Information Sciences 257 (2014) 1–13

Contents lists available at ScienceDirect

Information Sciences

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

On the importance of the validation technique for classification with imbalanced datasets: Addressing covariate shift when data is skewed

Victoria López [a,*], Alberto Fernández [b], Francisco Herrera [a]

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

ARTICLE INFO

Article history:
Received 13 August 2012
Received in revised form 14 June 2013
Accepted 15 September 2013
Available online 21 September 2013

Keywords:
Classification
Imbalanced dataset
Covariate shift
Dataset shift
Validation technique
Partitioning

ABSTRACT

In the field of Data Mining, the estimation of the quality of the learned models is a key step in order to select the most appropriate tool for the problem to be solved. Traditionally, a *k*-fold validation technique has been carried out so that there is a certain degree of independency among the results for the different partitions. In this way, the highest average performance will be obtained by the most robust approach. However, applying a "random" division of the instances over the folds may result in a problem known as dataset shift, which consists in having a different data distribution between the training and test folds.

In classification with imbalanced datasets, in which the number of instances of one class is much lower than the other class, this problem is more severe. The misclassification of minority class instances due to an incorrect learning of the real boundaries caused by a not well fitted data distribution, truly affects the measures of performance in this scenario. Regarding this fact, we propose the use of a specific validation technique for the partitioning of the data, known as "Distribution optimally balanced stratified cross-validation" to avoid this harmful situation in the presence of imbalance. This methodology makes the decision of placing close-by samples on different folds, so that each partition will end up with enough representatives of every region.

We have selected a wide number of imbalanced datasets from KEEL dataset repository for our study, using several learning techniques from different paradigms, thus making the conclusions extracted to be independent of the underlying classifier. The analysis of the results has been carried out by means of the proper statistical study, which shows the goodness of this approach for dealing with imbalanced data.

1. Introduction

Standard learning algorithms are designed under the premise of a balanced class distribution. When dealing with skewed class distributions, the classification problem becomes more difficult, specifically for correctly identifying the minority concepts within the data [11]. This issue is known as the class imbalance problem [21,38], in which there is an under-represented class (positive) and a majority class (negative). This problem is present in many real-world classification tasks and has been considered as a challenge within the Data Mining community [48].

* Corresponding author. Tel.: +34 958 240598; fax: +34 958 243317.
E-mail addresses: vlopez@decsai.ugr.es (V. López), alberto.fernandez@ujaen.es (A. Fernández), herrera@decsai.ugr.es (F. Herrera).

0020-0255/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.09.038

In order to validate the performance of a classifier, both in standard and imbalanced classification, stratified cross-validation (SCV) is the most commonly employed method in the literature. It places an equal number of samples of each class on each partition to maintain class distributions similar in all partitions [9]. However, when this process is carried out in a random way, it may introduce a different data distribution between the training and test partitions, thus leading to inaccurate conclusions when learning a model from the training data. This issue is known as dataset shift [8], or more specifically covariate shift [30].

In the presence of imbalance, this problem is even more critic according to the metrics of performance applied in this scenario. Since misclassifications for the positive class instances severely hinder the average precision, we must try to avoid those errors in test which are due to a "random clustering" of the classes, i.e. generating outliers.

A more suitable validation technique needs to be employed in order to avoid introducing dataset shift issues artificially. In this paper, we suggest the use of a novel methodology called "Distribution optimally balanced SCV" (DOB–SCV) [31] when dealing with imbalanced datasets. This method attempts to minimize covariate shift by keeping data distribution as similar as possible between training and test folds by maximizing diversity on each fold and trying to keep all folds as resembling as possible to each other. The mechanism of this approach consists in selecting the k closest neighbours for a given instance and place them in different folds (with k being the number of total partitions), so that the data distribution between the training and test partitions remains as close as possible.

We must point out that neither SCV nor DOB–SCV can undoubtedly estimate the true classification error of a given model. In particular, there are several factors which may affect the output for unseen samples, and make some problems more difficult than others. Among others, we may stress uneven class distribution (as studied in this paper), the dimensionality of the problem and its relationship with the overlapping between the classes, and the presence of noise and/or outliers. However, we suggest that, by making the training and test partitions more similar between them, the use of DOB–SCV can guarantee a better average validation of the results. As pointed out previously, in this way we may avoid those classification errors which are due to dataset shift, especially those regarded to the minority class instances.

