% [1344/2040])
|  |  F3 <= 0.809 (69.49% [1323/1904])
|  |  |  N1 <= 0.372 (73.4% [1098/1496])
|  |  |  |  T2 > 15.385 (75.28% [1075/1428])
|  |  |  |  N4 <= 0.356 (74.44% [1063/1428])
|  |  |  N3 <= 0.258 (73.4% [1098/1496])
|  |  |  |  T2 > 15.385 (75.28% [1075/1428])
|  |  |  |  N4 <= 0.356 (74.44% [1063/1428])
|  |  N1 <= 0.365 (69.12% [1081/1564])
|  |  |  F1v <= 25.483 (71.99% [1077/1496])
|  |  |  T1 > 0.21 (71.99% [1077/1496])
```

In view of the rules obtained, if we have a dataset with low values for the directional-vector maximum Fisher's discriminant ratio, *F1v*, it would be a good idea to apply some diversity techniques.[9]

In the selected rules do not appear the basic ensemble nor the diversity-enhancing technique. This means that to determine if the basic ensemble could be improved, the meta-features are more relevant than the specific ensemble methods.

Of course all the above analysis would need further investigation, for example to find relations between the ensembles and the diversity technique more suitable when the dataset meta-features verify certain values. We include this analysis here just to give general insights about for which datasets the use of diversity techniques has a higher expectation to improve the use the ensemble alone.

3.6. The impact of noisy and borderline examples

Now we will test the suitability of a specific diversity technique, Disturbing Neighbors, for dealing with datasets that have presence of noisy and borderline examples. The repository used for this purpose comes from [47]. It is a repository that contains 30 different synthetic imbalance datasets.[10]

The artificial data sets are all two-dimensional datasets, so the increasing diversity technique more appropriate is Disturbing Neighbors, because it increases the diversity by adding new features to the dataset.

To summarize the results, we used average ranks. They are calculated using all the ensemble methods together with their enhanced version using Disturbing neighbors.

Table 7 shows the results, the first column in each subtable contains the name of the method, the second its average rank and the third value is the improvement (difference between the enhanced ensemble method and its counterpart without additional diversity).

It is clearly seen that the methods which have been combined with Disturbing Neighbors occupy the top positions of the ranking for the three measures. In the ranking calculated with the AUC and the other calculated with the F-Measure, all the methods combined with Disturbing neighbors obtains a better rank than their equivalent. This is also true for 14 of the 18 methods when the average rank is calculated with the G-Mean. The extra dimensions added by Disturbing Neighbors help in the classification task when there is presence of noisy and borderline examples.

4. Lessons learned

This paper has presented a vast experimental study in which seventeen of the state of the art methods for imbalance learning (RAMOBoost, Random Balance Boost, RUSBoost, SMOTEBoost and many more are) tested in its *basic* form and *enhanced* in combination with four different diversifying techniques: Random Oracles, Random Feature Weights, Disturbing Neighbors and Rotation Forest. Experiments were carried out using datasets from KEEL repository and the HDDT collection. Five different analysis were conducted in this paper, this section enumerates them and theirs conclussions:

1. Effect of diversity techniques in combination with ensembles in imbalanced classification. Average ranks were used for multiple method comparisons. The ranks were computed over AUC, F-Measure and Geometric Mean, comparing each ensemble method with its diversity enhanced variants. This is the summary of findings:
   - In the average rank computed using the AUC and according to Hochberg's test, all basic ensembles are significantly worse than at least one of its enhanced counterpart.

[9] A high *F1v* indicates that there is a vector that can separate well the classes once instances are projected on it, a low value indicates the opposite.

[10] It can be downloaded from http://sci2s.ugr.es/keel/imbalanced.php#sub50.

Table 7
Average ranks and Δ (increment/decrement) of rank between basic and enhanced ensembles in presence of noisy and borderline instances.

