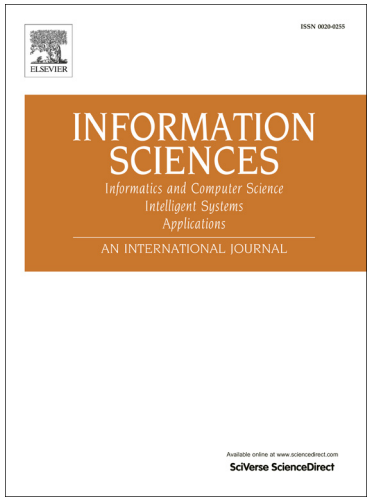
Accepted Manuscript



An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics

Victoria López, Alberto Fernández, Salvador García, Vasile Palade, Francisco Herrera

PII: S0020-0255(13)00512-4
DOI: http://dx.doi.org/10.1016/j.ins.2013.07.007
Reference: INS 10192

To appear in: *Information Sciences*

Received Date: 2 October 2012
Revised Date: 16 April 2013
Accepted Date: 5 July 2013

Please cite this article as: 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, *Information Sciences* (2013), doi: http://dx.doi.org/10.1016/j.ins.2013.07.007

This is a PDF file of an unedited manuscript that has been accepted for publication. As a service to our customers we are providing this early version of the manuscript. The manuscript will undergo copyediting, typesetting, and review of the resulting proof before it is published in its final form. Please note that during the production process errors may be discovered which could affect the content, and all legal disclaimers that apply to the journal pertain.

ACCEPTED MANUSCRIPT

An Insight into Classification with Imbalanced Data: Empirical Results and Current Trends on Using Data Intrinsic Characteristics

Victoria López[a,*], Alberto Fernández[b], Salvador García[b], Vasile Palade[c], Francisco Herrera[a]

[a] *Dept. of Computer Science and Artificial Intelligence, CITIC-UGR (Research Center on Information and Communications Technology), University of Granada, Granada, Spain*
[b] *Dept. of Computer Science, University of Jaén, Jaén, Spain*
[c] *Department of Computer Science, University of Oxford, OX1 3QD, United Kingdom*

Abstract

Training classifiers with datasets which suffer of imbalanced class distributions is an important problem in data mining. This issue occurs when the number of examples representing the class of interest is much lower than the ones of the other classes. Its presence in many real-world applications has brought along a growth of attention from researchers.

We shortly review the many issues in machine learning and applications of this problem, by introducing the characteristics of the imbalanced dataset scenario in classification, presenting the specific metrics for evaluating performance in class imbalanced learning and enumerating the proposed solutions. In particular, we will describe preprocessing, cost-sensitive learning and ensemble techniques, carrying out an experimental study to contrast these approaches in an intra and inter-family comparison.

We will carry out a thorough discussion on the main issues related to using data intrinsic characteristics in this classification problem. This will help to improve the current models with respect to: the presence of small disjuncts, the lack of density in the training data, the overlapping between classes, the identification of noisy data, the significance of the borderline instances, and the dataset shift between the training and the test distributions. Finally, we introduce several approaches and recommendations to address these problems in conjunction with imbalanced data, and we will show some experimental examples on the behavior of the learning algorithms on data with such intrinsic characteristics.

Keywords:

*Corresponding author. Tel:+34-953-213016; Fax: +34-953-212472
Email addresses: vlopez@decsai.ugr.es (Victoria López), alberto.fernandez@ujaen.es (Alberto Fernández), sglopez@ujaen.es (Salvador García), vasile.palade@cs.ox.ac.uk (Vasile Palade), herrera@decsai.ugr.es (Francisco Herrera)

ACCEPTED MANUSCRIPT

Imbalanced Datasets, Sampling, Cost-Sensitive Learning, Small Disjuncts, Noisy Data, Borderline Examples, Dataset Shift

1. Introduction

In many supervised learning applications, there is a significant difference between the prior probabilities of different classes, i.e, between the probabilities with which an example belongs to the different classes of the classification problem. This situation is known as the class imbalance problem [29, 66, 118] and it is common in many real problems from telecommunications, web, finance-world, ecology, biology, medicine not only, and which can be considered one of the top problems in data mining today [143]. Furthermore, it is worth to point out that the minority class is usually the one that has the highest interest from a learning point of view and it also implies a great cost when it is not well classified [42].

The hitch with imbalanced datasets is that standard classification learning algorithms are often biased towards the majority class (known as the "negative" class) and therefore there is a higher misclassification rate for the minority class instances (called the "positive" examples). Therefore, throughout the last years, many solutions have been proposed to deal with this problem, both for standard learning algorithms and for ensemble techniques [50]. They can be categorized into three major groups:

1. **Data sampling**: In which the training instances are modified in such a way to produce a more or less balanced class distribution that allow classifiers to perform in a similar manner to standard classification [9, 27].
2. **Algorithmic modification**: This procedure is oriented towards the adaptation of base learning methods to be more attuned to class imbalance issues [147]
3. **Cost-sensitive learning**: This type of solutions incorporate approaches at the data level, at the algorithmic level, or at both levels combined, considering higher costs for the misclassification of examples of the positive class with respect to the negative class, and therefore, trying to minimize higher cost errors [38, 148].

In this paper, our first goal is to come up with a review on this type of methodologies, presenting a taxonomy for each group, enumerating and briefly describing the main properties of the most significant approaches that have been traditionally applied in this field. Furthermore, we carry out an experimental study in order to highlight the behavior of the different paradigms that were previously presented.

Most of the studies on the behavior of several standard classifiers in imbalance domains have shown that significant loss of performance is mainly due to the skewed class distribution, given by the imbalance ratio (IR), defined as the ratio of the number of instances in the majority class to the number of examples in the minority class [58, 98]. However, there are several investigations which also suggest that there are other factors that contribute to such performance

ACCEPTED MANUSCRIPT

degradation [72]. Therefore, as a second goal, we present a discussion about six significant problems related to data intrinsic characteristics and that must be taken into account in order to provide better solutions for correctly identifying both classes of the problem:

1. The identification of areas with small disjuncts [136, 137].
2. The lack of density and information in the training data [133].
3. The problem of overlapping between the classes [37, 55].
4. The impact of noisy data in imbalanced domains [20, 111].
5. The significance of the borderline instances to carry out a good discrimination between the positive and negative classes, and its relationship with noisy examples [39, 97].
6. The possible differences in the data distribution for the training and test data, also known as the dataset shift [95, 114].

This thorough study of the problem can guide us about the source where the difficulties for imbalanced classification emerge, focusing on the analysis of significant data intrinsic characteristics. Specifically, for each established scenario we show an experimental example on how it affects the behavior of the learning algorithms, in order to stress its significance.

We must point out that some of these topics have recent studies associated, which are described along this paper, examining their main contributions and recommendations. However, we emphasize that they still need to be addressed in more detail in order to have models of high quality in this classification scenario and, therefore, we have stressed them as future trends of research for imbalanced learning. Overcoming these problems can be the key for developing new approaches that improve the correct identification of both the minority and majority classes.

In summary, the main contributions of this new review on former works on this topic [66, 118] can be highlighted with respect to two points: (1) the extensive experimental study with a large benchmark of 66 imbalanced datasets for analysing the behavior of the solutions proposed to address the problem of imbalanced data; and (2) a detailed analysis and study of the data intrinsic characteristics in this scenario and a brief description on how they affect the performance of the classification algorithms.

With this aim in mind, this paper is organized as follows. First, Section 2 presents the problem of imbalanced datasets, introducing its features and the metrics employed in this context. Section 3 describes the diverse preprocessing, cost-sensitive learning and ensemble methodologies that have been proposed to deal with this problem. Next, we develop an experimental study for contrasting the behavior of these approaches in Section 4. Section 5 is devoted to analyzing and discussing the aforementioned problems associated with data intrinsic characteristics. Finally, Section 6 summarizes and concludes the work.

ACCEPTED MANUSCRIPT

2. Imbalanced Datasets in Classification

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

2.1. The problem of imbalanced datasets

In the classification problem field, the scenario of imbalanced datasets appears frequently. The main property of this type of classification problem is that the examples of one class significantly outnumber the examples of the other one [66, 118]. The minority class usually represents the most important concept to be learned, and it is difficult to identify it since it might be associated with exceptional and significant cases [135], or because the data acquisition of these examples is costly [139]. In most cases, the imbalanced class problem is associated to binary classification, but the multi-class problem often occurs and, since there can be several minority classes, it is more difficult to solve [48, 81].

Since most of the standard learning algorithms consider a balanced training set, this may generate suboptimal classification models, i.e. a good coverage of the majority examples, whereas the minority ones are misclassified frequently. Therefore, those algorithms, which obtain a good behavior in the framework of standard classification, do not necessarily achieve the best performance for imbalanced datasets [47]. There are several reasons behind this behavior:

1. The use of global performance measures for guiding the learning process, such as the standard accuracy rate, may provide an advantage to the majority class.
2. Classification rules that predict the positive class are often highly specialized and thus their coverage is very low, hence they are discarded in favor of more general rules, i.e. those that predict the negative class.
3. Very small clusters of minority class examples can be identified as noise, and therefore they could be wrongly discarded by the classifier. On the contrary, few real noisy examples can degrade the identification of the minority class, since it has fewer examples to train with.

In recent years, the imbalanced learning problem has received much attention from the machine learning community. Regarding real world domains, the importance of the imbalance learning problem is growing, since it is a recurring issue in many applications. As some examples, we could mention very high resolution airbourne imagery [31], forecasting of ozone levels [125], face recognition [78], and especially medical diagnosis [11, 86, 91, 93, 132]. It is important to remember that the minority class usually represents the concept of interest and it is the most difficult to obtain from real data, for example patients with illnesses in a medical diagnosis problem; whereas the other class represents the counterpart of that concept (healthy patients).

ACCEPTED MANUSCRIPT

2.2. Evaluation in imbalanced domains

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

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

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

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

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

In imbalanced domains, the evaluation of the classifiers' performance must be carried out using specific metrics in order to take into account the class distribution. Concretely, we can obtain four metrics from Table 1 to measure the classification performance of both, positive and negative, classes independently:

- **True positive rate**: $TP_{rate} = \frac{TP}{TP+FN}$ is the percentage of positive instances correctly classified.
- **True negative rate**: $TN_{rate} = \frac{TN}{FP+TN}$ is the percentage of negative instances correctly classified.
- **False positive rate**: $FP_{rate} = \frac{FP}{FP+TN}$ is the percentage of negative instances misclassified.
- **False negative rate**: $FN_{rate} = \frac{FN}{TP+FN}$ is the percentage of positive instances misclassified.

Since in this classification scenario we intend to achieve good quality results for both classes, there is a necessity of combining the individual measures of both the positive and negative classes, as none of these measures alone is adequate by itself.

A well-known approach to unify these measures and to produce an evaluation criteria is to use the Receiver Operating Characteristic (ROC) graphic [19]. This graphic allows the visualization of the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}), as it evidences that any classifier cannot increase the number

ACCEPTED MANUSCRIPT

of true positives without also increasing the false positives. The Area Under the ROC Curve (AUC) [70] corresponds to the probability of correctly identifying which one of the two stimuli is noise and which one is signal plus noise. The AUC provides a single measure of a classifier's performance for evaluating which model is better on average. Fig. 1 shows how to build the ROC space plotting on a two-dimensional chart the TP_{rate} (Y-axis) against the FP_{rate} (X-axis). Points in $(0, 0)$ and $(1, 1)$ are trivial classifiers where the predicted class is always the negative and positive one, respectively. On the contrary, $(0, 1)$ point represents the perfect classifier. TheAUC measure is computed just by obtaining the area of the graphic:

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

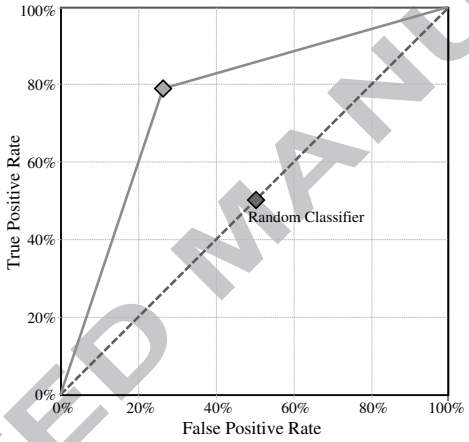


Figure 1: Example of a ROC plot. Two classifiers' curves are depicted: the dashed line represents a random classifier, whereas the solid line is a classifier which is better than the random classifier.

In [103], the significance of these graphical methods for the classification predictive performance evaluation is stressed. According to the authors, the main advantage of this type of methods resides in their ability to depict the trade-offs between evaluation aspects in a multidimensional space rather than reducing these aspects to an arbitrarily chosen (and often biased) single scalar measure. In particular, they present a review of several representation mechanisms emphasizing the best scenario for their use; for example, in imbalanced domains, when we are interested in the positive class, it is recommended the use of precision-recall graphs [36]. Furthermore, the expected cost or profit of each model might be analyzed using cost curves [40], lift and ROI graphs [83].

Other metric of interest to be stressed in this area is the geometric mean of the true rates [7], which can be defined as:

$$GM = \sqrt{\frac{TP}{TP + FN} \cdot \frac{TN}{FP + TN}} \tag{3}$$

ACCEPTED MANUSCRIPT

This metric attempts to maximize the accuracy on each of the two classes with a good balance, being a performance metric that correlates both objectives. However, due to this symmetric nature of the distribution of the geometric mean over TP_{rate} (sensitivity) and the TN_{rate} (specificity), it is hard to contrast different models according to their precision on each class.

Another significant performance metric that is commonly used is the F-measure [6]:

$$F_m = \frac{(1+\beta^2)(PPV \cdot TP_{rate})}{\beta^2 PPV + TP_{rate}}$$
$$PPV = \frac{TP}{TP+FP} \tag{4}$$

A popular choice for β is 1, where equal importance is assigned for both TP_{rate} and the positive predictive value (PPV). This measure would be more sensitive to the changes in the PPV than to the changes in TP_{rate}, which can lead to the selection of sub-optimal models.

According to the previous comments, some authors try to propose several measures for imbalanced domains in order to be able to obtain as much information as possible about the contribution of each class to the final performance and to take into account the IR of the dataset as an indication of its difficulty. For example, in [10, 14] the *Adjusted G-mean* is proposed. This measure is designed towards obtaining the highest sensitivity (TP_{rate}) without decreasing too much the specificity (TN_{rate}). This fact is measured with respect to the original model, i.e. the original classifier without addressing the class imbalance problem. Equation 5 shows its definition:

$$\begin{aligned} AGM &= \frac{GM + TN_{rate} \cdot (FP+TN)}{1+FP+TN}; && \text{If } TP_{rate} > 0, \\ AGM &= 0; && \text{If } TP_{rate} = 0 \end{aligned} \tag{5}$$

Additionally, in [54] the authors presented a simple performance metric, called *Dominance*, which is aimed to point out the dominance or prevalence relationship between the positive class and the negative class, in the range $[-1, +1]$ (Equation 6). Furthermore, it can be used as a visual tool to analyze the behavior of a classifier on a 2-D space from the joint perspective of global precision (Y-axis) and dominance (X-axis).

$$Dom = TP_{rate} - TN_{rate} \tag{6}$$

The same authors, using the previous concept of *dominance*, define a new metric called *Index of Balanced Accuracy* (IBA) [56, 57]. IBA weights a performance measure, that aims to make it more sensitive for imbalanced domains. The weighting factor favors those results with moderately better classification rates on the minority class. IBA is formulated as follows:

$$IBA_\alpha(M) = (1 + \alpha \cdot Dom)M \tag{7}$$

ACCEPTED MANUSCRIPT

where $(1+\alpha \cdot Dom)$ is the weighting factor and M represents a performance metric. The objective is to moderately favor the classification models with higher prediction rate on the minority class (without underestimating the relevance of the majority class) by means of a weighted function of any plain performance evaluation measure.

A comparison regarding these evaluation proposals for imbalanced datasets is out of the scope of this paper. For this reason, we refer any interested reader to find a deep experimental study in [57] and [105].