In order to evaluate the goodness and validity of the use of this new partitioning mechanism for imbalanced datasets, we develop a thorough empirical study by setting up an experimental framework which includes a set of sixty-six real-world problems from the KEEL dataset repository [3,4] (http://www.keel.es/dataset.php). We measure the performance of the classifiers based on its Area Under the Curve (AUC) metric [23] as suggested in imbalanced domains. Additionally, we study the significance of the results by the proper statistical tests as suggested in the literature [17,20]. Finally, we check the robustness of the DOB–SCV strategy using several well-known classifiers from different Machine Learning paradigms: decision trees [34], fuzzy rule based classification systems (FRBCS) [24], instance-based learning [1], and Support Vector Machines (SVMs) [12,15].

This study provides three significant contributions to the research community on classification with imbalanced data, namely:

1. We establish the motivation for the use of a new validation technique for avoiding dataset shift, which highly affects the performance in this scenario.
2. The goodness of this novel methodology is confirmed by means of a thorough experimental analysis. In this study, several algorithms from different paradigms were selected, showing better average performance estimates when using DOB–SCV.
3. Finally, we have concluded that the optimistic/pessimistic estimation of the performance also depends on the problem to be classified. In this way, the intrinsic data characteristics may have some degree of influence on the final results obtained by the classifier.

In order to carry out the study, this manuscript is organized as follows. First, Section 2 introduces the problem of imbalanced data. Next, Section 3 contains the main concepts that are developed in this work, i.e. the basis on validation techniques and the problem of covariate/dataset shift. Then, the experimental framework is presented in Section 4, whereas all the analysis of the results is shown along Section 5. Finally, Section 6 summarises and concludes the work.

2. Imbalanced datasets in classification

In this section, we will first introduce the problem of imbalanced datasets, describing its features and why is so difficult to learn in this classification scenario. Then, we will present how to address this problem, enumerating diverse approaches that can be applied to ease the discrimination of the minority (positive) and majority (negative) classes. Finally, we will discuss how to evaluate the performance of the results in this situation.

2.1. The problem of imbalanced datasets

The main property of this type of classification problem (in a binary context) is that the examples of one class outnumber the examples of the other one [11,38]. The minority classes are usually the most important concepts to be learnt, since they might be associated with exceptional and significant cases [42] or because the data acquisition of these examples is costly

[44]. Since most of the standard learning algorithms consider a balanced training set, this situation may cause the obtention of suboptimal classification models, i.e. a good coverage of the majority examples whereas the minority ones are misclassified more frequently [21,38].

Traditionally, the Imbalance Ratio (IR), i.e. the ratio between the majority and minority class examples [32], is the main hint to identify a set of problems which need to be addressed in a special way. Additionally, other data intrinsic characteristics that are related to this concept may include the overlapping between classes [26], lack of representative data [41], small disjuncts [33,43], dataset shift [29] and other issues which have interdependent effects with data distribution (imbalance).

The hitch here is that most learning algorithms aim to obtain a model with a high prediction accuracy and a good generalization capability. However, this inductive bias towards such a model poses a serious challenge to the classification of imbalanced data [38]. First, if the search process is guided by the standard accuracy rate, it benefits the covering of the majority examples; second, classification rules that predict the positive class are often highly specialized and thus their coverage is very low, hence they are discarded in favour of more general rules, i.e. those that predict the negative class. Furthermore, it is not easy to distinguish between noisy examples and positive class examples and they can be completely ignored by the classifier.

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

A large number of approaches have been proposed to deal with the class imbalance problem [28], which can be categorized in three groups:

1. Data level solutions: the objective consists in rebalancing the class distribution by sampling the data space to diminish the effect caused by class imbalance, acting as an external approach [6,10,39].
2. Algorithmic level solutions: these solutions try to adapt several classification algorithms to reinforce the learning towards the positive class. Therefore, they can be defined as internal approaches that create new algorithms or modify existing ones to take the class imbalance problem into consideration [5,49].
3. Cost-sensitive solutions: this type of solutions incorporate approaches at the data level, at the algorithmic level, or at both levels jointly, 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 [18,40,50].