(a) AUC			(b) F-measure			(c) G-Mean		
Method	Rank	Δ	Method	Rank	Δ	Method	Rank	Δ
DN+Ba-SM100	5.333	12.767	DN+Ba-SM	8.500	4.000	DN+E-RB	4.300	4.467
DN+Ba-SM	5.600	10.033	DN+RAMOBo	8.667	3.567	DN+E-RUS	6.067	8.933
DN+SMBa	6.700	10.433	DN+SMBa	9.367	4.933	SMBa	6.433	
DN+E-RB	7.167	13.567	DN+E-SM	9.600	6.733	DN+Ba-RUS	7.433	5.567
DN+Ba-RB	8.600	10.100	DN+RUSBo	10.533	3.767	DN+SMBa	7.867	−1.433
DN+Ba	9.067	14.050	DN+Ba-RB	11.133	5.433	E-RB	8.767	
DN+RUSBo	9.533	7.967	DN+RBBo	11.967	5.033	DN+E-SM	9.133	0.333
DN+RAMOBo	12.467	4.067	RAMOBo	12.233		RUSBo	9.167	
DN+E-RUS	13.767	14.833	Ba-SM	12.500		E-SM	9.467	
DN+E-SM	14.133	11.700	DN+SMBo	12.767	8.133	Ba-SM	12.233	
DN+Ba-RUS	14.900	11.633	DN+E-RB	12.800	8.267	DN+Ba-SM	12.967	−0.733
DN+RBBo	14.933	1.933	DN+E-SM100	13.233	6.367	Ba-RUS	13	
Ba-SM	15.633		RUSBo	14.300		DN+RUSBo	13.100	−3.933
DN+E-SM100	15.633	12.233	SMBa	14.300		DN+Ba-RB	13.433	0.200
DN+MBo	16	11.400	DN+Ba-SM100	15.067	2.833	Ba-RB	13.633	
RAMOBo	16.533		E-SM	16.333		E-RUS	15	
RBBo	16.867		Ba-RB	16.567		DN+RAMOBo	15.767	2.433
SMBa	17.133		RBBo	17		E-SM100	18.067	
DN+SMBo	17.300	5.067	Ba-SM100	17.900		RAMOBo	18.200	
RUSBo	17.500		DN+ABo2	18.333	9.833	DN+E-SM100	18.400	−0.333
Ba-SM100	18.100		DN+ABo1	18.867	7.467	DN+RBBo	20.100	3.467
Ba-RB	18.700		DN+MBo	19.133	7.067	DN+SMBo	20.667	4.467
E-RB	20.733		DN+Ba-RUS	19.400	6.100	Ba-SM100	21.300	
DN+ABo2	21.500	4.817	E-SM100	19.600		DN+Ba-SM100	22.867	−1.567
SMBo	22.367		SMBo	20.900		RBBo	23.567	
Ba	23.117		E-RB	21.067		DN+ABo2	24.100	6.167
DN+ABo1	23.533	6.067	DN+E-RUS	22.067	5.633	SMBo	25.133	
E-SM	25.833		Ba-RUS	25.500		DN+ABo1	25.433	3.933
ABo2	26.317		MBo	26.200		DN+MBo	26.667	3.000
Ba-RUS	26.533		ABo1	26.333		ABo1	29.367	
MBo	27.400		DN+Ba	26.567	3.833	MBo	29.667	
E-SM100	27.867		E-RUS	27.700		ABo2	30.267	
E-RUS	28.600		ABo2	28.167		DN+Ba	31.300	0.833
ABo1	29.600		Ba	30.400		Ba	32.133	

- In the case of using the F-measure to compute the average ranks, the improvement is not statistically significant for all cases, but the improvement exists for the best methods. The best non-enhanced methods are RBBo, Ba-RB and RUSBo, and these methods are improved significantly when combined with the Rotation Forest strategy.
- Something similar happens when examining the rank calculated with the G-mean. Ba-RUS, E-RB and RUSBo are improved significantly when combined with Random Oracles and Disturbing Neighbors.

2. Determination of which is the best combination to according different metrics of performance.
   - According to the AUC, the overall winner is Rotation Forest.
   - According to the F-measure, the best method is RF+RAMOBo (RAMOBo combined with Rotation Forest).
   - According to the G-mean, the best method is O+Ba-RUS (Ba-RUS combined with Random Oracles).