3. Addressing Classification with Imbalanced Data: Preprocessing, Cost-Sensitive Learning and Ensemble Techniques

A large number of approaches have been proposed to deal with the class imbalance problem. These approaches can be categorized into two groups: the internal approaches that create new algorithms or modify existing ones to take the class-imbalance problem into consideration [7, 41, 82, 129, 152] and external approaches that preprocess the data in order to diminish the effect of their class imbalance [9, 43]. Furthermore, cost-sensitive learning solutions incorporating both the data (external) and algorithmic level (internal) approaches assume higher misclassification costs for samples in the minority class and seek to minimize the high cost errors [15, 38, 59, 117, 150]. Ensemble methods [101, 108] are also frequently adapted to imbalanced domains, either by modifying the ensemble learning algorithm at the data-level approach to preprocess the data before the learning stage of each classifier [17, 30, 112] or by embedding a cost-sensitive framework in the ensemble learning process [44, 117, 122].

Regarding this, in this section we first introduce the main aspects of the preprocessing techniques. Next, we describe the cost-sensitive learning approach. Finally, we present some relevant ensemble techniques in the framework of imbalanced datasets.

3.1. Preprocessing imbalanced datasets: resampling techniques

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

Those works have proved empirically that applying a preprocessing step in order to balance the class distribution is usually an useful solution [9, 12, 45, 46]. Furthermore, the main advantage of these techniques is that they are independent of the underlying classifier.

Resampling techniques can be categorized into three groups or families:

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

ACCEPTED MANUSCRIPT

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

In order to deal with the mentioned problems, more sophisticated methods have been proposed. Among them, the "Synthetic Minority Oversampling TEchnique" (SMOTE) [27] has become one of the most renowned approaches in this area. In brief, its main idea is to create new minority class examples by interpolating several minority class instances that lie together for oversampling the training set.

With this technique, the positive class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbors. Depending upon the amount of over-sampling required, neighbors from the k nearest neighbors are randomly chosen. This process is illustrated in Figure 2, where x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbors and r_1 to r_4 the synthetic data points created by the randomized interpolation.

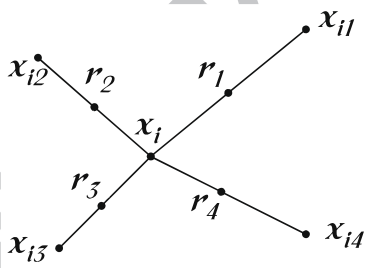


Figure 2: An illustration of how to create the synthetic data points in the SMOTE algorithm

However, in oversampling techniques, and especially for the SMOTE algorithm, the problem of over generalization is largely attributed to the way in which synthetic samples are created. Precisely, SMOTE generates the same number of synthetic data samples for each original minority example and does so without consideration to neighboring examples, which increases the occurrence of overlapping between classes [128]. To this end, various adaptive sampling methods have been proposed to overcome this limitation; some representative works include the Borderline-SMOTE [63], Adaptive Synthetic Sampling [65], Safe-Level-SMOTE [21] and SPIDER2 [116] algorithms.

Regarding undersampling, most of the proposed approaches are based on data cleaning techniques. Some representative works in this area include the Wilson's edited nearest neighbor (ENN) [140] rule, which removes examples that differ from two of its three nearest neighbors, the one-sided selection (OSS) [76], an integration method between the condensed nearest neighbor rule [64] and Tomek Links [124] and the neighborhood cleaning rule [79], which is based

ACCEPTED MANUSCRIPT

on the ENN technique. Additionally, the NearMiss-2 method [149] selects the majority class examples whose average distance to the three farthest minority class examples is the smallest, and in [5] the authors proposed a method that removes the majority instances far from the decision boundaries. Furthermore, a Support Vector Machine (SVM) [35] may be used to discard redundant or irrelevant majority class examples [119]. Finally, the combination of preprocessing of instances with data cleaning techniques could lead to diminishing the overlapping that is introduced by sampling methods, i.e. the integrations of SMOTE with ENN and SMOTE with Tomek links [9]. This behavior is also present in a wrapper technique introduced in [28] that defines the best percentage to perform both undersampling and oversampling.

On the other hand, these techniques are not only carried out by means of a "neighborhood', but we must also stress some cluster-based sampling algorithms, all of which aim to organize the training data into groups with significant characteristics and then performing both undersampling and/or oversampling. Some significant examples are the Cluster-Based Oversampling (CBO) [73], Class Purity Maximization [146], Sampling-Based Clustering [145],the agglomerative Hierarchical Clustering [34] or the DBSMOTE algorithm based on DBSCAN clustering [22].

Finally, the application of genetic algorithms or particle swarm optimization for the correct identification of the most useful instances has shown to achieve good results [53, 142]. Also, a training set selection can be carried out in the area of imbalanced datasets [51, 52]. These methods select the best set of examples to improve the behavior of several algorithms considering for this purpose the classification performance using an appropriate imbalanced measure.

3.2. Cost-sensitive learning

Cost-sensitive learning takes into account the variable cost of a misclassification with respect to the different classes [38, 148]. In this case, a cost matrix codifies the penalties $C(i, j)$ of classifying examples of one class i as a different one j, as illustrated in Table 2.

Table 2: Example of a cost matrix for a fraud detection classification problem

	fraudulent	**legitimate**
refuse	20$	-20$
approve	-100$	50$

These misclassification cost values can be given by domain experts, or can be learned via other approaches [117, 118]. Specifically, when dealing with imbalanced problems, it is usually more interesting to recognize the positive instances rather than the negative ones. Therefore, the cost when misclassifying a positive instance must be higher than the cost of misclassifying a negative one, i.e. $C(+,-) > C(-,+)$.

ACCEPTED MANUSCRIPT

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

$$R(i|x) = \sum_j P(j|x) \cdot C(i,j) \tag{8}$$

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

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

or, which is equivalent:

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

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

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

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

$$p^* = \frac{C(1,0)}{C(1,0) - C(0,1)} = \frac{FP}{FP + FN} \tag{9}$$

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

$$p(1)FN : p(0)FP \tag{10}$$

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

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

1. **Direct methods**: The main idea of building a direct cost-sensitive learning algorithm is to directly introduce and utilize misclassification costs into the learning algorithms.
   For example, in the context of decision tree induction, the tree-building strategies are adapted to minimize the misclassification costs. The cost information is used to: (1) choose the best attribute to split the data [84, 107]; and (2) determine whether a subtree should be pruned [18]. On the other hand, other approaches based on genetic algorithms can incorporate misclassification costs in the fitness function [126].

ACCEPTED MANUSCRIPT

2. **Meta-learning**: This methodology implies the integration of a "preprocessing" mechanism for the training data or a "postprocessing" of the output, in such a way that the original learning algorithm is not modified. Cost-sensitive meta-learning can be further classified into two main categories: *thresholding* and *sampling*, which are based on expressions (9) and (10) respectively:

   - **Thresholding** is based on the basic decision theory that assigns instances to the class with minimum expected cost. For example, a typical decision tree for a binary classification problem assigns the class label of a leaf node depending on the majority class of the training samples that reach the node. A cost-sensitive algorithm assigns the class label to the node that minimizes the classification cost [38, 147].
   - **Sampling** is based on modifying the training dataset. The most popular technique lies in resampling the original class distribution of the training dataset according to the cost decision matrix by means of undersampling/oversampling [148] or assigning instance weights [123]. These modifications have shown to be effective and can also be applied to any cost insensitive learning algorithm [150].

3.3. Ensemble methods

Ensemble-based classifiers, also known as multiple classifier systems [101], try to improve the performance of single classifiers by inducing several classifiers and combining them to obtain a new classifier that outperforms every one of them. Hence, the basic idea is to construct several classifiers from the original data and then aggregate their predictions when unknown instances are presented.

In recent years, ensembles of classifiers have arisen as a possible solution to the class imbalance problem [77, 85, 112, 117, 127, 131]. Ensemble-based methods are based on a combination between ensemble learning algorithms and one of the previously discussed techniques, namely data and algorithmic approaches, or cost-sensitive learning solutions. In the case of adding a data level approach to the ensemble learning algorithm, the new hybrid method usually preprocess the data before training each classifier. On the other hand, cost-sensitive ensembles, instead of modifying the base classifier in order to accept costs in the learning process, guide the cost minimization procedure via the ensemble learning algorithm. In this way, the modification of the base learner is avoided, but the major drawback, which is the costs definition, is still present.

A complete taxonomy for ensemble methods for learning with imbalanced classes can be found on a recent review [50], which we summarize in Figure 3. Mainly, the authors distinguish four different families among ensemble approaches for imbalanced learning. On the one hand, they identified cost-sensitive boosting approaches which are similar to cost-sensitive methods, but where the costs minimization procedure is guided by a boosting algorithm. On the other hand, they distinguish three more families which have a common feature: all

ACCEPTED MANUSCRIPT

of them consist on embedding a data preprocessing technique in an ensemble learning algorithm. They categorized these three families depending on the ensemble learning algorithm used, i.e. boosting, bagging and hybrid ensembles.

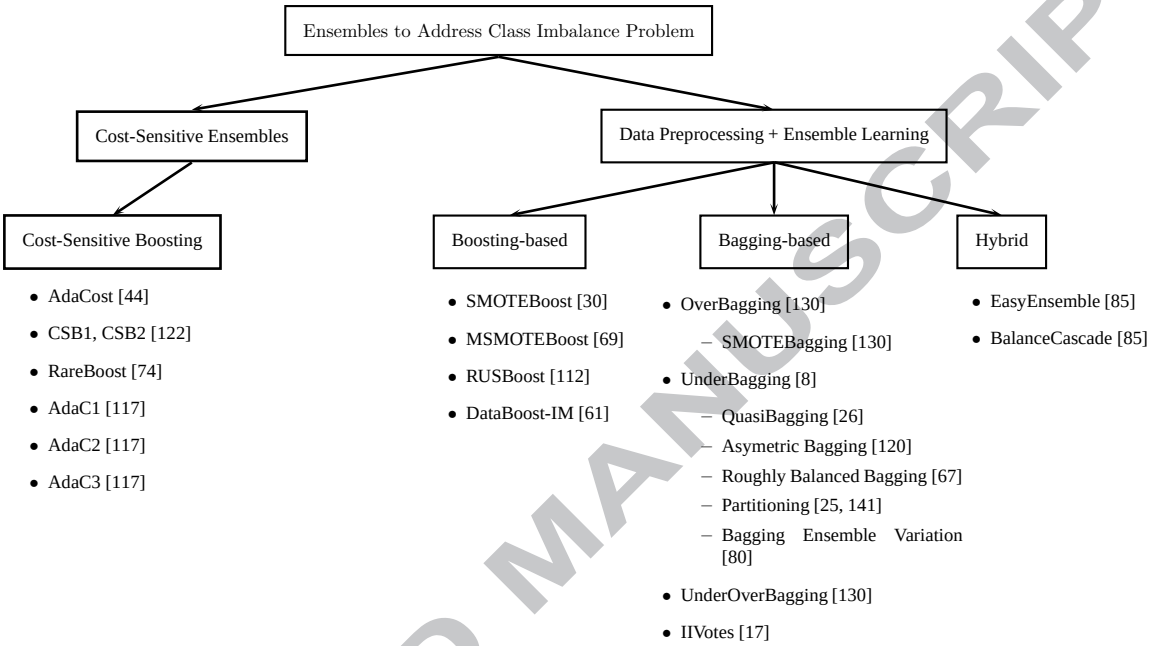


Figure 3: Galar et al.'s proposed taxonomy for ensembles to address class imbalance problem

From the study in [50], the authors concluded that ensemble-based algorithms are worthwhile, improving the results obtained by using data preprocessing techniques and training a single classifier. They also highlighted the good performance of simple approaches such as RUSBoost [112] or UnderBagging [8], which despite of being simple approaches, achieve a higher performance than many other more complex algorithms.

4. Analyzing the Behavior of Imbalanced Learning Methods

Several authors, and especially [9], have developed an ordering of the approaches to address learning with imbalanced datasets regarding a classification metric such as the AUC. In this section we present a complete study on the suitability of some recent proposals for preprocessing, cost-sensitive learning and ensemble-based methods, carrying out an intra-family comparison for selecting the best performing approaches and then developing and inter-family analysis, with the aim of observing whether there are differences among them.

In order to achieve well founded conclusions, we will make use of three classifiers based on different paradigms, namely decision trees with C4.5 [104], Support Vector Machines (SVMs) [35, 100], and the well-known k-Nearest Neighbor

ACCEPTED MANUSCRIPT

(kNN) [92] as an Instance-Based Learning approach. The analysis will be structured in the same manner within each section: first, the average results in training and testing, together with their standard deviations, will be shown for every classifier. Then, the average rankings will be depicted in order to organize the algorithms according to their performance on the different datasets. Finally, the two highest ranked approaches will be selected for the final comparison among all the techniques.

We must remark that this study tries to be carried out in a more descriptive way. For this reason, we just carry out an "ad hoc" selection of the best approaches, even if no significant differences are found in a statistical analysis, which will be performed by means of a Shaffer post-hoc test [113] (n×n comparison). Therefore, the reader must acknowledge that some of the decisions taken along this empirical analysis are carried out for the sake of simplifying our study, thus presenting an overview on the behavior of the state of the art methods on classification with imbalanced data.

According to the previous aim, we divide this section into five parts: first, in Section 4.1 we introduce the experimental framework, that is, the classification algorithms used, their parameters and the selected datasets for the study. Next, we develop a separate study for preprocessing (Section 4.2), cost-sensitive learning (Section 4.3) and ensembles (Section 4.4). As explained earlier, the two best models will be selected as representative approaches and, finally, Section 4.5 presents a global study for the different paradigms that are analyzed.

4.1. Experimental Framework

In the first place, we need to define a set of baseline classifiers to be used in all the experiments. Next, we enumerate these algorithms and also their parameter values, which have been set considering the recommendation of the corresponding authors. We must point out that these algorithms are available within the KEEL software tool [4].

1. **C4.5 Decision Tree [104]**: For C4.5, we have set a confidence level of 0.25, the minimum number of item-sets per leaf was set to 2 and pruning was used as well to obtain the final tree.
2. **Support Vector Machines [35]**: For the SVM, we have chosen *Polykernel reference functions*, with an internal parameter of 1.0 for the exponent of each kernel function and a penalty parameter of the error term of 1.0.
3. **Instance Based Learning (kNN) [92]**: In this case, we have selected 1 neighbor for determining the output class, using the euclidean distance metric.

We have gathered 66 datasets, whose features are summarized in Table 3, namely the number of examples (#Ex.), number of attributes (#Atts.) and IR. Estimates of the AUC metric were obtained by means of a 5-fold cross-validation. That is, we split the dataset into 5 folds, each one containing 20% of the patterns of the dataset. For each fold, the algorithm was trained with the examples contained in the remaining folds and then tested with the current fold.

ACCEPTED MANUSCRIPT

This value is set up with the aim of having enough positive class instances in the different folds, hence avoiding additional problems in the data distribution [94, 96], especially for highly imbalanced datasets.

We must point out that the dataset partitions employed in this paper are available for download at the KEEL dataset repository[1] [3], so that any interested researcher can use the same data for comparison.