The advantage of the data level solutions is that they are more versatile, since their use is independent of the classifier selected. Furthermore, we may preprocess all datasets before-hand in order to use them to train different classifiers. In this manner, we only need to prepare the data once. Furthermore, previous analysis on preprocessing methods with several classifiers have shown the goodness of the oversampling techniques [6].

The simplest approach, random oversampling, makes exact copies of existing instances, and therefore several authors agree that this method can increase the likelihood of occurring overfitting [6]. According to the previous fact, more sophisticated methods have been proposed based on the generation of synthetic samples. Among them, the "Synthetic Minority Over-sampling TEchnique" (SMOTE) [10] algorithm, whose main idea is to form new positive class examples by interpolating between several positive class examples that lie together, has become one of the most significant approaches in this area.

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 neighbours. Depending upon the amount of over-sampling required, neighbours from the k nearest neighbours are randomly chosen. This process is illustrated in Fig. 1, where x_i is the selected point, x_{i1} to x_{i4} are some selected nearest neighbours and r_1 to r_4 the synthetic data points created by the randomised interpolation.

Synthetic samples are generated in the following way: take the difference between the feature vector (sample) under consideration and its nearest neighbour. Multiply this difference by a random number between 0 and 1, and add it to the feature vector under consideration. This causes the selection of a random point along the line segment between two specific features. This approach effectively forces the decision region of the positive class to become more general.

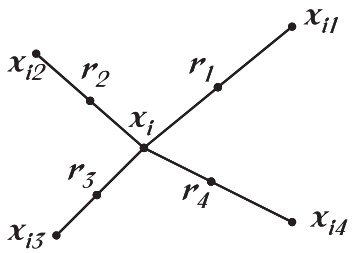


Fig. 1. An illustration of how to create the synthetic data points in the SMOTE algorithm.

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)

2.3. Evaluation in imbalanced domains

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

Traditionally, 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 to take into account the class distribution. Specifically, a well-known approach to produce an evaluation criteria in an imbalanced scenario is to use the Receiver Operating Characteristic (ROC) graphic [7]. This graphic allows to visualize the trade-off between the benefits (TP_{rate}) and costs (FP_{rate}), thus it evidences that any classifier cannot increase the number of true positives without also increasing the false positives. The Area Under the ROC Curve (*AUC*) [22] corresponds to the probability of correctly identifying which one of the two stimuli is noise and which one is signal plus noise. *AUC* provides a single measure of a classifier's performance for evaluating which model is better on average. Fig. 2 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 respectively. On the contrary, $(0,1)$ point represents the perfect classification. The *AUC* measure is computed just by obtaining the area of the graphic:

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

3. Classifier evaluation techniques and the issue of dataset shift

As stated in the introduction of this work, the estimation of the performance of a classifier, via partitioning in training and test folds, is a necessary procedure in order to validate the results for a given experiment. However, the way this task is developed has a direct influence in the analysis of the obtained models. Specifically, the issue of dataset shift can occur when the distribution of the samples in training and test is quite different between them, leading to "overfitting".

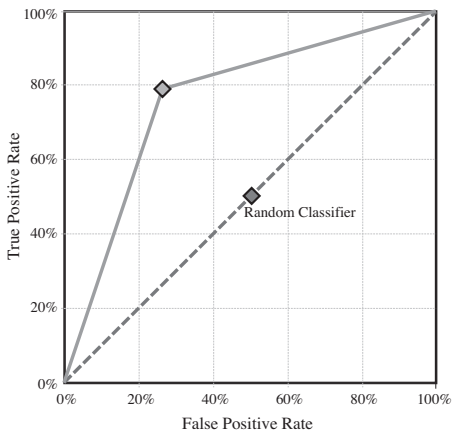


Fig. 2. Example of an ROC plot. Two classifiers' curves are depicted: the dashed line represents a random classifier, whereas the solid line is a classifier which is better than the random classifier.