   Another interesting finding is that the best combinations according to the F-measure use Oversampling strategies, while the best combinations according to the G-mean use Undersampling. The method that gets the top position in the rank according to the AUC does not use any balancing strategy.
3. Impact of ensemble size in ensemble performance. We wanted to ensure that the performance of an ensemble increases as its size increases (from 10 to 100). It is found that in general, bigger ensemble sizes give better average ranks. This trend is also seen in the enhanced ensembles. So the use of ensembles of size 100 is justified. Among the evaluated methods it was found one with a different behavior, RUSBo, whose best performance, according to the G-mean, is obtained when the size of the ensemble is in 20 to 30 range.
4. Use of complexity metrics to predict the convenience of applying diversity-enhancing techniques to improve ensemble results.
   - It was verified that it is possible to establish relationships between complexity metrics and the fact that the combination of a diversity technique with an ensemble can give better results than the ensemble alone.
   - Using the rule-learner algorithm HotSpot it was found that, for example when the overlap of the per-class bounding boxes is high or the directional-vector maximum Fisher's discriminant ratio is low (which indicates the classes are hardly separable when projected into the maximum separability vector), the application of diversity-enhancing techniques will improve the ensemble results.

5. Suitability of Disturbing Neighbors for dealing with noisy and borderline examples. Average ranks were used and it was found that methods which have been combined with Disturbing Neighbors perform better than their non-enhanced counterpart for all performance measures.

5. Concluding remarks

This article presents an exhaustive experimental study that combines techniques especially designed to work with imbalanced data with ensemble diversifying techniques. Examining 17 ensembles on their own and with four diversifying techniques, using 84 imbalanced data sets, we found that enhancing diversity pays off. Diversity-enhanced ensembles ranked better than their original counterpart. This is a curious finding because all diversity-enhancing techniques that we applied are "imbalance-blind". The method with best ranking in our experiments was the basic Rotation Forest according to AUC, and Rotation Forest combined with balancing techniques according to the F-Measure. When the G-Mean is used, there is not a technique of increasing diversity that highlights so clearly, but the techniques of increasing diversity clearly improve the ensembles to which they are applied. One interesting conclusion of this study is that the results obtained for one measure can not be extrapolated to others, and one method that is the best according to a measure, not necessarily is the best according to others.

In order to justify the ensemble size used in the experiments, it has been checked whether the ensemble size can influence the results on the improvement that can be achieved by the diversity-enhancing techniques. In general, bigger ensemble sizes give better average ranks. The RUSBo, according to the G-mean, is an exception to this general tendency, as its best results are for sizes 20 and 30. However, the general tendency is observed if AUC and F-Measure are considered. This reinforces the previous observation, the results obtained using one measure not necessarily are obtained when the others are used. Another conclusion is that it does not matter the size of the ensemble, it is always possible to improve the results by using a diversity-enhancing technique.

A preliminary study has been made that attempts to characterize for which datasets the use of diversity-enhancing techniques could be beneficial. From this study it seems that the use in ensembles of these techniques is beneficial when the overlap of the per-class bounding boxes is high or the Directional-vector maximum Fisher's discriminant ratio is low.

6. Future research directions

One interesting future line of research would be to take this study further to investigate whether it is possible to find an optimal combination of balancing and diversity techniques, or which is the balancing technique and diversity strategy best suited to a problem based on certain meta-features.

Another future research line could be to investigate further the effect of the diversity techniques on the class overlapping and small disjuncts, two of the characteristics that make imbalance problems so difficult to solve. The aim will not only be to provide insight on why these techniques work, but also could inspire the development of new diversity strategies especially designed to deal with imbalanced datasets.

Acknowledgments

This work was supported by the project TIN2011-24046 of the Spanish Ministry of Economy and Competitiveness.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.ins.2015.07.025.

References

[1] J. Alcala-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository and integration of algorithms and experimental analysis framework, J. Multiple-Valued Logic Soft Comput. 17 (2-3) (2011) 255–287.
[2] K. Bache, M. Lichman, UCI machine learning repository. 2013, http://archive.ics.uci.edu/ml.