Table 3: Summary of imbalanced datasets used

Name	#Ex.	#Atts.	IR	Name	#Ex.	#Atts.	IR
Glass1	214	9	1.82	Glass04vs5	92	9	9.22
Ecoli0vs1	220	7	1.86	Ecoli0346vs5	205	7	9.25
Wisconsin	683	9	1.86	Ecoli0347vs56	257	7	9.28
Pima	768	8	1.90	Yeast05679vs4	528	8	9.35
Iris0	150	4	2.00	Ecoli067vs5	220	6	10.00
Glass0	214	9	2.06	Vowel0	988	13	10.10
Yeast1	1484	8	2.46	Glass016vs2	192	9	10.29
Vehicle1	846	18	2.52	Glass2	214	9	10.39
Vehicle2	846	18	2.52	Ecoli0147vs2356	336	7	10.59
Vehicle3	846	18	2.52	Led7digit02456789vs1	443	7	10.97
Haberman	306	3	2.68	Glass06vs5	108	9	11.00
Glass0123vs456	214	9	3.19	Ecoli01vs5	240	6	11.00
Vehicle0	846	18	3.23	Glass0146vs2	205	9	11.06
Ecoli1	336	7	3.36	Ecoli0147vs56	332	6	12.28
New-thyroid2	215	5	4.92	Cleveland0vs4	177	13	12.62
New-thyroid1	215	5	5.14	Ecoli0146vs5	280	6	13.00
Ecoli2	336	7	5.46	Ecoli4	336	7	13.84
Segment0	2308	19	6.01	Yeast1vs7	459	8	13.87
Glass6	214	9	6.38	Shuttle0vs4	1829	9	13.87
Yeast3	1484	8	8.11	Glass4	214	9	15.47
Ecoli3	336	7	8.19	Page-blocks13vs2	472	10	15.85
Page-blocks0	5472	10	8.77	Abalone9vs18	731	8	16.68
Ecoli034vs5	200	7	9.00	Glass016vs5	184	9	19.44
Yeast2vs4	514	8	9.08	Shuttle2vs4	129	9	20.50
Ecoli067vs35	222	7	9.09	Yeast1458vs7	693	8	22.10
Ecoli0234vs5	202	7	9.10	Glass5	214	9	22.81
Glass015vs2	172	9	9.12	Yeast2vs8	482	8	23.10
Yeast0359vs78	506	8	9.12	Yeast4	1484	8	28.41
Yeast02579vs368	1004	8	9.14	Yeast1289vs7	947	8	30.56
Yeast0256vs3789	1004	8	9.14	Yeast5	1484	8	32.78
Ecoli046vs5	203	6	9.15	Ecoli0137vs26	281	7	39.15
Ecoli01vs235	244	7	9.17	Yeast6	1484	8	39.15
Ecoli0267vs35	224	7	9.18	Abalone19	4174	8	128.87

Finally, with respect to the **evaluation metric**, we use the Area Under the ROC Curve (AUC) [19, 70] as evaluation criteria.

4.2. Study on the preprocessing methods

In this section, we analyze the behavior of the preprocessing methods on imbalanced datasets. For this purpose, we compare some of the most representative techniques, previously presented in Section 3.1, developing a ranking according to the performance obtained in each case. This representative set of methods is composed by the following techniques: SMOTE [27], SMOTE+ENN [9], Borderline-SMOTE (Border-SMOTE) [63], Adaptive Synthetic Sampling (ADASYN) [65], Safe-Level-SMOTE (SL-SMOTE) [21], SPIDER2 [97] and DB-SMOTE [22]. In all cases we try to obtain a level of balance in the training

[1] `http://www.keel.es/datasets.php`

ACCEPTED MANUSCRIPT

data near to the 50:50 distribution. Additionally, the interpolations that are computed to generate new synthetic data are made considering the 5-nearest neighbors of minority class instances using the euclidean distance.

In Table 4 we show the average results for all preprocessing methods, also including the performance with the original data (None). We observe that, in all cases, the oversampling mechanisms are very good solutions for achieving a higher performance by comparison to using the original training data.

Table 4: Average AUC results for the preprocessing techniques

	C4.5		SVM		kNN	
Preprocessing	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
None	.8790 ± .1226	.7873 ± .1437	.7007 ± .1706	.6891 ± .1681	.8011 ± .1339	.8028 ± .1383
SMOTE	.9613 ± .0504	.8288 ± .1192	.8631 ± .1045	**.8470 ± .1152**	.9345 ± .1247	.8341 ± .1194
SMOTE+ENN	.9482 ± .0525	**.8323 ± .1166**	.8815 ± .1001	.8461 ± .1162	.9284 ± .1262	**.8443 ± .1158**
Border-SMOTE	.9333 ± .0595	.8187 ± .1272	.9082 ± .0941	.8397 ± .1163	.9144 ± .0682	.8177 ± .1314
SL-SMOTE	.9175 ± .0615	.8285 ± .1112	.8365 ± .1020	.8427 ± .1176	.8024 ± .1331	.8029 ± .1381
ADASYN	.9589 ± .0469	.8225 ± .1234	.8283 ± .1054	.8323 ± .1148	.9347 ± .0500	.8355 ± .1163
SPIDER2	.9684 ± .0378	.8018 ± .1329	.7252 ± .1493	.7371 ± .1542	.8381 ± .1176	.8207 ± .1338
DBSMOTE	.8908 ± .1006	.7877 ± .1441	.8612 ± .0778	.7546 ± .1368	.8147 ± .1163	.8082 ± .1293

This behavior is contrasted in Figure 4, where we have ordered the corresponding methods according to their AUC results in testing for each dataset, considering the average ranking value. We must stress SMOTE+ENN and SMOTE as the top methodologies, since they obtain the highest rank for the three classification algorithms used in this study. We can also observe that both Border-SMOTE and ADASYN are quite robust on average, obtaining a fair average ranking for all datasets.

For the sake of finding out which algorithms are distinctive among an n×n comparison, we carry out a Shaffer post-hoc test [113], which is shown in Tables 5, 6 and 7. In these tables, a "+" symbol implies that the algorithm in the row is statistically better than the one in the column, whereas "-" implies the contrary; "=" means that the two algorithms compared show no significant differences. In brackets, the adjusted p-value associated to each comparison is shown.

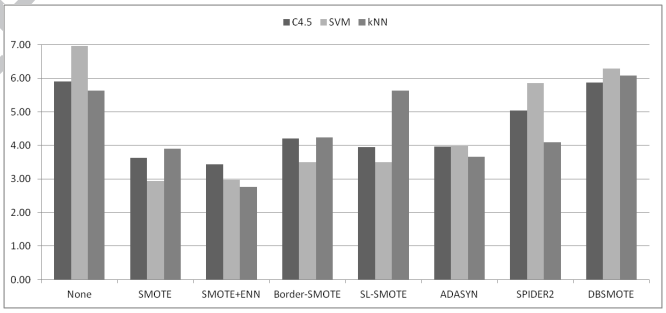


Figure 4: Average Ranking of the preprocessing algorithms for classification with imbalanced datasets

In order to explain why SMOTE+ENN and SMOTE obtain the highest performance, we may emphasize two feasible reasons. The first one is related

ACCEPTED MANUSCRIPT

Table 5: Shaffer test for the preprocessing techniques with C4.5 using the AUC measure

C4.5	None	SMOTE	SMOTE+ENN	Border-SMOTE	SL-SMOTE	ADASYN	SPIDER2	DBSMOTE
None	x	-(.000002)	-(.000000)	-(.001104)	-(.000096)	-(.000124)	=(.580860)	=(1.00000)
SMOTE	+(.000002)	x	=(1.00000)	=(1.00000)	=(1.00000)	=(1.00000)	+(.013398)	+(.000003)
SMOTE+ENN	+(.000000)	=(1.00000)	x	=(.769498)	=(1.00000)	=(1.00000)	+(.002466)	+(.000000)
Border-SMOTE	+(.001104)	=(1.00000)	=(.769498)	x	=(1.00000)	=(1.00000)	=(.631767)	+(.001379)
SL-SMOTE	+(.000096)	=(1.00000)	=(1.00000)	=(1.00000)	x	=(1.00000)	=(.159840)	+(.000124)
ADASYN	+(.000124)	=(1.00000)	=(1.00000)	=(1.00000)	=(1.00000)	x	=(.174600)	+(.000159)
SPIDER2	=(.580860)	-(.013398)	-(.002466)	=(.631767)	=(.159840)	=(.174600)	x	=(.631767)
DBSMOTE	=(1.00000)	-(.000003)	-(.000000)	-(.001379)	-(.000124)	-(.000159)	=(.631767)	x

Table 6: Shaffer test for the preprocessing techniques with SVM using the AUC measure

SVM	None	SMOTE	SMOTE+ENN	Border-SMOTE	SL-SMOTE	ADASYN	SPIDER2	DBSMOTE
None	x	-(.000000)	-(.000000)	-(.000000)	-(.000000)	-(.000000)	=(.129870)	=(1.00000)
SMOTE	+(.000000)	x	=(1.00000)	=(1.00000)	=(1.00000)	=(.179175)	+(.000000)	+(.000000)
SMOTE+ENN	+(.000000)	=(1.00000)	x	=(1.00000)	=(1.00000)	=(.199418)	+(.000000)	+(.000000)
Border-SMOTE	+(.000000)	=(1.00000)	=(1.00000)	x	=(1.00000)	=(1.00000)	+(.000000)	+(.000000)
SL-SMOTE	+(.000000)	=(1.00000)	=(1.00000)	=(1.00000)	x	=(1.00000)	+(.000000)	+(.000000)
ADASYN	+(.000000)	=(.179175)	=(.199418)	=(1.00000)	=(1.00000)	x	+(.000126)	+(.000001)
SPIDER2	=(.129870)	-(.000000)	-(.000000)	-(.000000)	-(.000000)	-(.000126)	x	=(1.00000)
DBSMOTE	=(1.00000)	-(.000000)	-(.000000)	-(.000000)	-(.000000)	-(.000001)	=(1.00000)	x

Table 7: Shaffer test for the preprocessing techniques with kNN using the AUC measure

kNN	None	SMOTE	SMOTE+ENN	Border-SMOTE	SL-SMOTE	ADASYN	SPIDER2	DBSMOTE
None	x	-(.000757)	-(.000000)	-(.014934)	=(1.00000)	-(.000081)	-(.004963)	=(1.00000)
SMOTE	+(.000757)	x	-(.089266)	=(1.00000)	+(.000701)	=(1.00000)	=(1.00000)	+(.000006)
SMOTE+ENN	+(.000000)	+(.089266)	x	+(.007968)	+(.000000)	=(.360402)	+(.022513)	+(.000000)
Border-SMOTE	+(.014934)	=(1.00000)	-(.007968)	x	+(.014027)	=(1.00000)	=(1.00000)	+(.000253)
SL-SMOTE	=(1.00000)	-(.000701)	-(.000000)	-(.014027)	x	-(.000074)	-(.004634)	=(1.00000)
ADASYN	+(.000081)	=(1.00000)	=(.360402)	=(1.00000)	+(.000074)	x	=(1.00000)	+(.000000)
SPIDER2	+(.004963)	=(1.00000)	-(.022513)	=(1.00000)	+(.004634)	=(1.00000)	x	+(.000062)
DBSMOTE	=(1.00000)	-(.000006)	-(.000000)	-(.000253)	=(1.00000)	-(.000000)	-(.000062)	x

to the addition of significant information within the minority class examples by including new synthetic examples. These new examples allow the formation of larger clusters to help the classifiers to separate both classes, and the cleaning procedure also adds benefits to the generalisation ability during learning. The second reason is that the more sophisticated the technique is, the less general it becomes for the high number of benchmark problems selected for our study.

According to these results, we select both SMOTE+ENN and SMOTE as good behaving methodologies for our final comparison.

4.3. Study on the cost-sensitive learning algorithms

In this section, we carry out an analysis regarding cost-sensitive classifiers. We use three different approaches, namely "Weighted-Classifier" (CS-Weighted) [7, 123], MetaCost [38], and the CostSensitive Classifier (CS-Classifier) from the Weka environment [62]. In the first case, the base classifiers are modified usually by weighting the instances of the dataset to take into account the a priori probabilities, according to the number of samples in each class. In the two latter cases, we use an input cost-matrix defining $C(+,-) = IR$ and $C(-,+) = 1$.

Table 8 shows the average AUC results. From this table, we may conclude, as in the previous case for preprocessing, the goodness of the use of this type of solution for imbalanced data, as there is a significant difference with respect to the results obtained with the original data. We may also observe the good

ACCEPTED MANUSCRIPT

behavior of the "CS-Weighted" in contrast with the remaining techniques, and also the good accuracy for the MetaCost algorithm, for both C4.5 and kNN.

Table 8: Average AUC results for the cost-sensitive learning techniques

	C4.5		SVM		kNN	
Cost-Sensitive	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
None	.8790 ± .1226	.7873 ± .1437	.7007 ± .1706	.6891 ± .1681	.8011 ± .1339	.8028 ± .1383
CS-Weighted	.9711 ± .0580	.8284 ± .1263	.8751 ± .1068	**.8464 ± .1124**	.8427 ± .1201	**.8463 ± .1177**
MetaCost	.9159 ± .0797	**.8370 ± .1287**	.6931 ± .1715	.6802 ± .1696	.9849 ± .0118	.8250 ± .1301
CS-Classifier	.8915 ± .1191	.8116 ± .1387	.8701 ± .1053	.8391 ± .1152	.9993 ± .0046	.8084 ± .1343

Figure 5 presents the ranking for the selected methods. We can appreciate that the "CS-Weighted" approach achieves the highest rank overall, as pointed out before. The MetaCost method obtains also a good average for C4.5 and kNN, but it is outperformed by the CS-Classifier when SVM is used.

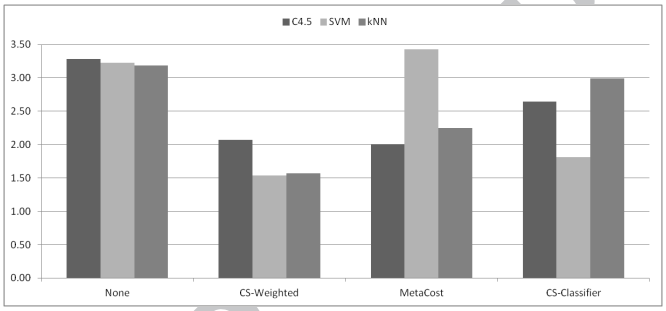


Figure 5: Average Ranking of the cost-sensitive learning algorithms for the classification with imbalanced datasets

As in the latter case, we show a Shaffer post-hoc test for detecting significant differences among the results (Tables 9,10 and 11).

Table 9: Shaffer test for the cost-sensitive learning techniques with C4.5 using the AUC measure

C4.5	None	CS-Weighted	MetaCost	CS-Classifier
None	x	-(.000000)	-(.000000)	-(.013893)
CS-Weighted	+(.000000)	x	=(.787406)	+(.020817)
MetaCost	+(.000000)	=(.787406)	x	+(.013893)
CS-Classifier	+(.013893)	-(.020817)	-(.013893)	x

The good behavior shown by introducing weights to the training examples can be explained by its simplicity, because the algorithm procedure is maintained and is adapted to the imbalanced situation. Therefore, it works similarly to an oversampling approach but without adding new samples and complexity to the problem itself. On the other hand, the MetaCost method follows a similar aim, therefore obtaining high quality results. Regarding these facts, we will select these two methods as the representative ones for this family.

ACCEPTED MANUSCRIPT

Table 10: Shaffer test for the cost-sensitive learning techniques with SVM using the AUC measure

SVM	None	CS-Weighted	MetaCost	CS-Classifier
None	x	-(.000000)	=(.449832)	-(.000000)
CS-Weighted	+(.000000)	x	+(.000000)	=(.449832)
MetaCost	=(.449832)	-(.000000)	x	-(.000000)
CS-Classifier	+(.000000)	=(.449832)	+(.000000)	x

Table 11: Shaffer test for the cost-sensitive learning techniques with kNN using the AUC measure

kNN	None	CS-Weighted	MetaCost	CS-Classifier
None	x	-(.000000)	-(.000075)	=(.345231)
CS-Weighted	+(.000000)	x	+(.004828)	+(.000000)
MetaCost	+(.000075)	-(.004828)	x	+(.003228)
CS-Classifier	=(.345231)	-(.000000)	-(.003228)	x

4.4. Study on the ensemble-based techniques

The last family of approaches for dealing with imbalanced datasets that we will analyze is the one based on ensemble techniques. In this case, we have selected five different algorithms which showed a very good behavior on the study carried out in [50], namely AdaBoost.M1 (AdaB-M1) [110], AdaBoost with costs outside the exponent (AdaC2) [117], RUSBoost (RUSB) [112], SMOTE-Bagging (SBAG) [130], and EasyEnsemble (EASY) [85]. We must point out that AdaB-M1 was not included in the taxonomy presented in Section 3.3 since it is not strictly oriented towards imbalanced classification, but we have decided to study it as a classical ensemble approach and because it has shown a good behavior in [50]. Regarding the number of internal classifiers used within each approach, AdaB-M1, AdaC2 and SBAG use 40 classifiers, whereas the remaining approaches use only 10. Additionally, EASY considers 4 bags for the learning stage.