In this section, we describe dataset shift in order to understand the nature of the problem we are dealing with. Next, we recall the standard and well-known SCV technique, and we identify its handicap for classification with imbalanced data. Finally, we present a recent methodology to alleviate this situation by a better organization of the instances among the different folds.

3.1. Dataset shift

The problem of dataset shift [2,8,36] 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.

There are three potential types of dataset shift:

1. *Prior Probability Shift:* It happens when the class distribution is different between the training and test sets [37]. In the most extreme example, the training set would not have a single example of a class, leading to a degenerate classifier. The problems caused by this kind of shift have already been studied, and they are commonly prevented by applying a SCV scheme [46].
2. *Covariate Shift:* In this case, it is the input attribute values that have different distributions between the training and test sets [36]. We focus on the impact of this type of shift for classification problems with imbalanced data.
3. *Concept Shift:* We refer to this problem when the relationship between the input and class variables changes [2,47], which presents the hardest challenge among the different types of dataset shift. In the specialized literature it is usually referred to as "Concept Drift" [27,45].

The dataset shift issue is specially relevant when dealing with imbalanced classification, because in highly imbalanced domains, the positive class is particularly sensitive to singular classification errors, due to the typically low number of examples it presents [29]. In the most extreme cases, a single misclassified example of the positive class can create a significant drop in performance.

For clarity, Figs. 3 and 4 present two examples of the influence of dataset shift in imbalanced classification. In the first case (Fig. 3), 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 (Fig. 4) it must be noted how some positive class examples in test are at the bottom and rightmost areas where there were not represented in the training set, leading to a gap between the training and test performance. These problems are represented in a two-dimensional space by means of a linear transformation of the inputs variables following the technique given by [29].

3.2. Cross-validation for classifier evaluation: distribution optimally balanced SCV

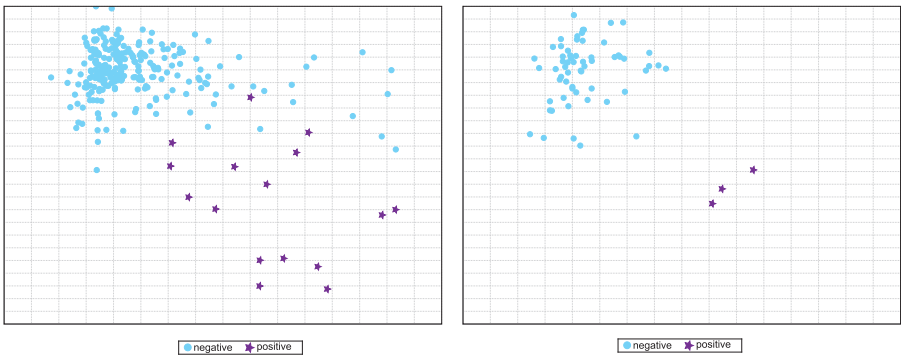
Cross-validation is a technique used for assessing how a classifier will perform when classifying new instances of the task at hand. One iteration of cross-validation involves partitioning a sample of data into two complementary subsets: training the classifier on one subset (called the training set) and testing its performance on the other subset (test set).

In k-fold cross-validation, the original sample is randomly partitioned into k subsamples. Of the k subsamples, a single subsample is retained as the validation data for testing the classifier, and the remaining $k-1$ subsamples are used as training data. The cross-validation process is then repeated k times, with each of the k subsamples used exactly once as the test data. The k results from the folds are then averaged to produce a single performance estimation.

The way the subsamples are assigned to each fold determines the impact of the final performance estimation in the validation stage. The most straightforward procedure is known as SCV, which works as follows: it counts how many samples of each class are there in the dataset, and distributes them evenly on the folds, so that each fold contains the same number of examples of each class. This avoids prior probability shift, because with an equal distribution class-wise on each fold, training and test set will have the same class distribution. However, this method does not take into account the covariates of the samples, so it can potentially generate covariate shift.