[3] R. Barandela, R. Valdovinos, J. Sánchez, New applications of ensembles of classifiers, Pattern Anal. Appl. 6 (3) (2003) 245–256.
[4] G. Batista, R. Prati, M. Monard, A study of the behavior of several methods for balancing machine learning training data, ACM SIGKDD Explor. Newslett. 6 (1) (2004) 20–29.
[5] E. Bauer, R. Kohavi, An empirical comparison of voting classification algorithms: Bagging, boosting, and variants, Machine learn. 36 (1–2) (1999) 105–139.
[6] L. Breiman, Bagging predictors, Machine Learn. 24 (1996) 123–140.
[7] L. Breiman, Random forests, Machine learn. 45 (1) (2001) 5–32.
[8] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. 11 (1999) 131–167.
[9] C. Bunkhumpornpat, K. Sinapiromsaran, C. Lursinsap, Safe-level-SMOTE: Safe-level-synthetic minority over-sampling technique for handling the class imbalanced problem, Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD09), Vol. 5476 of Lecture Notes on Computer Science, Springer-Verlag, 2009, pp. 475–482.
[10] N. Chawla, K. Bowyer, L. Hall, W. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, J. Artif. Intell. Res. 16 (2002) 321–357.
[11] N. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, ACM SIGKDD Explor. Newslett. 6 (1) (2004) 1–6.
[12] N. Chawla, A. Lazarevic, L. Hall, K. Bowyer, SMOTEBoost: improving prediction of the minority class in boosting, in: 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD 2003), 2003, pp. 107–119.
[13] S. Chen, H. He, E.A. Garcia, Ramoboost: ranked minority oversampling in boosting, Neural Net. IEEE Transactions on 21 (10) (2010) 1624–1642.
[14] D.A. Cieslak, N.V. Chawla, Learning decision trees for unbalanced data, in: Proceedings of the 2008 European Conference on Machine Learning and Knowledge Discovery in Databases - Part I. ECML PKDD '08, Springer-Verlag, Berlin, Heidelberg, 2008, pp. 241–256.

[15] D.A. Cieslak, T.R. Hoens, N.V. Chawla, W.P. Kegelmeyer, Hellinger distance decision trees are robust and skew-insensitive, Data Min. Know. Discovery 24 (1) (Jan. 2012) 136–158.
[16] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[17] M. Di Martino, F. Decia, J. Molinelli, A. Fernández, Improving electric fraud detection using class imbalance strategies, in: ICPRAM, volume 2, 2012, pp. 135–141.
[18] T. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural comput. 10 (7) (1998) 1895–1923.
[19] J.F. Díez-Pastor, J.J. Rodríguez, C.I. García-Osorio, L.I. Kuncheva, Random balance: Ensembles of variable priors classifiers for imbalanced data, Knowledge-Based Systems 85 (2015) 96–111, doi:10.1016/j.knosys.2015.04.022.
[20] W. Fan, S.J. Stolfo, J. Zhang, P.K. Chan, AdaCost: misclassification cost-sensitive boosting, Proceedings of the Sixteenth International Conference on Machine Learning. ICML '99, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1999, pp. 97–105.
[21] T. Fawcett, An introduction to ROC analysis, Pattern recognit. lett. 27 (8) (2006) 861–874.
[22] Y. Freund, R.E. Schapire, Experiments with a new boosting algorithm, in: Machine Learning, Proceedings of the Thirteenth International Conference (ICML '96), Bari, Italy, July 3–6, 1996, pp. 148–156.
[23] M. Galar, A. Fernandez, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, Syst. Man Cybernetics Part C: appl. Rev. IEEE Transactions on 42 (4) (july 2012) 463–484.
[24] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, Eusboost: enhancing ensembles for highly imbalanced data-sets by evolutionary undersampling, Pattern Recognit. 46 (12) (2013) 3460–3471.
[25] V. García, A.I. Marqués, J.S. Sánchez, Improving risk predictions by preprocessing imbalanced credit data, Neural Information Processing, Springer, 2012, pp. 68–75.
[26] N. García-Pedrajas, C. García-Osorio, Constructing ensembles of classifiers using supervised projection methods based on misclassified instances, Expert Syst. Appl. 38 (1) (2011) 343–359.