In this case, the average AUC results for training and testing are shown in Table 12. From this table we may conclude the good performance of RUSB, SBAG and EASY. Among them, SBAG stands out for obtaining slightly better results. Anyway, these three algorithms outperform the others considered in this study. The reader might have also noticed that, the great behavior of RUSB is attained using only 10 base classifiers.

This can also be seen from Figure 6, where we can observe that these three algorithms obtain the first rank positions in almost all cases. It is noticeable that RUSB decreases its results in the case of the SVM algorithm, which can be due to the removal of significant samples for determining the support vectors for the margin classifier in each iteration of the learning.

Tables 13 to 15 present a Shaffer test, where we can observe, in a nutshell, the statistical differences among the ensemble methodologies selected for this

ACCEPTED MANUSCRIPT

Table 12: Average AUC results for the ensemble methodologies

	C4.5		SVM		kNN	
Ensemble	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
None	.8790 ± .1226	.7873 ± .1437	.7007 ± .1706	.6891 ± .1681	.8011 ± .1339	.8028 ± .1383
AdaB-M1	.9915 ± .0468	.8072 ± .1334	.7862 ± .1659	.7615 ± .1630	.9983 ± .0101	.8090 ± .1345
AdaC2	.9470 ± .0858	.8188 ± .1312	.6366 ± .1497	.6271 ± .1479	.9991 ± .0062	.8080 ± .1344
RUSB	.9481 ± .0545	.8519 ± .1129	.7667 ± .1652	.7517 ± .1642	.9359 ± .0495	.8465 ± .1118
SBAG	.9626 ± .0455	**.8545 ± .1111**	.8662 ± .1050	**.8456 ± .1137**	.9825 ± .0253	**.8485 ± .1164**
Easy	.9076 ± .0626	.8399 ± .1091	.8565 ± .1057	.8370 ± .1150	.9093 ± .0667	.8440 ± .1095

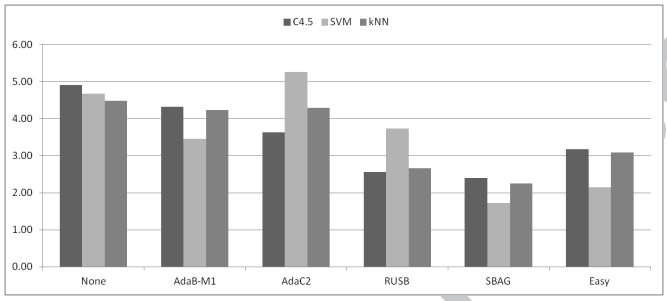


Figure 6: Average Ranking of the ensemble algorithms for the classification with imbalanced datasets

study.

Table 13: Shaffer test for the ensemble methodologies with C4.5 using the AUC measure

C4.5	None	AdaB-M	AdaC2	RUSB	SBAG	Easy
None	x	=(.214054)	-(.000767)	-(.000000)	-(.000000)	-(.000001)
AdaB-M	=(.214054)	x	=(.137090)	-(.000001)	-(.000000)	-(.00339)
AdaC2	+(.000767)	=(.137090)	x	-(.006691)	-(.00115)	=(.339838)
RUSB	+(.000000)	+(.000001)	+(.006691)	x	=(.641758)	=(.214054)
SBAG	+(.000000)	+(.000000)	+(.00115)	=(.641758)	x	+(.099451)
Easy	+(.000001)	+(.003390)	=(.339838)	=(.214054)	-(.099451)	x

Table 14: Shaffer test for the ensemble methodologies with SVM using the AUC measure

SVM	None	AdaB-M	AdaC2	RUSB	SBAG	Easy
None	x	-(.000721)	=(.208828)	-(.015681)	-(.000000)	-(.000000)
AdaB-M	+(.000721)	x	+(.000000)	=(.401501)	-(.000001)	-(.000343)
AdaC2	=(.208828)	-(.000000)	x	-(.000018)	-(.000000)	-(.000000)
RUSB	+(.015681)	=(.401501)	+(.000018)	x	-(.000000)	-(.000007)
SBAG	+(.000000)	+(.000001)	+(.000000)	+(.000000)	x	=(.401501)
Easy	+(.000000)	+(.000343)	+(.000000)	+(.000007)	=(.401501)	x

Nevertheless, we must point out that more complex methods do not perform much better than simpler ones. Bagging techniques are easy to develop, but also powerful when dealing with class imbalance if they are properly combined. Their hybridization with data preprocessing techniques has shown competitive results

ACCEPTED MANUSCRIPT

Table 15: Shaffer test for the ensemble methodologies with kNN using the AUC measure

kNN	None	AdaB-M	AdaC2	RUSB	SBAG	Easy
None	x	=(1.00000)	=(1.00000)	-(.000000)	-(.000000)	-(.000118)
AdaB-M	=(1.00000)	x	=(1.00000)	-(.000017)	-(.000000)	-(.003106)
AdaC2	=(1.00000)	=(1.00000)	x	-(.000006)	-(.000000)	-(.001517)
RUSB	+(.000000)	+(.000017)	+(.000006)	x	=(.803003)	=(.803003)
SBAG	+(.000000)	+(.000000)	+(.000000)	=(.803003)	x	+(.063015)
Easy	+(.000118)	+(.003106)	+(.001517)	=(.803003)	-(.063015)	x

and the key issue of these methods resides in properly exploiting the diversity when each bootstrap replica is formed.

Since we have to select only two methodologies for the global analysis, we will stress SBAG as the best ranked method and RUSB, because it presents a robust behavior on average and the second best mean performance in two of the three algorithms.

4.5. Global analysis for the methodologies that address imbalanced classification

In this last section of the experimental analysis on the behavior of the methodologies for addressing classification with imbalanced datasets, we will perform a cross-family comparison for the approaches previously selected as the representatives for each case, namely preprocessing (SMOTE and SMOTE+ENN), cost-sensitive learning (CS-Weighted and MetaCost) and ensemble techniques (RUSB and SBAG). The global results are shown in Table 16, whereas the new performance ranking is shown in Figure 7.

Table 16: Average global results for C4.5 with the representative methodologies for addressing imbalanced classification

	C4.5		SVM		kNN	
Preprocessing	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
None	.8790 ± .1226	.7873 ± .1437	.7007 ± .1706	.6891 ± .1681	.8011 ± .1339	.8028 ± .1383
SMOTE	.9613 ± .0504	.8288 ± .1192	.8631 ± .1045	**.8470 ± .1152**	.9345 ± .1247	.8341 ± .1194
SMOTE+ENN	.9482 ± .0525	.8323 ± .1166	.8815 ± .1001	.8461 ± .1162	.9284 ± .1262	.8443 ± .1158
CS-Weighted	.9711 ± .0580	.8284 ± .1263	.8751 ± .1068	.8464 ± .1124	.8427 ± .1201	.8463 ± .1177
MetaCost	.9159 ± .0797	.8370 ± .1287	.6931 ± .1715	.6802 ± .1696	.9849 ± .0118	.8250 ± .1301
RUSB	.9481 ± .0545	.8519 ± .1129	.7667 ± .1652	.7517 ± .1642	.9359 ± .0495	.8465 ± .1118
SBAG	.9626 ± .0455	**.8545 ± .1111**	.8662 ± .1050	.8456 ± .1137	.9825 ± .0253	**.8485 ± .1164**

Considering these results, we must highlight the dominance of the ensemble approaches versus the remaining models for the "weak classifiers", i.e. C4.5 and kNN. For SVM, the best results are achieved by preprocessing and CS-weighted, showing the significance of adjusting the objective function towards the positive instances, for biasing the separating hyperplane. Regarding the comparison between the cost-sensitive classifiers and the oversampling methods, we observe that, on average, SMOTE+ENN, CS-Weighted and SMOTE obtain very good results and, therefore, they have a similar ranking, followed by the MetaCost method. We must point out that these conclusions regarding the latter techniques are in concordance with the study done in [88].

ACCEPTED MANUSCRIPT

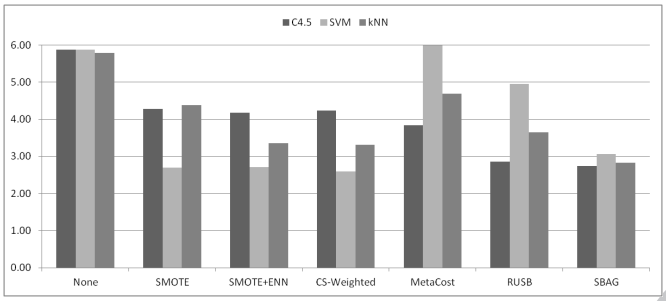


Figure 7: Average Ranking of the representative algorithms for the classification with imbalanced datasets

In the same way as in the previous sections of this study, we proceed with a Shaffer test (Tables 17, 18 and 19) that aims to contrast whether two algorithms are significantly different and how different they are.

Table 17: Shaffer test for the representative methodologies with C4.5 using the AUC measure

C4.5	None	SMOTE	SMOTE+ENN	CS-Weighted	MetaCost	RUSB	SBAG
None	x	-(.000292)	-(.000087)	-(.000203)	-(.000001)	-(.000000)	-(.000000)
SMOTE	+(.000292)	x	=(1.00000)	=(1.00000)	=(1.00000)	-(.001816)	-(.000648)
SMOTE+ENN	+(.000087)	=(1.00000)	x	=(1.00000)	=(1.00000)	-(.004560)	-(.001423)
CS-Weighted	+(.000203)	=(1.00000)	=(1.00000)	x	=(1.00000)	-(.002500)	-(.000671)
MetaCost	+(.000001)	=(1.00000)	=(1.00000)	=(1.00000)	x	-(.061745)	-(.02942)
RUSB	+(.000000)	+(.001816)	+(.004560)	+(.002500)	+(.061745)	x	=(1.00000)
SBAG	+(.000000)	+(.000648)	+(.001423)	+(.000671)	+(.02942)	=(1.00000)	x

Table 18: Shaffer test for the representative methodologies with SVM using the AUC measure

SVM	None	SMOTE	SMOTE+ENN	CS-Weighted	MetaCost	RUSB	SBAG
None	x	-(.000000)	-(.000000)	-(.000000)	=(1.00000)	-(.097865)	-(.000000)
SMOTE	+(.000000)	x	=(1.00000)	=(1.00000)	+(.000000)	+(.000000)	=(1.00000)
SMOTE+ENN	+(.000000)	=(1.00000)	x	=(1.00000)	+(.000000)	+(.000000)	=(1.00000)
CS-Weighted	+(.000000)	=(1.00000)	=(1.00000)	x	+(.000000)	+(.000000)	=(1.00000)
MetaCost	=(1.00000)	-(.000000)	-(.000000)	-(.000000)	x	-(.019779)	-(.000000)
RUSB	+(.097865)	-(.000000)	-(.000000)	-(.000000)	+(.019779)	x	-(.000005)
SBAG	+(.000000)	=(1.00000)	=(1.00000)	=(1.00000)	+(.000000)	+(.000005)	x

Table 19: Shaffer test for the representative methodologies with kNN using the AUC measure

kNN	None	SMOTE	SMOTE+ENN	CS-Weighted	MetaCost	RUSB	SBAG
None	x	-(.002684)	-(.000000)	-(.000000)	-(.038367)	-(.000000)	-(.000000)
SMOTE	+(.002684)	x	-(.058815)	-(.049543)	=(1.00000)	=(.371813)	-(.000545)
SMOTE+ENN	+(.000000)	+(.058815)	x	=(1.00000)	+(.004309)	=(1.00000)	=(.950901)
CS-Weighted	+(.000000)	+(.049543)	=(1.00000)	x	+(.002705)	=(1.00000)	=(.986440)
MetaCost	+(.038367)	=(1.00000)	-(.004309)	-(.002705)	x	-(.057811)	-(.000011)
RUSB	+(.000000)	=(.371813)	=(1.00000)	=(1.00000)	+(.057811)	x	=(.196710)
SBAG	+(.000000)	+(.000545)	=(.950901)	=(.986440)	+(.000011)	=(.196710)	x

As a final remark, we must state that all the solutions analyzed here present different particularities, which make them more appropriate for a given application. For example, ensemble methodologies have shown to be very accurate,

ACCEPTED MANUSCRIPT

but their learning time may be high and the output model can be difficult to comprehend by the final user. Cost-sensitive approaches have also shown to be very precise, but the necessity of defining an optimal cost-matrix impose hard restrictions to their use. Finally, the preprocessing algorithms have shown their robustness and obtained very good global results, and therefore they can be viewed as a standard approach for imbalanced datasets.

5. Problems Related to Data Intrinsic Characteristics in Imbalanced Classification

As it was stated in the introduction of this work, skewed class distributions do not hinder the learning task by itself [66, 118], but usually a series of difficulties related with this problem turn up. This issue is depicted in Figure 8, in which we show the performance of the SBAG with the different datasets used in the previous section, ordered according to the IR, in order to search for some regions of interesting good or bad behavior. As we can observe, there is no pattern of behavior for any range of IR, and the results can be poor both for low and high imbalanced data.

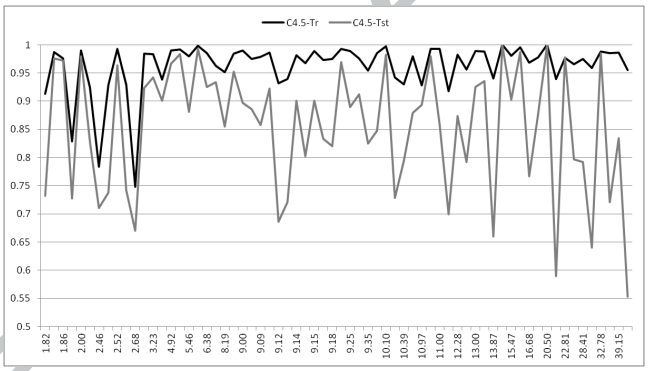


Figure 8: Performance in training and testing for the C4.5 decision tree with SBAG as a function of IR

Related to this issue, in this section we aim to make a discussion on the nature of the problem itself, emphasizing several data intrinsic characteristics that do have a strong influence on imbalanced classification, in order to be able to address this problem in a more feasible way.

With this objective in mind, we focus our analysis on using the C4.5 classifier, in order to develop a basic but descriptive study by showing a series of patterns of behavior, following a kind of "educational scheme". With respect to the previous section, which was carried out in an empirical way, this part of the study is devoted to enumerating the scenarios that can be found when dealing with classification with imbalanced data, emphasizing their main issues that will allow us to design a better algorithm that can be adapted to different niches of the problem.

ACCEPTED MANUSCRIPT

We acknowledge that some of the data intrinsic characteristics described along this section share some features and it is usual that, for a given dataset, several "sub-problems" can be found simultaneously. Nevertheless, we consider a simplified view of all these scenarios to serve as a global introduction to the topic.

First, we discuss about the difficulties related to the presence of small disjuncts in the imbalanced data (Section 5.1). Then, we present the issues about the size of the dataset and the lack of density in the training set (Section 5.2). Next, we focus on the class overlap, showing that it is extremely significant on imbalanced domains (Section 5.3). Then, we analyze the presence of noisy data in this type of problems and how it affects the behavior of both preprocessing techniques and classification algorithms (Section 5.4). After that, we introduce the concept of borderline instances and its relationship with noise examples (Section 5.5). Finally, we define the dataset shift problem in the classification with imbalanced datasets (Section 5.6).

5.1. Small disjuncts

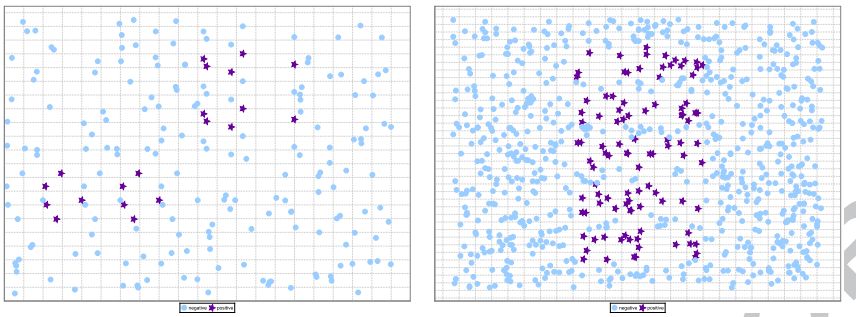
The presence of the imbalanced classes is closely related to the problem of small disjuncts. This situation occurs when the concepts are represented within small clusters, which arise as a direct result of underrepresented subconcepts [99, 138]. Although those small disjuncts are implicit in most of the problems, the existence of this type of areas highly increases the complexity of the problem in the case of class imbalance, because it becomes hard to know whether these examples represent an actual subconcept or are merely attributed to noise [73]. This situation is represented in Figure 9, where we show an artificially generated dataset with small disjuncts for the minority class and the *"Subclus"* problem created in [97], where we can find small disjuncts for both classes: the negative samples are underrepresented with respect to the positive samples in the central region of positive rectangular areas, while the positive samples only cover a small part of the whole dataset and are placed inside the negative class. We must point out that, in all figures of this section, positive instances are represented with dark stars whereas negative instances are depicted with light circles.

The problem of small disjuncts becomes accentuated for those classification algorithms which are based on a divide-and-conquer approach [135]. This methodology consists in subdividing the original problem into smaller ones, such as the procedure used in decision trees, and can lead to data fragmentation [49], that is, to obtain several partitions of data with a few representation of instances. If the IR of the data is high, this handicap is obviously more severe.

Several studies by Weiss [136, 137] analyze this factor in depth and enumerate several techniques for handling the problem of small disjuncts:

1. **Obtain additional training data**. The lack of data can induce the apparition of small disjuncts, especially in the minority class, and these areas may be better covered just by employing an informed sampling scheme [71].

ACCEPTED MANUSCRIPT



(a) Artificial dataset: small disjuncts for the minority class

(b) Subclus dataset: small disjuncts for both classes

Figure 9: Example of small disjuncts on imbalanced data

2. **Use a more appropriate inductive bias**. If we aim to be able to properly detect the areas of small disjuncts, some sophisticated mechanisms must be employed for avoiding the preference for the large areas of the problem. For example, [68] modified CN2 so that its maximum generality bias is used only for large disjuncts, and a maximum specificity bias was then used for small disjuncts. However, this approach also degrades the performance of the small disjuncts, and some authors proposed to refine the search and to use different learners for the examples that fall in the large disjuncts and on the small disjuncts separately [24, 121].
3. **Using more appropriate metrics**. This issue is related to the previous one in the sense that, for the data mining process, it is recommended to use specific measures for imbalanced data, in a way that the minority classes in the small disjuncts are positively weighted when obtaining the classification model [134]. For example, the use of precision and recall for the minority and majority classes, respectively, can lead to generate more precise rules for the positive class [41, 74].
4. **Disabling pruning**. Pruning tends to eliminate most small disjuncts by a generalization of the obtained rules. Therefore, this methodology is not recommended.
5. **Employ boosting**. Boosting algorithms, such as the AdaBoost algorithm, are iterative algorithms that place different weights on the training distribution each iteration [110]. Following each iteration, boosting increases the weights associated with the incorrectly classified examples and decreases the weights associated with the correctly classified examples. Because instances in the small disjuncts are known to be difficult to predict, it is reasonable to believe that boosting will improve their classification performance. Following this idea, many approaches have been developed by modifying the standard boosting weight-update mechanism in order to improve the performance on the minority class and the small disjuncts [30, 44, 61, 69, 74, 112, 117, 122].

ACCEPTED MANUSCRIPT

Finally, we must emphasize the use of the CBO method [73], which is a resampling strategy that is used to counteract simultaneously the between-class imbalance and the within-class imbalance. Specifically, this approach detects the clusters in the positive and negative classes using the k-means algorithm in a first step. In a second step, it randomly replicates the examples for each cluster (except the largest negative cluster) in order to obtain a balanced distribution between clusters from the same class and between classes. These clusters can be viewed as small disjuncts in the data, and therefore this preprocessing mechanism is aimed to stress the significance of these regions.

In order to show the goodness of this approach, we depict a short analysis on the two previously presented artificial datasets, that is, our artificial problem and the Subclus dataset, studying the behavior of the C4.5 classifier according to both the differences in performance between the original and the preprocessed data and the boundaries obtained in each case. We must point out that the whole dataset is used in both cases.

Table 20 shows the results of C4.5 in each case, where we must emphasize that the application of CBO enables the correct identification of all the examples for both classes. Regarding the visual output of the C4.5 classifier (Figure 10), in the first case we observe that for the original data no instances of the positive class are recognized, and that there is an overgeneralization of the negative instances, whereas the CBO method achieves the correct identification of the four clusters in the data, by replicating an average of 11.5 positive examples and 1.25 negative examples. In the Subclus problem, there is also an overgeneralization for the original training data, but in this case we found that the small disjuncts of the negative class surrounding the positive instances are the ones which are misclassified now. Again, the application of the CBO approach results on a perfect classification for all data, having 7.8 positive instances for each "data point" and 1.12 negative ones.

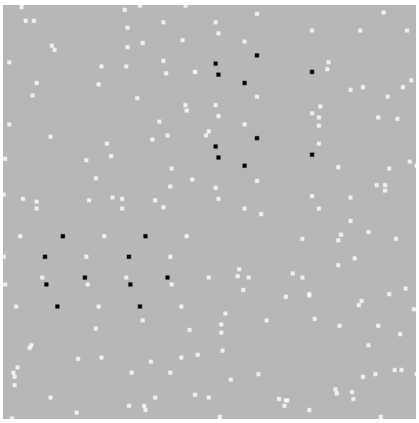
Table 20: Performance obtained by C4.5 in datasets suffering from small disjuncts

Dataset	Original Data			Preprocessed Data with CBO		
	TP_{rate}	TN_{rate}	AUC	TP_{rate}	TN_{rate}	AUC
Artificial dataset	.0000	1.000	.5000	1.000	1.000	1.000
Subclus dataset	1.000	.9029	.9514	1.000	1.000	1.000

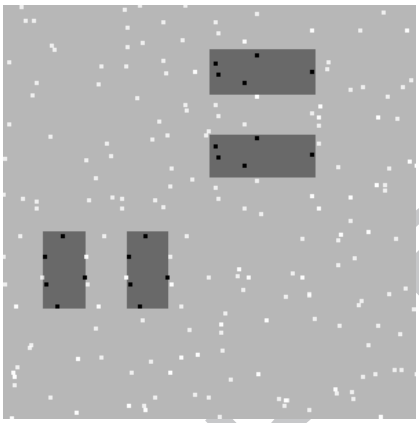
5.2. Lack of density

One problem that can arise in classification is the small sample size [106]. This issue is related to the "lack of density" or "lack of information", where induction algorithms do not have enough data to make generalizations about the distribution of samples, a situation that becomes more difficult in the presence of high dimensional and imbalanced data. A visual representation of this problem is depicted in Figure 11, where we show a scatter plot for the training data of the yeast4 problem (attributes mcg vs. gvh) only with a 10 % of the original instances (Figure 11(a)) and with all the data (Figure 11(b)). We can appreciate

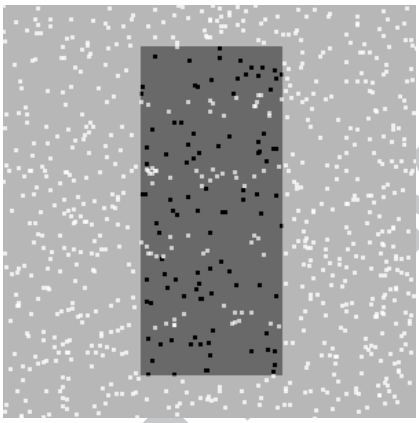
ACCEPTED MANUSCRIPT



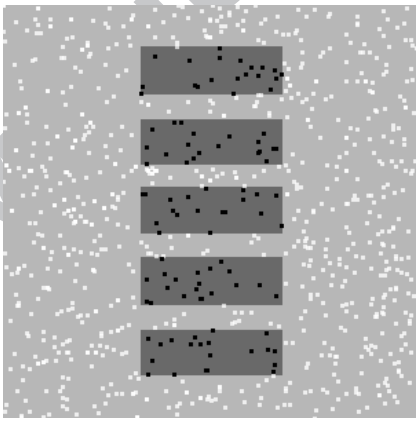
(a) Artificial dataset with the original data: 20 positive and 182 negative instances



(b) Artificial dataset with CBO: 228 positive and 228 negative instances



(c) Subclus dataset with the original data: 100 positive and 700 negative instances



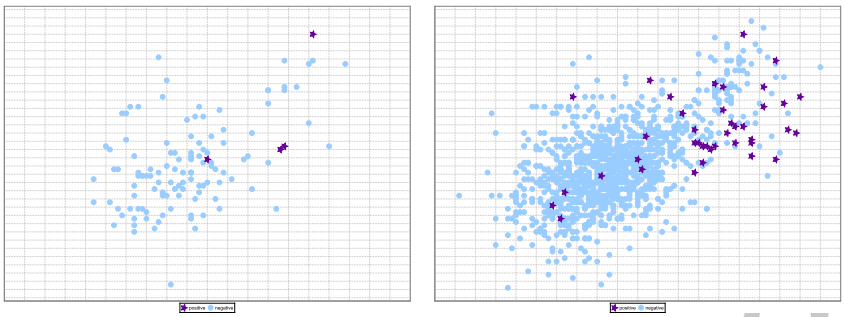
(d) Subclus dataset with CBO: 780 positive and 780 negative instances

Figure 10: Boundaries obtained by C4.5 with the original and preprocessed data using CBO for addressing the problem of small disjuncts. The new instances for (b) and (d) are just replicates of the initial examples.

that it becomes very hard for the learning algorithm to obtain a model that is able to perform a good generalization when there is not enough data that represents the boundaries of the problem and, what it is also most significant, when the concentration of minority examples is so low that they can be simply treated as noise.

The combination of imbalanced data and the small sample size problem presents a new challenge to the research community [133]. In this scenario, the minority class can be poorly represented and the knowledge model to learn this data space becomes too specific, leading to overfitting. Furthermore, as stated in the previous section, the lack of density in the training data may also

ACCEPTED MANUSCRIPT



(a) 10 % of training instances (b) 100 % of training instances

Figure 11: Lack of density or small sample size on the yeast4 dataset

cause the introduction of small disjuncts. Therefore, two datasets can not be considered to present the same complexity because they have the same IR, as it is also important how the training data represents the minority instances.

In [138], the authors have studied the effect of class distribution and training-set size on the classifier performance using C4.5 as base learning algorithm. Their analysis consisted in varying both the available training data and the degree of imbalance for several datasets and observing the differences for the AUC metric in those cases.

The first finding they extracted is somehow quite trivial, that is, the higher the number of training data, the better the performance results are, independently of the class distribution. A second important fact that they highlighted is that the IR that yields the best performances occasionally vary from one training-set size to another, giving the support to the notion that there may be a "best" marginal class distribution for a learning task and suggests that a progressive sampling algorithm may be useful in locating the class distribution that yields the best, or nearly best, classifier performance.

In order to visualize the effect of the density of examples in the learning process, in Figure 12 we show the results in AUC for the C4.5 classifier both for training (black line) and testing (grey line) for the *vowel0* problem, varying the percentage of training instances from 10% to the original training size. This short experiment is carried out on a 5-fold cross validation, where the test data is not modified, i.e. in all cases it represents a 20% of the original data; the results shown are the average of the five partitions.

From this graph, we may distinguish a growth rate directly proportional to the number of training instances that are being used. This behavior reflects the findings enumerated previously from [138].

5.3. Overlapping or class separability

The problem of overlapping between classes appears when a region of the data space contains a similar quantity of training data from each class. This

ACCEPTED MANUSCRIPT

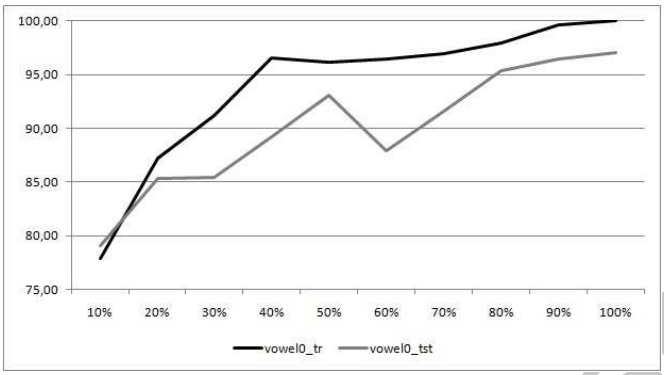


Figure 12: AUC performance for the C4.5 classifier with respect to the proportion of examples in the training set for the vowel0 problem

situation leads to develop an inference with almost the same a priori probabilities in this overlapping area, which makes very hard or even impossible the distinction between the two classes. Indeed, any "linearly separable" problem can be solved by any simple classifier regardless of the class distribution.

There are several works which aim to study the relationship between overlapping and class imbalance. Particularly, in [102] one can find a study where the authors propose several experiments with synthetic datasets varying the imbalance ratio and the overlap existing between the two classes. Their conclusions stated that the class probabilities are not the main responsibles for the hinder in the classification performance, but instead the degree of overlapping between the classes.

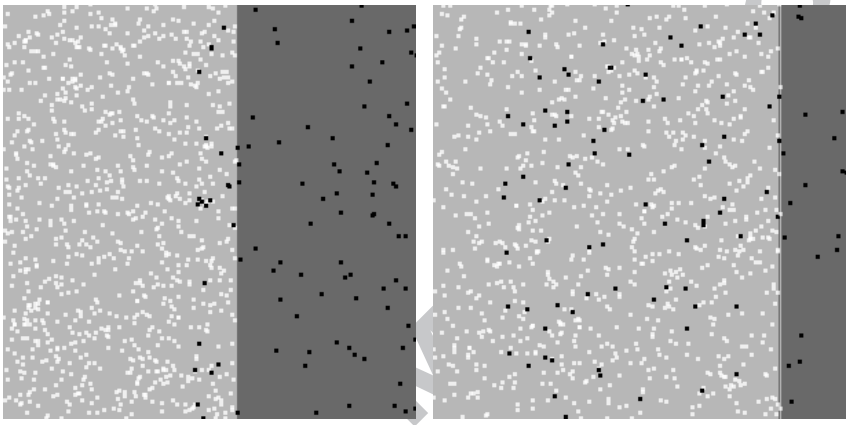
To reproduce the example for this scenario, we have created an artificial dataset with 1,000 examples having an IR of 9, i.e. 1 positive instance per 10 instances. Then, we have varied the degree of overlap for individual feature values, from no overlap to 100% of overlap, and we have used the C4.5 classifier in order to determine the influence of overlapping with respect to a fixed IR. First, Table 21 shows the results for the considered cases, where we observe that the performance is highly degrading with the increase of the overlap. Additionally, Figure 13 shows this issue, where we can observe that the decision tree is not only unable to obtain a correct discrimination between both classes when they are overlapped, but also that the preferred class is the majority one, leading to low values for the AUC metric.