[27] N. García-Pedrajas, J. Maudes-Raedo, C. García-Osorio, J.J. Rodríguez-Díez, Supervised subspace projections for constructing ensembles of classifiers, Inf. Sci. 193 (0) (2012) 1–21.
[28] N. García-Pedrajas, J. Pérez-Rodríguez, M.D. García-Pedrajas, D. Ortiz-Boyer, C. Fyfe, Class imbalance methods for translation initiation site recognition in DNA sequences, Knowledge-Based Sys. 25 (1) (2012) 22–34.
[29] G.-G. Geng, C.-H. Wang, Q.-D. Li, L. Xu, X.-B. Jin, Boosting the performance of web spam detection with ensemble under-sampling classification, 2007. FSKD 2007. Fourth International Conference on, Fuzzy Systems and Knowledge Discovery, Vol. 4, IEEE, 2007, pp. 583–587.
[30] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The WEKA data mining software: an update, ACM SIGKDD Explor. Newslett. 11 (1) (Nov. 2009) 10–18.
[31] H. Han, W. Wang, B. Mao, Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning, 2005 International Conference on Intelligent Computing (ICIC05), Vol. 3644 of Lecture Notes on Computer Science, Springer-Verlag, 2005, pp. 878–887.
[32] H. He, Y. Bai, E.A. Garcia, S. Li, Adasyn: adaptive synthetic sampling approach for imbalanced learning, in: IEEE International Joint Conference on. Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence), IEEE, 2008, pp. 1322–1328.
[33] T.K. Ho, The random subspace method for constructing decision forests, IEEE Transactions on Pattern Anal. Mach. Intell. 20 (8) (Aug 1998) 832–844.
[34] T.K. Ho, M. Basu, Complexity measures of supervised classification problems, IEEE Transactions on Pattern Anal. Mach. Intell. 24 (3) (2002) 289–300.
[35] T.K. Ho, M. Basu, M.H.C. Law, Measures of geometrical complexity in classification problems, in: Data complexity in pattern recognition, Springer, 2006, pp. 1–23.
[36] Y. Hochberg, A sharper Bonferroni procedure for multiple tests of significance, Biometrika 75 (1988) 800–803.
[37] R. Iman, J. Davenport, Approximations of the critical region of the fbietkan statistic, Commun. Stat. Theory Method. 9 (6) (1980) 571–595.
[38] T. Jo, N. Japkowicz, Class imbalances versus small disjuncts, ACM SIGKDD Explor. Newslett. 6 (1) (2004) 40–49.
[39] M. Kubat, Matwin, Addressing the Curse of imbalanced training sets: one-Sided selection, in: Proceedings of the 14th International Conference on Machine Learning, 1997, pp. 179–186.
[40] L. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.
[41] L.I. Kuncheva, J.J. Rodriguez, Classifier ensembles with a random linear oracle, IEEE Transactions on Knowledge Data Eng. 19 (4) (2007) 500–508.
[42] W. Liu, S. Chawla, D.A. Cieslak, N.V. Chawla, A robust decision tree algorithm for imbalanced data sets, in: Proceedings of the SIAM International Conference on Data Mining, SDM 2010, 2010, pp. 766–777.
[43] X.-Y. Liu, J. Wu, Z.-H. Zhou, Exploratory undersampling for class-imbalance learning. systems, man, and cybernetics, IEEE Transactions on Part B: Cybernetics 39 (2) (2009) 539–550.
[44] V. López, A. Fernández, S. García, V. Palade, F. Herrera, An insight into classification with imbalanced data: empirical results and current trends on using data intrinsic characteristics, Inf. Sci. 250 (2013) 113–141.
[45] J. Maudes, J.J. Rodríguez, C. García-Osorio, Disturbing neighbors diversity for decision forests, in: O. Okun, G. Valentini (Eds.), Applications of Supervised and Unsupervised Ensemble Methods, Vol. 245 of Studies in Computational Intelligence, Springer Berlin Heidelberg, 2009, pp. 113–133.
[46] J. Maudes, J.J. Rodríguez, C. García-Osorio, N. García-Pedrajas, Random feature weights for decision tree ensemble construction, Inf. Fusion 13 (1) (2012) 20–30.