Additionally, in [55], a similar study with several algorithms in different situations of imbalance and overlap focusing on the the kNN algorithm was developed. In this case, the authors proposed two different frameworks: on the one hand, they try to find the relation when the imbalance ratio in the overlap region is similar to the overall imbalance ratio whereas, on the other hand, they search for the relation when the imbalance ratio in the overlap region is inverse to the overall one (the positive class is locally denser than the negative class in the overlap region). They showed that when the overlapped data is not

ACCEPTED MANUSCRIPT

Table 21: Performance obtained by C4.5 with different degrees of overlapping

Overlap Degree	TP_{rate}	TN_{rate}	AUC
0 %	1.000	1.000	1.000
20 %	.79.00	1.000	.8950
40 %	.4900	1.000	.7450
50 %	.4700	1.000	.7350
60 %	.4200	1.000	.7100
80 %	.2100	.9989	.6044
100 %	.0000	1.000	.5000



(a) 20% overlap (b) 80% overlap

Figure 13: Example of overlapping imbalanced datasets: boundaries detected by C4.5

balanced, the IR in overlapping can be more important than the overlapping size. In addition, classifiers using a more global learning procedure attain greater TP rates whereas more local learning models obtain better TN rates than the former.

In [37], the authors examine the effects of overlap and imbalance on the complexity of the learned model and demonstrate that overlapping is a far more serious factor than imbalance in this respect. They demonstrate that these two problems acting in concert cause difficulties that are more severe than one would expect by examining their effects in isolation. In order to do so, they also use synthetic datasets for classifying with a SVM, where they vary the imbalance ratio, the overlap between classes and the imbalance ratio and overlap jointly. Their results show that, when the training set size is small, high levels of imbalance cause a dramatic drop in classifier performance, explained by the presence of small disjuncts. Overlapping classes cause a consistent drop in performance regardless of the size of the training set. However, with overlapping and imbalance combined, the classifier performance is degraded significantly beyond what the model predicts.

In one of the latest researches on the topic [89], the authors have empirically

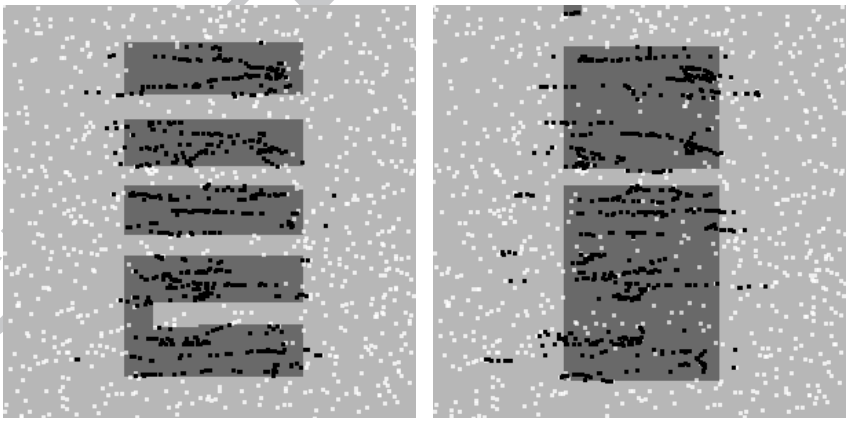
ACCEPTED MANUSCRIPT

extracted some interesting findings on real world datasets. Specifically, the authors depicted the performance of the different datasets ordered according to different data complexity measures (including the IR) in order to search for some regions of interesting good or bad behavior. They could not characterize any interesting behavior related to IR, but they do for other metrics that measure the overlap between the classes.

Finally, in [90], an approach that combines preprocessing and feature selection (strictly in this order) is proposed. This approach works in a way where preprocessing deals with class distribution and small disjuncts and feature selection somehow reduces the degree of overlapping. In a more general way, the idea behind this approach tries to overcome different sources of data complexity such as the class overlap, irrelevant and redundant features, noisy samples, class imbalance, low ratios of the sample size to dimensionality and so on, using different approaches used to solve each complexity.

5.4. Noisy data

Noisy data is known to affect the way any data mining system behaves [20, 109, 151]. Focusing on the scenario of imbalanced data, the presence of noise has a greater impact on the minority classes than on usual cases [135]; since the positive class has fewer examples to begin with, it will take fewer "noisy" examples to impact the learned subconcept. This issue is depicted in Figure 14, in which we can observe the decision boundaries obtained with SMOTE+C4.5 in the Subclus problem without noisy data (Figure 14.a) and how the frontiers between the classes are wrongly generated by introducing a 20% gaussian noise (Figure 14.b).



(a) Original problem and decision functions

(b) Noisy instances and new undesirable decision functions

Figure 14: Example of the effect of noise in imbalanced datasets for SMOTE+C4.5 in the Subclus dataset

According to [135], these "noise-areas" can be somehow viewed as "small disjuncts" and in order to avoid the erroneous generation of discrimination

ACCEPTED MANUSCRIPT

functions for these examples, some overfitting management techniques must be employed, such as pruning. However, the handicap of this methodology is that some correct minority classes will be ignored and, in this manner, the bias of the learner should be tuned-up in order to be able to provide a good global behavior for both classes of the problem.

For example, Batuwita and Palade developed the FSVM-CIL algorithm [13], a synergy between SVMs and fuzzy logic aimed to reflect the within-class importance of different training examples in order to suppress the effect of outliers and noise. The idea is to assign different fuzzy membership values to positive and negative examples and to incorporate this information in the SVM learning algorithm, aimed to reduce the effect of outliers and noise when finding the separating hyperplane.

In [111] we may find an empirical study on the effect of class imbalance and class noise on different classification algorithms and data sampling techniques. From this study, the authors extracted three important lessons on the topic:

1. Classification algorithms are more sensitive to noise than imbalance. However, as imbalance increases in severity, it plays a larger role in the performance of classifiers and sampling techniques.
2. Regarding the preprocessing mechanisms, simple undersampling techniques such as random undersampling and ENN performed the best overall, at all levels of noise and imbalance. Peculiarly, as the level of imbalance is increased, ENN proves to be more robust in the presence of noise. Additionally, OSS consistently proves itself to be relatively unaffected by an increase in the noise level. Other techniques such as random oversampling, SMOTE or Borderline-SMOTE obtain good results on average, but do not show the same behavior as undersampling.
3. Finally, the most robust classifiers tested over imbalanced and noisy data are bayesian classifiers and SVMs, performing better on average than rule induction algorithms or instance based learning. Furthermore, whereas most algorithms only experience small changes in AUC when imbalance was increased, the performance of Radial Basis Functions is significantly hindered when the imbalance ratio increases. For rule learning algorithms, the presence of noise degrades the performance more quickly than in other algorithms.

Additionally, in [75], the authors presented a similar study on the significance of noise and imbalance data using bagging and boosting techniques. Their results show the goodness of the bagging approach without replacement, and they recommend the use of noise reduction techniques prior to the application of boosting procedures.

As a final remark, we show a brief experimental study on the effect of noise over a specific imbalanced problem such as the Subclus dataset [97]. Table 22 includes the results for C4.5 with no preprocessing (None) and four different approaches, namely random undersampling, SMOTE [27], SMOTE+ENN [9] and SPIDER2 [97], a method designed for addressing noise and borderline examples, which will be detailed in the next section.

ACCEPTED MANUSCRIPT

This table is divided into two parts, the leftmost columns show the results with the original data and the columns in the right side show the results when adding a 20% of gaussian noise to the data. From this table we may conclude that in all cases the presence of noise degrades the performance of the classifier especially on the positive instances (TP_{rate}). Regarding the preprocessing approaches, the best behavior is obtained by SMOTE+ENN and SPIDER2, both of which include a cleaning mechanism to alleviate the problem of noisy data, whereas the latter also oversample the borderline minority examples.

Table 22: Performance obtained by C4.5 in the Subclus dataset with and without noisy instances

Dataset	Original Data			20% of Gaussian Noise		
	TP_{rate}	TN_{rate}	AUC	TP_{rate}	TN_{rate}	AUC
None	1.000	.9029	.9514	.0000	1.000	.5000
RandomUnderSampling	1.000	.7800	.8900	.9700	.7400	.8550
SMOTE	.9614	.9529	.9571	.8914	.8800	.8857
SMOTE+ENN	.9676	.9623	.9649	.9625	.9573	.9599
SPIDER2	1.000	1.000	1.000	.9480	.9033	.9256

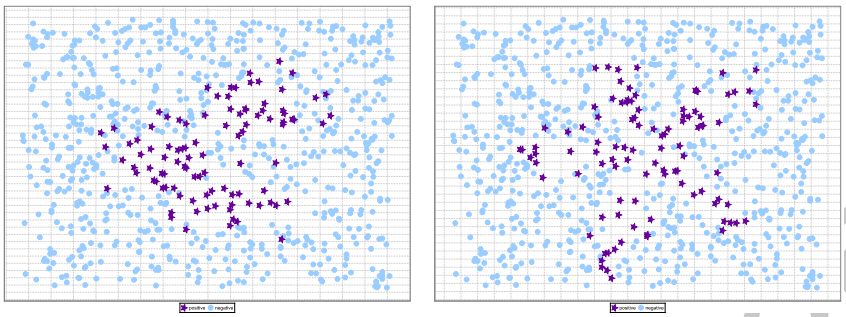
5.5. Borderline examples

Inspired by [76], we may distinguish between safe, noisy and borderline examples. Safe examples are placed in relatively homogeneous areas with respect to the class label. By noisy examples we understand individuals from one class occurring in safe areas of the other class, as introduced in the previous section. Finally, *Borderline examples* are located in the area surrounding class boundaries, where the minority and majority classes overlap. Figure 15 represents two examples given by [97], named *"Paw"* and *"Clover"*, respectively. In the former, the minority class is decomposed into 3 elliptic subregions, where two of them are located close to each other, and the remaining smaller sub-region is separated (upper right cluster). The latter also represents a non-linear setting, where the minority class resembles a flower with elliptic petals, which makes difficult to determine the boundaries examples in order to carry out a correct discrimination of the classes.

The problem of noisy data and the management of borderline examples are closely related, and most of the cleaning techniques briefly introduced in Section 3.1 can be used, or are the basis for detecting and emphasizing these borderline instances and, what is most important, to distinguish them from noisy instances that can degrade the overall classification. In brief, the better the definition of the borderline areas the more precise the discrimination between the positive and negative classes will be [39].

The family of SPIDER methods were proposed in [115] to ease the problem of the improvement of sensitivity at the cost of specificity that appears in the standard cleaning techniques. The SPIDER techniques works by combining a cleaning step of the majority examples with a local oversampling of the borderline minority examples [97, 115, 116].

ACCEPTED MANUSCRIPT



(a) Paw dataset (b) Clover dataset

Figure 15: Example of data with difficult borderline examples

We may also find other related techniques such as the Borderline-SMOTE [63], which seeks to oversample the minority class instances in the borderline areas, by defining a set of "Danger" examples, i.e. those which are most likely to be misclassified since they appear in the borderline areas, from which SMOTE generates synthetic minority samples in the neighborhood of the boundaries.

Other approaches such as Safe-Level-SMOTE [21] and ADASYN [65] work in a similar way. The former is based on the premise that previous approaches, such as SMOTE and Borderline-SMOTE, may generate synthetic instances in unsuitable locations, such as overlapping regions and noise regions; therefore, the authors compute a "safe-level" value for each positive instance before generating synthetic instances and generate them closer to the largest safe level. On the other hand, the key idea of the ADASYN algorithm is to use a density distribution as a criterion to automatically decide the number of synthetic samples that need to be generated for each minority example, by adaptively changing the weights of different minority examples to compensate the skewed distributions.

In [87], the authors use a hierarchical fuzzy rule learning approach, which defines a higher granularity for those problem subspaces in the borderline areas. The results have shown to be very competitive for highly imbalanced datasets in which this problem is accentuated.

Finally, in [97], the authors presented a series of experiments in which it is shown that the degradation in performance of a classifier is strongly affected by the number of borderline examples. They showed that focused resampling mechanisms (such as the Neighborhood Cleaning Rule [79] or SPIDER2 [97]) work well when the number of borderline examples is large enough whereas, on the contrary case, oversampling methods allow the improvement of the precision for the minority class.

The behavior of the SPIDER2 approach is shown in Table 15 for both the Paw and Clover problems. There are 10 different problems for each one of these datasets, depending on the number of examples and IR (600-5 or 800-7), and the "disturbance ratio" [97], defined as the ratio of borderline examples from

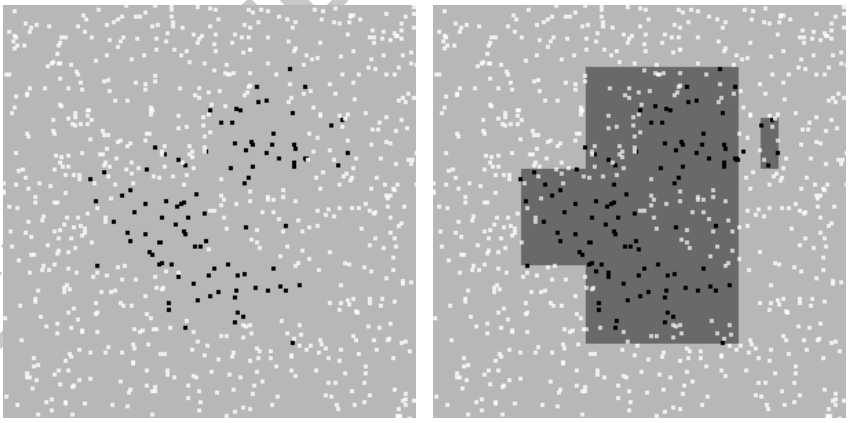
ACCEPTED MANUSCRIPT

the minority class subregions (0 to 70%). From these results we must stress the goodness of the SPIDER2 preprocessing step especially for those problems with a high disturbance ratio, which are harder to solve.

Table 23: AUC results in training and testing for the Clover and Paw problems with C4.5 (Original data and data preprocessed with SPIDER2)

		600 examples - IR 5				800 examples - IR 7			
		None		SPIDER2		None		SPIDER2	
Dataset	Disturbance	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}	AUC_{Tr}	AUC_{Tst}
Paw	0	.9568	.9100	.9418	.9180	.7095	.6829	.9645	.9457
	30	.7298	.7000	.9150	.8260	.6091	.5671	.9016	.8207
	50	.7252	.6790	.9055	.8580	.5000	.5000	.9114	.8400
	60	.5640	.5410	.9073	.8150	.5477	.5300	.8954	.7829
	70	.6250	.5770	.8855	.8350	.5000	.5000	.8846	.8164
	Average	.7202	.6814	.9110	.8504	.5732	.5560	.9115	.8411
Clover	0	.7853	.7050	.7950	.7410	.7607	.7071	.8029	.7864
	30	.6153	.5430	.9035	.8290	.5546	.5321	.8948	.7979
	50	.5430	.5160	.8980	.8070	.5000	.5000	.8823	.7907
	60	.5662	.5650	.8798	.8100	.5000	.5000	.8848	.8014
	70	.5000	.5000	.8788	.7690	.5250	.5157	.8787	.7557
	Average	.6020	.5658	.8710	.7912	.5681	.5510	.8687	.7864

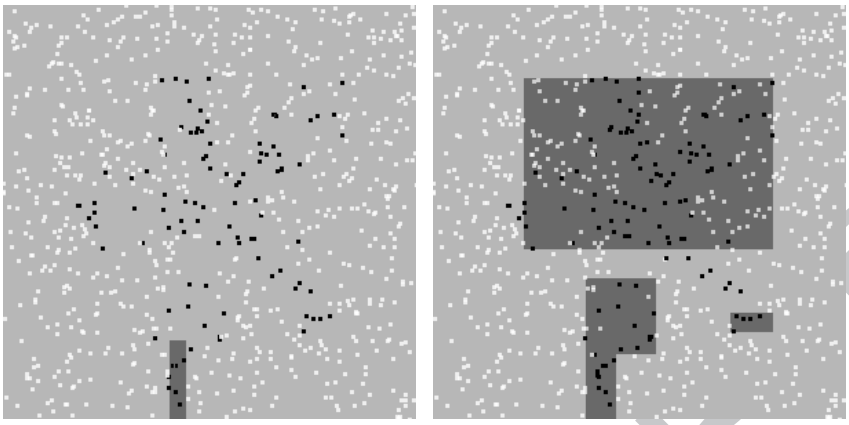
Additionally, and as a visual example of the behavior of this kind of methods, we show in Figures 16 and 17 the classification regions detected with C4.5 for the Paw and Clover problems using the original data and applying the SPIDER2 method. From these results we may conclude that the use of a methodology for stressing the borderline areas is very beneficial for correctly identifying the minority class instances.