[47] K. Napierała, J. Stefanowski, S. Wilk, Learning from imbalanced data in presence of noisy and borderline examples, in: Rough Sets and Current Trends in Computing, Springer, 2010, pp. 158–167.
[48] A. Orriols-Puig, N. Macià, T.K. Ho, Documentation for the data complexity library in C++, Tech. rep., La Salle - Universitat Ramon Llull, 2010.
[49] R. Polikar, Ensemble learning, in: Ensemble Machine Learning, Springer, 2012, pp. 1–34.
[50] R.C. Prati, G.E. Batista, D.F. Silva, Class imbalance revisited: a new experimental setup to assess the performance of treatment methods, in: Knowledge and Information Systems, 2014, pp. 1–24.
[51] F. Provost, P. Domingos, Tree induction for probability-based ranking, Machine Learn. 52 (3) (2003) 199–215.
[52] F. Provost, V. Kolluri, A survey of methods for scaling up inductive algorithms, Data min. knowledge discovery 3 (2) (1999) 131–169.
[53] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kauffman, 1993.
[54] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[55] J.J. Rodriguez, 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.
[56] A. Schclar, L. Rokach, Random projection ensemble classifiers, in: J. Filipe, J. Cordeiro (Eds.), Enterprise Information Systems, Vol. 24 of Lecture Notes in Business Information Processing, Springer Berlin Heidelberg, 2009, pp. 309–316.
[57] C. Seiffert, T. Khoshgoftaar, J. Van Hulse, A. Napolitano, RUSBoost: A hybrid approach to alleviating class imbalance, IEEE Transactions on Syst. Man Cybernetics Part A: Syst. Humans 40 (1) (2010) 185–197.
[58] V. Sofia, Issues in mining imbalanced data sets — a review paper, in: Proceedings of the Sixteen Midwest Artificial Intelligence and Cognitive Science Conference, 2005, pp. 67–73.
[59] J. Stefanowski, Overlapping, rare examples and class decomposition in learning classifiers from imbalanced data, in: Emerging Paradigms in Machine Learning, Springer, 2013, pp. 277–306.
[60] Y. Sun, M. Kamel, A. Wong, Y. Wang, Cost-sensitive boosting for classification of imbalanced data, in: Pattern Recognition, volume 40, 2007, pp. 3358–3378.
[61] I. Tomek, Two modifications of cnn. systems, man and cybernetics, transactions on 6, 1976, 769–772.
[62] C. Van Rijsbergen, Information Retrieval, 1979, Butterworths.
[63] F. Verhein, S. Chawla, Using significant, positively associated and relatively class correlated rules for associative classification of imbalanced datasets, in: Seventh IEEE International Conference on Data Mining, 2007. ICDM 2007, Oct 2007, pp. 679–684.

[64] S. Wang, X. Yao, Diversity analysis on imbalanced data sets by using ensemble models, in: IEEE Symposium Series on Computational Intelligence and Data Mining (IEEE CIDM 2009), 2009, pp. 324–331.
[65] S. Wang, X. Yao, Using class imbalance learning for software defect prediction, IEEE Transactions on Reliability 62 (2) (June 2013) 434–443.
[66] M. Wasikowski, X. wen Chen, Combating the small sample class imbalance problem using feature selection, Knowledge Data Eng. IEEE Transactions on 22 (10) (Oct 2010) 1388–1400.
[67] G.I. Webb, Multiboosting: A technique for combining boosting and wagging, Machine Learn. 40 (2) (2000) 159–196.
[68] G.M. Weiss, The impact of small disjuncts on classifier learning, in: Data Mining, Springer, 2010, pp. 193–226.
[69] D. Wilson, Asymptotic properties of nearest neighbor rules using edited data, IEEE Transactions on Syst. Man Cybernetics 2 (3) (1972) 408–421.
[70] H. Yu, J. Ni, Y. Dan, S. Xu, Mining and integrating reliable decision rules for imbalanced cancer gene expression data sets, Tsinghua Sci. Tech. 17 (6) (2012) 666–673.
[71] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, 2001, pp. 204–213.