(a) Original data (b) Data preprocessed with SPIDER2

Figure 16: Boundaries detected by C4.5 in the Paw problem (800 examples, IR 7 and disturbance ratio of 30))

ACCEPTED MANUSCRIPT



(a) Original data (b) Data preprocessed with SPIDER2

Figure 17: Boundaries detected by C4.5 in the Clover problem (800 examples, IR 7 and disturbance ratio of 30)

5.6. Dataset shift

The problem of dataset shift [2, 23, 114] is defined as the case where training and test data follow different distributions. This is a common problem that can affect all kind of classification problems, and it often appears due to sample selection bias issues. A mild degree of dataset shift is present in most real-world problems, but general classifiers are often capable of handling it without a severe performance loss.

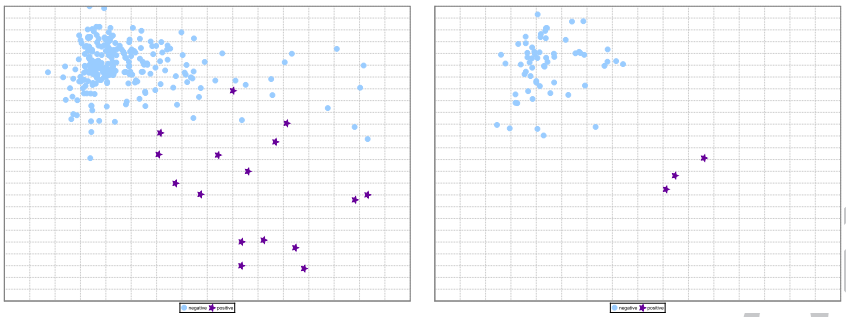
However, the dataset shift issue is specially relevant when dealing with imbalanced classification, because in highly imbalanced domains, the minority class is particularly sensitive to singular classification errors, due to the typically low number of examples it presents [94]. In the most extreme cases, a single misclassified example of the minority class can create a significant drop in performance.

For clarity, Figures 18 and 19 present two examples of the influence of the dataset shift in imbalanced classification. In the first case (Figure 18), it is easy to see a separation between classes in the training set that carries over perfectly to the test set. However, in the second case (Figure 19), it must be noted how some minority class examples in the test set are at the bottom and rightmost areas while they are localized in other areas in the training set, leading to a gap between the training and testing performance. These problems are represented in a two-dimensional space by means of a linear transformation of the inputs variables, following the technique given in [94].

Since the dataset shift is a highly relevant issue in imbalanced classification, it is easy to see why it would be an interesting perspective to focus on in future research regarding this topic. There are two different potential approaches in the study of the dataset shift in imbalanced domains:

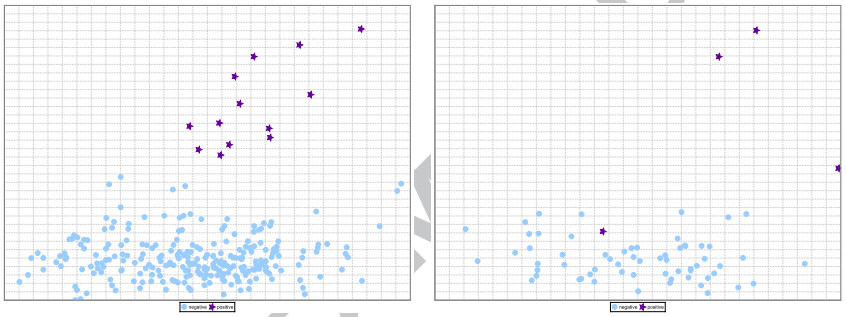
1. The first one focuses on intrinsic dataset shift, that is, the data of interest

ACCEPTED MANUSCRIPT



(a) Training data. AUC = .9043 (b) Test data. AUC = 1.000

Figure 18: Example of good behavior (no dataset shift) in imbalanced domains: ecoli4 dataset, 5th partition



(a) Training data. AUC = 1.000 (b) Test data. AUC = .8750

Figure 19: Example of bad behavior caused by dataset shift in imbalanced domains: ecoli4 dataset, 1st partition

includes some degree of shift that is producing a relevant drop in performance. In this case, we may develop techniques to discover and measure the presence of dataset shift [32, 33, 144], but adapting them to focus on the minority class. Furthermore, we may design algorithms that are capable of working under dataset shift conditions, either by means of preprocessing techniques [95] or with ad hoc algorithms [1, 16, 60]. In both cases, we are not aware of any proposals in the literature that focus on the problem of imbalanced classification in the presence of dataset shift.

2. The second approach in terms of dataset shift in imbalanced classification is related to induced dataset shift. Most current state of the art research is validated through stratified cross-validation techniques, which are another potential source of shift in the learning process. A more suitable validation technique needs to be developed in order to avoid introducing dataset shift issues artificially.

ACCEPTED MANUSCRIPT

6. Concluding Remarks

In this paper, we have reviewed the topic of classification with imbalanced datasets, and focused on two main issues: (1) to present the main approaches for dealing with this problem, namely, preprocessing of instances, cost-sensitive learning and ensemble techniques, and (2) to develop a thorough discussion on the effect of data intrinsic characteristics in learning from imbalanced datasets.

Mainly, we have pointed out that the imbalanced ratio by itself does not have the most significant effect on the classifiers' performance, but that there are other issues that must be taken into account. We have presented six different cases, which, in conjunction with a skewed data distribution, impose a strong handicap for achieving a high classification performance for both classes of the problem, i.e., the presence of small disjuncts, the lack of density or small sample size, the class overlapping, the noisy data, the correct management of borderline examples, and the dataset shift.

For each one of the mentioned issues, we have described the main features that makes learning algorithms to be wrongly biased and we have presented several solutions proposed along the years in the specialized literature. This review paper emphasizes that there is a current need to study the aforementioned intrinsic characteristics of the data, so that future research on classification with imbalanced data should focus on detecting and measuring the most significant data properties, in order to be able to define good solutions as well as alternatives to overcome the problems.

Acknowledgment

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

References

[1] Alaiz-Rodríguez, R., Guerrero-Curieses, A., Cid-Sueiro, J., 2009. Improving classification under changes in class and within-class distributions. In: Proceedings of the 10th International Work-Conference on Artificial Neural Networks (IWANN '09). Springer-Verlag, Berlin, Heidelberg, pp. 122–130.

[2] Alaiz-Rodríguez, R., Japkowicz, N., 2008. Assessing the impact of changing environments on classifier performance. In: Proceedings of the 21st Canadian conference on Advances in artificial intelligence (CCAI'08). Springer-Verlag, Berlin, Heidelberg, pp. 13–24.

ACCEPTED MANUSCRIPT

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

[4] Alcalá;-Fdez, J., Sánchez, L., García, S., del Jesus, M. J., Ventura, S., Garrell, J. M., Otero, J., Romero, C., Bacardit, J., Rivas, V. M., Fernández, J. C., Herrera, F., 2009. KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Computing 13, 307–318.

[5] Anand, A., Pugalenthi, G., Fogel, G. B., Suganthan, P. N., 2010. An approach for classification of highly imbalanced data using weighting and undersampling. Amino Acids 39 (5), 1385–1391.

[6] Baeza-Yates, R., Ribeiro-Neto, B., 1999. Modern Information Retrieval. Addison Wesley.

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

[8] Barandela, R., Valdovinos, R. M., Sánchez, J. S., 2003. New applications of ensembles of classifiers. Pattern Analysis Applications 6 (3), 245–256.

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

[10] Batuwita, R., Palade, V., 2009. AGm: A new performance measure for class imbalance learning. application to bioinformatics problems. In: Proceedings of the 8th International Conference on Machine Learning and Applications (ICMLA 2009). pp. 545–550.

[11] Batuwita, R., Palade, V., 2009. microPred: Effective classification of pre-miRNAs for human miRNA gene prediction. Bioinformatics 25 (8), 989–995.

[12] Batuwita, R., Palade, V., 2010. Efficient resampling methods for training support vector machines with imbalanced datasets. In: Proceedings of the 2010 International Joint Conference on Neural Networks (IJCNN).

[13] Batuwita, R., Palade, V., 2010. FSVM-CIL: Fuzzy support vector machines for class imbalance learning. IEEE Transactions on Fuzzy Systems 18 (3), 558–571.

[14] Batuwita, R., Palade, V., 2012. Adjusted Geometric-mean: A Novel Performance Measure for Imbalanced Bioinformatics Datasets Learning. Journal of Bioinformatics and Computational Biology 10 (4).

[15] Batuwita, R., Palade, V., 2013. Class imbalance learning methods for support vector machines. In: He, H., Ma, Y. (Eds.), Imbalanced Learning: Foundations, Algorithms, and Applications. Vol. in press. Wiley.

ACCEPTED MANUSCRIPT

[16] Bickel, S., Brückner, M., Scheffer, T., 2009. Discriminative learning under covariate shift. Journal of Machine Learning Research 10, 2137–2155.

[17] Błaszczyński, J., Deckert, M., Stefanowski, J., Wilk, S., 2010. Integrating selective pre-processing of imbalanced data with ivotes ensemble. In: Szczuka, M., Kryszkiewicz, M., Ramanna, S., Jensen, R., Hu, Q. (Eds.), Rough Sets and Current Trends in Computing. Vol. 6086 of LNSC. Springer Berlin / Heidelberg, pp. 148–157.

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

[19] Bradley, A. P., 1997. The use of the area under the roc curve in the evaluation of machine learning algorithms. Pattern Recognition 30 (7), 1145–1159.

[20] Brodley, C. E., Friedl, M. A., 1999. Identifying mislabeled training data. Journal of Artificial Intelligence Research 11, 131–167.

[21] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2009. Safe–level–SMOTE: Safe–level–synthetic minority over–sampling TEchnique for handling the class imbalanced problem. In: Proceedings of the 13th Pacific–Asia Conference on Advances in Knowledge Discovery and Data Mining PAKDD'09. pp. 475–482.

[22] Bunkhumpornpat, C., Sinapiromsaran, K., Lursinsap, C., 2012. DB-SMOTE: Density-based synthetic minority over-sampling TEchnique. Applied Intelligence 36 (3), 664–684.

[23] Candela, J. Q., Sugiyama, M., Schwaighofer, A., Lawrence, N. D., 2009. Dataset Shift in Machine Learning. The MIT Press.

[24] Carvalho, D. R., Freitas, A. A., 2004. A hybrid decision tree/genetic algorithm method for data mining. Information Sciences 163 (1–3), 13–35.

[25] Chan, P. K., Stolfo, S. J., 1998. Toward scalable learning with non-uniform class and cost distributions: A case study in credit card fraud detection. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining (KDD'98). pp. 164–168.

[26] Chang, E. Y., Li, B., Wu, G., Goh, K., 2003. Statistical learning for effective visual information retrieval. In: Proceedings of the 2003 International Conference on Image Processing (ICIP'03). Vol. 3. pp. 609–612.

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

ACCEPTED MANUSCRIPT

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

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

[30] Chawla, N. V., Lazarevic, A., Hall, L. O., Bowyer, K. W., 2003. SMOTEBoost: Improving prediction of the minority class in boosting. In: Proceedings of 7th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD'03). pp. 107–119.

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

[32] Cieslak, D. A., Chawla, N. V., 2008. Analyzing pets on imbalanced datasets when training and testing class distributions differ. In: Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD08). Osaka, Japan, pp. 519–526.

[33] Cieslak, D. A., Chawla, N. V., 2009. A framework for monitoring classifiers' performance: when and why failure occurs? Knowledge and Information Systems 18 (1), 83–108.

[34] Cohen, G., Hilario, M., Sax, H., Hugonnet, S., Geissbuhler, A., 2006. Learning from imbalanced data in surveillance of nosocomial infection. Artificial Intelligence in Medicine 37, 7–18.

[35] Cortes, C., Vapnik, V., 1995. Support vector networks. Machine Learning 20, 273–297.

[36] Davis, J., Goadrich, M., 2006. The relationship between precisionrecall and ROC curves. In: Proceedings of the 23th International Conference on Machine Learning (ICML'06). ACM, pp. 233–240.

[37] Denil, M., Trappenberg, T., 2010. Overlap versus imbalance. In: Proceedings of the 23rd Canadian conference on Advances in artificial intelligence (CCAI'10). Vol. 6085 of Lecture Notes on Artificial Intelligence. pp. 220–231.

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

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

ACCEPTED MANUSCRIPT

[40] Drummond, C., Holte, R. C., 2006. Cost curves: An improved method for visualizing classifier performance. Machine Learning 65 (1), 95–130.

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

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

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

[44] Fan, W., Stolfo, S. J., Zhang, J., Chan, P. K., 1999. Adacost: Misclassification cost-sensitive boosting. In: Proceedings of the 16th International Conference on Machine Learning (ICML'96). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 97–105.

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

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

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

[48] Fernández, A., López, V., Galar, M., del Jesus, M. J., Herrera, F., 2013. Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches. Knowledge-Based Systems 42, 97–110.

[49] Friedman, J. H., Kohavi, R., Yun, Y., 1996. Lazy decision trees. In: Proceedings of the AAAI/IAAI, Vol. 1. pp. 717–724.

[50] Galar, M., Fernández, A., Barrenechea, E., Bustince, H., Herrera, F., 2012. A review on ensembles for class imbalance problem: Bagging, boosting and hybrid based approaches. IEEE Transactions on Systems, Man, and Cybernetics–part C: Applications and Reviews 42 (4), 463–484.

[51] García, S., Derrac, J., Triguero, I., Carmona, C. J., Herrera, F., 2012. Evolutionary-based selection of generalized instances for imbalanced classification. Knowledge Based Systems 25 (1), 3–12.

ACCEPTED MANUSCRIPT

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

[53] García, S., Herrera, F., 2009. Evolutionary under-sampling for classification with imbalanced data sets: Proposals and taxonomy. Evolutionary Computation 17 (3), 275–306.

[54] García, V., Mollineda, R. A., Sánchez, J. S., 2008. A new performance evaluation method for two-class imbalanced problems. In: Proceedings of the Structural and Syntactic Pattern Recognition (SSPR'08) and Statistical Techniques in Pattern Recognition (SPR'08). Vol. 5342 of Lecture Notes on Computer Science. pp. 917–925.

[55] García, V., Mollineda, R. A., Sánchez, J. S., 2008. On the k–NN performance in a challenging scenario of imbalance and overlapping. Pattern Analysis Applications 11 (3–4), 269–280.

[56] García, V., Mollineda, R. A., Sánchez, J. S., 2010. Theoretical analysis of a performance measure for imbalanced data. In: 20th International Conference on Pattern Recognition (ICPR'10). pp. 617–620.

[57] García, V., Mollineda, R. A., Sánchez, J. S., 2012. Classifier performance assessment in two-class imbalanced problems. Internal Communication.

[58] García, V., Sánchez, J. S., Mollineda, R. A., 2012. On the effectiveness of preprocessing methods when dealing with different levels of class imbalance. Knowledge Based Systems 25 (1), 13–21.

[59] García-Pedrajas, N., Pérez-Rodríguez, J., García-Pedrajas, M., Ortiz-Boyer, D., Fyfe, C., 2012. Class imbalance methods for translation initiation site recognition in DNA sequences. Knowledge Based Systems 25 (1), 22–34.

[60] Globerson, A., Teo, C. H., Smola, A., Roweis, S., 2009. An Adversarial View of Covariate Shift and a Minimax Approach. In: Quiñonero Candela, J., Sugiyama, M., Schwaighofer, A., Lawrence, N. D. (Eds.), Dataset Shift in Machine Learning. The MIT Press, pp. 179–198.

[61] Guo, H., Viktor, H. L., 2004. Learning from imbalanced data sets with boosting and data generation: the DataBoost-IM approach. SIGKDD Explorations Newsletter 6, 30–39.

[62] Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I. H., 2009. The WEKA data mining software: An update. SIGKDD Explorations 11 (1), 10–18.

ACCEPTED MANUSCRIPT

[63] Han, H., Wang, W. Y., Mao, B. H., 2005. Borderline–SMOTE: A new over–sampling method in imbalanced data sets learning. In: Proceedings of the 2005 International Conference on Intelligent Computing (ICIC'05). Vol. 3644 of Lecture Notes in Computer Science. pp. 878–887.

[64] Hart, P. E., 1968. The condensed nearest neighbor rule. IEEE Transactions on Information Theory 14, 515–516.

[65] He, H., Bai, Y., Garcia, E. A., Li, S., 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In: Proceedings of the 2008 IEEE International Joint Conference Neural Networks (IJCNN'08). pp. 1322–1328.

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

[67] Hido, S., Kashima, H., Takahashi, Y., 2009. Roughly balanced bagging for imbalanced data. Statistical Analysis and Data Mining 2, 412–426.

[68] Holte, R. C., Acker, L., Porter, B. W., 1989. Concept learning and the problem of small disjuncts. In: Proceedings of the International Joint Conferences on Artificial Intelligence IJCAI'89. pp. 813–818.

[69] Hu, S., Liang, Y., Ma, L., He, Y., 2009. MSMOTE: Improving classification performance when training data is imbalanced. In: Proceedings of the 2nd International Workshop on Computer Science and Engineering (WCSE'09). Vol. 2. pp. 13–17.

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

[71] Japkowicz, N., 2001. Concept-learning in the presence of between-class and within-class imbalances. In: Stroulia, E., Matwin, S. (Eds.), Proceedings of the 14th Canadian conference on Advances in artificial intelligence (CCAI'08). Vol. 2056 of Lecture Notes in Computer Science. Springer, pp. 67–77.

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

[73] Jo, T., Japkowicz, N., 2004. Class imbalances versus small disjuncts. ACM SIGKDD Explorations Newsletter 6 (1), 40–49.

[74] Joshi, M. V., Kumar, V., Agarwal, R. C., 2001. Evaluating boosting algorithms to classify rare classes: Comparison and improvements. In: Proceedings of the 2001 IEEE International Conference on Data Mining (ICDM'01). IEEE Computer Society, Washington, DC, USA, pp. 257–264.

ACCEPTED MANUSCRIPT

[75] Khoshgoftaar, T. M., Van Hulse, J., Napolitano, A., 2011. Comparing boosting and bagging techniques with noisy and imbalanced data. IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans 41 (3), 552–568.

[76] Kubat, M., Matwin, S., 1997. Addressing the curse of imbalanced training sets: one–sided selection. In: Proceedings of the 14th International Conference on Machine Learning (ICML'97). pp. 179–186.

[77] Kuncheva, L. I., Rodrguez, J. J., 2013. A weighted voting framework for classifiers ensembles. Knowledge and Information Systems In press, doi: 10.1007/s10115-012-0586-6.

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

[79] Laurikkala, J., 2001. Improving identification of difficult small classes by balancing class distribution. In: Proceedings of the 8th Conference on AI in Medicine in Europe: Artificial Intelligence Medicine (AIME'01). pp. 63–66.

[80] Li, C., 2007. Classifying imbalanced data using a bagging ensemble variation (BEV). In: Proceedings of the 45th annual southeast regional conference. ACM-SE 45. ACM, New York, NY, USA, pp. 203–208.

[81] Lin, M., Tang, K., Yao, X., 2013. Dynamic sampling approach to training neural networks for multiclass imbalance classification. IEEE Transactions on Neural Networks and Learning Systems 24 (4), 647–660.

[82] Lin, W., Chen, J. J., 2013. Class-imbalanced classifiers for high-dimensional data. Briefings in Bioinformatics 14 (1), 13–26.

[83] Ling, C. X., Li, C., 1998. Data mining for direct marketing: Problems and solutions. In: Proceedings of the 4th International Conference on Knownledge Discovery and Data Mining (KDD'98). pp. 73–79.

[84] Ling, C. X., Yang, Q., Wang, J., Zhang, S., 2004. Decision trees with minimal costs. In: Brodley, C. E. (Ed.), Proceedings of the 21st International Conference on Machine Learning (ICML'04). Vol. 69 of ACM International Conference Proceeding Series. ACM, pp. 69–77.

[85] Liu, X.-Y., Wu, J., Zhou, Z.-H., 2009. Exploratory undersampling for class-imbalance learning. IEEE Transactions on System, Man and Cybernetics B 39 (2), 539–550.

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

ACCEPTED MANUSCRIPT

[87] López, V., Fernández, A., del Jesus, M. J., Herrera, F., 2013. A hierarchical genetic fuzzy system based on genetic programming for addressing classification with highly imbalanced and borderline data-sets. Knowledge-Based Systems 38, 85–104.

[88] López, V., Fernández, A., Moreno-Torres, J. G., Herrera, F., 2012. Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. open problems on intrinsic data characteristics. Expert Systems with Applications 39 (7), 6585–6608.

[89] Luengo, J., Fernández, A., García, S., Herrera, F., 2011. Addressing data complexity for imbalanced data sets: analysis of SMOTE–based oversampling and evolutionary undersampling. Soft Computing 15 (10), 1909–1936.

[90] Martín-Félez, R., Mollineda, R. A., 2010. On the suitability of combining feature selection and resampling to manage data complexity. In: Proceedings of the Conferencia de la Asociacin Espaola de Inteligencia Artificial (CAEPIA'09). Vol. 5988 of Lecture Notes on Artificial Intelligence. pp. 141–150.

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

[92] McLachlan, G. J., 2004. Discriminant Analysis and Statistical Pattern Recognition. John Wiley and Sons.

[93] Mena, L., González, J. A., 2009. Symbolic one-class learning from imbalanced datasets: Application in medical diagnosis. International Journal on Artificial Intelligence Tools 18 (2), 273–309.

[94] Moreno-Torres, J. G., Herrera, F., 2010. A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction. In: Proceedings of the 10th International Conference on Intelligent Systems Design and Applications(ISDA'10). pp. 501–506.

[95] Moreno-Torres, J. G., Llorà, X., Goldberg, D. E., Bhargava, R., 2013. Repairing Fractures between Data using Genetic Programming-based Feature Extraction: A Case Study in Cancer Diagnosis. Information Sciences 222, 805–823.

[96] Moreno-Torres, J. G., Raeder, T., Aláiz-Rodríguez, R., Chawla, N. V., Herrera, F., 2012. A unifying view on dataset shift in classification. Pattern Recognition 45 (1), 521–530.

ACCEPTED MANUSCRIPT

[97] Napierala, K., Stefanowski, J., Wilk, S., 2010. Learning from imbalanced data in presence of noisy and borderline examples. In: Proceedings of the 7th International Conference on Rough Sets and Current Trends in Computing (RSCTC'10). Vol. 6086 of Lecture Notes on Artificial Intelligence. pp. 158–167.

[98] Orriols-Puig, A., Bernadó-Mansilla, E., 2009. Evolutionary rule–based systems for imbalanced datasets. Soft Computing 13 (3), 213–225.

[99] Orriols-Puig, A., Bernadó-Mansilla, E., Goldberg, D. E., Sastry, K., Lanzi, P. L., 2009. Facetwise analysis of XCS for problems with class imbalances. IEEE Transactions on Evolutionary Computation 13, 260–283.

[100] Platt, J., 1998. Fast training of support vector machines using sequential minimal optimization. In: Advances in Kernel Methods - Support Vector Learning. MIT Press, Cambridge, MA, pp. 42–65.

[101] Polikar, R., 2006. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine 6 (3), 21–45.

[102] Prati, R. C., Batista, G. E. A. P. A., 2004. Class imbalances versus class overlapping: an analysis of a learning system behavior. In: Proceedings of the 2004 Mexican International Conference on Artificial Intelligence (MICAI'04). pp. 312–321.

[103] Prati, R. C., Batista, G. E. A. P. A., Monard, M. C., 2011. A survey on graphical methods for classification predictive performance evaluation. IEEE Transactions on Knowledge and Data Engineering 23 (11), 1601–1618.

[104] Quinlan, J. R., 1993. C4.5: Programs for Machine Learning. Morgan Kauffman.

[105] Raeder, T., Forman, G., Chawla, N. V., 2012. Learning from imbalanced data: Evaluation matters. In: Holmes, D. E., Jain, L. C. (Eds.), Data Mining: Found. and Intell. Paradigms. Vol. ISRL 23. Springer-Verlag, pp. 315–331.

[106] Raudys, S. J., Jain, A. K., 1991. Small sample size effects in statistical pattern recognition: Recommendations for practitioners. IEEE Transactions on Pattern Analysis and Machine Intelligence 13 (3), 252–264.

[107] Riddle, P., Segal, R., Etzioni, O., 1994. Representation design and brute–force induction in a boeing manufacturing domain. Applied Artificial Intelligence 8, 125–147.

[108] Rokach, L., 2010. Ensemble-based classifiers. Artificial Intelligence Review 33 (1), 1–39.

ACCEPTED MANUSCRIPT

[109] Sáez, J. A., Luengo, J., Herrera, F., 2010. A first study on the noise impact in classes for fuzzy rule based classification systems. In: Proceedings of the 2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering (ISKE'10). IEEE Press, pp. 153–158.

[110] Schapire, R. E., 1999. A brief introduction to boosting. In: Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'99). pp. 1401–1406.

[111] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., Folleco, A., 2012. An empirical study of the classification performance of learners on imbalanced and noisy software quality data. Information Sciences In press, doi: 10.1016/j.ins.2010.12.016.

[112] Seiffert, C., Khoshgoftaar, T. M., Van Hulse, J., Napolitano, A., 2010. RUSBoost: A hybrid approach to alleviating class imbalance. IEEE Transactions on System, Man and Cybernetics A 40 (1), 185–197.

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

[114] Shimodaira, H., 2000. Improving predictive inference under Covariate Shift by Weighting the Log-likelihood Function. Journal of Statistical Planning and Inference 90 (2), 227–244.

[115] Stefanowski, J., Wilk, S., 2007. Improving rule based classifiers induced by MODLEM by selective pre-processing of imbalanced data. In: Proceedings of the RSKD Workshop at ECML/PKDD'07. pp. 54–65.

[116] Stefanowski, J., Wilk, S., 2008. Selective pre-processing of imbalanced data for improving classification performance. In: Proceedings of the 10th International Conference on Data Warehousing and Knowledge Discovery (DaWaK08). pp. 283–292.

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

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

[119] Tang, Y., Zhang, Y.-Q., Chawla, N. V., Kresser, S., 2009. SVMs modeling for highly imbalanced classification. IEEE Transactions on Systems, Man and Cybernetics, Part B 39 (1), 281–288.

[120] Tao, D., Tang, X., Li, X., Wu, X., 2006. Asymmetric bagging and random subspace for support vector machines-based relevance feedback in image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (7), 1088–1099.

ACCEPTED MANUSCRIPT

[121] Ting, K. M., 1994. The problem of small disjuncts: its remedy in decision trees. In: Proceedings of the 10th Canadian Conference on Artificial Intelligence (CCAI'94). pp. 91–97.

[122] Ting, K. M., 2000. A comparative study of cost-sensitive boosting algorithms. In: Proceedings of the 17th International Conference on Machine Learning (ICML'00). Stanford, CA, USA, pp. 983–990.

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

[124] Tomek, I., 1976. Two modifications of CNN. IEEE Transactions on Systems Man and Communications 6, 769–772.

[125] Tsai, C.-H., Chang, L.-C., Chiang, H.-C., 2009. Forecasting of ozone episode days by cost–sensitive neural network methods. Science of the Total Environment 407 (6), 2124–2135.

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

[127] Van Hulse, J., Khoshgoftaar, T. M., Napolitano, A., 2009. An empirical comparison of repetitive undersampling techniques. In: Proceedings of the 2009 IEEE International Conference on Information Reuse Integration (IRI'09). pp. 29–34.

[128] Wang, B. X., Japkowicz, N., 2004. Imbalanced data set learning with synthetic samples. In: Proceedings of the IRIS Machine Learning Workshop.

[129] Wang, J., You, J., Li, Q., Xu, Y., 2012. Extract minimum positive and maximum negative features for imbalanced binary classification. Pattern Recognition 45 (3), 1136–1145.

[130] Wang, S., Yao, X., 2009. Diversity analysis on imbalanced data sets by using ensemble models. In: Proceedings of the 2009 IEEE Symposium on Computational Intelligence and Data Mining (CIDM'09). pp. 324–331.

[131] Wang, S., Yao, X., 2013. Relationships between diversity of classification ensembles and single-class performance measures. IEEE Transactions on Knowledge and Data Engineering 25 (1), 206–219.

[132] Wang, Z., Palade, V., 2011. Building interpretable fuzzy models for high dimensional data analysis in cancer diagnosis. BMC Genomics 12 ((S2):S5).

[133] Wasikowski, M., Chen, X.-W., 2010. Combating the small sample class imbalance problem using feature selection. IEEE Transactions on Knowledge and Data Engineering 22 (10), 1388–1400.

ACCEPTED MANUSCRIPT

[134] Weiss, G. M., 1999. Timeweaver: a genetic algorithm for identifying predictive patterns in sequences of events. In: Banzhaf, W., Daida, J., Eiben, A. E., Garzon, M. H., Honavar, V., Jakiela, M., Smith, R. E. (Eds.), Proceedings of the Genetic and Evolutionary Computation Conference (GECCO'99). Vol. 1. Morgan Kaufmann, Orlando, Florida, USA, pp. 718–725.

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

[136] Weiss, G. M., 2005. Mining with rare cases. In: Maimon, O., Rokach, L. (Eds.), The Data Mining and Knowledge Discovery Handbook. Springer, pp. 765–776.

[137] Weiss, G. M., 2010. The impact of small disjuncts on classifier learning. In: Stahlbock, R., Crone, S. F., Lessmann, S. (Eds.), Data Mining. Vol. 8 of Annals of Information Systems. Springer, pp. 193–226.

[138] Weiss, G. M., Provost, F. J., 2003. Learning when training data are costly: The effect of class distribution on tree induction. Journal of Artificial Intelligence Research 19, 315–354.

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

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

[141] Yan, R., Liu, Y., Jin, R., Hauptmann, A., 2003. On predicting rare classes with SVM ensembles in scene classification. In: Proceedings of the 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP'03). Vol. 3. pp. 21–24.

[142] Yang, P., Xu, L., Zhou, B. B., Zhang, Z., Zomaya, A. Y., 2009. A particle swarm based hybrid system for imbalanced medical data sampling. BMC Genomics 10 (SUPPL. 3), art. no. S34.

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

[144] Yang, Y., Wu, X., Zhu, X., 2008. Conceptual equivalence for contrast mining in classification learning. Data & Knowledge Engineering 67 (3), 413–429.

[145] Yen, S., Lee, Y., 2006. Under-sampling approaches for improving prediction of the minority class in an imbalanced dataset. In: Proceedings of the 2006 International Conference on Intelligent Computing (ICIC06). pp. 731–740.

ACCEPTED MANUSCRIPT

[146] Yoon, K., Kwek, S., 2005. An unsupervised learning approach to resolving the data imbalanced issue in supervised learning problems in functional genomics. In: Proceedings of the 5th International Conference on Hybrid Intelligent Systems (HIS'05). pp. 303–308.

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

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

[149] Zhang, J., Mani, I., 2003. KNN approach to unbalanced data distributions: A case study involving information extraction. In: Proceedings of the 20th International Conference on Machine Learning (ICML'03), Workshop Learning from Imbalanced Data Sets.

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

[151] Zhu, X., Wu, X., 2004. Class noise vs. attribute noise: A quantitative study. Artificial Intelligence Review 22 (3), 177–210.

[152] Zong, W., Huang, G.-B., Chen, Y., 2013. Weighted extreme learning machine for imbalance learning. Neurocomputing 101, 229–242.

ACCEPTED MANUSCRIPT