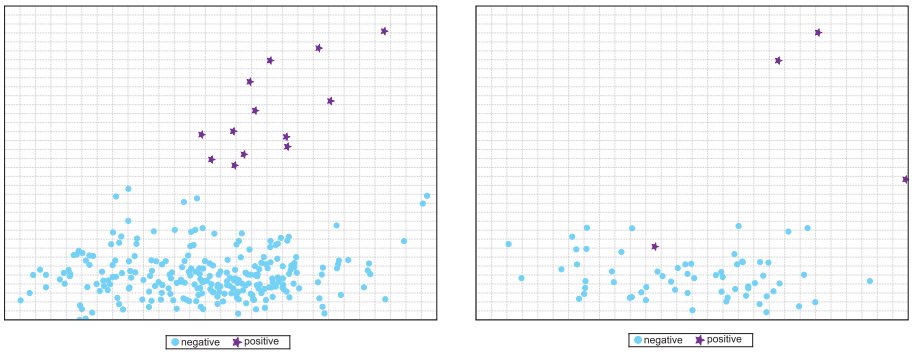
According to this fact, we consider a more sophisticated technique, known as DOB–SCV [31], which adds an extra consideration to the partitioning strategy as an attempt to alleviate the problem of covariate shift on top of preventing prior probability shift. The idea is that by assigning close-by examples to different folds, each fold will end up with enough representatives of every region, thus avoiding covariate shift.

This method is based on the Distribution-balanced SCV [52] and its pseudo-code is depicted in Algorithm 1. It picks a random unassigned example, and then finds its $k-1$ nearest unassigned neighbours of the same class. Once it has found them, it assigns each of those examples to a different fold. The process is repeated until there are no more examples of that class (when it gets to the last fold, it cycles and continues with the first one again). The whole process is repeated for each class.



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

Fig. 3. Example of good behaviour (no dataset shift) in imbalanced domains: ecoli4 dataset, 5th partition.



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

Fig. 4. Example of bad behaviour caused by dataset shift in imbalanced domains: ecoli4 dataset, 1st partition.

Algorithm 1. DOB–SCV partitioning method

for each class $c_j \in C$ **do**
  while count $(c_j) > 0$ **do**
    $e_0 \leftarrow$ randomly select an example of class c_j from D
    $e_i \leftarrow$ ith closest example to e_0 of class c_j from D $(i = 1, \ldots, k-1)$
    $F_i \leftarrow F_i \bigcup e_i (i = 0, \ldots, k-1)$
    $D \leftarrow D \backslash e_i (i = 0, \ldots, k-1)$
  end while
end for

4. Experimental framework

In this section we first provide details of the real-world binary-class imbalanced problems chosen for the experiments (Section 4.1). Then, we will describe the learning algorithms selected for this study and their configuration parameters (Section 4.2). Finally, we present the statistical tests applied to compare the results obtained with the different classifiers (Section 4.3).

4.1. Benchmark data

There is no consensus in the research community on what threshold must be set up for a given dataset to suffer from the imbalance problem. In this paper, we consider a dataset to be imbalanced when the positive class has a distribution of examples below 40% of the number of instances that belong to the majority class, that is, if the ratio between the examples of the

Table 2
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

majority and minority class is higher than 1.5. The data used in the study are summarized in Table 2, where we denote the number of examples (#Ex.), number of attributes (#Atts.) and IR. This table is in ascending order according to the IR.

As pointed out along this paper, the estimates of the AUC measure are obtained by means of a standard SCV and the DOB–SCV. The number of folds selected in both cases is 5. 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, especially for highly imbalanced datasets. Furthermore, we must point out that the original dataset partitions with 5-fold-cross-validation employed in this paper are available for download at the KEEL dataset repository [3] so that any interested researcher can use the same data for comparison.

4.2. Algorithms and parameters

In order to check the robustness of the DOB–SCV strategy, we have make use of several well-known classifiers from different Machine Learning paradigms: the C4.5 Decision Tree [34], the Chi et al. algorithm [13] as FRBCS [24], the well known *k*-NN algorithm [16] as instance-based learning method [1], and SVMs with both the Support Vector Machines with SMO optimization [15] and the Positive Definite Fuzzy Classifier (PDFC) [12]. Specifically, we have selected the following approaches as they are considered to be baseline algorithms in the field of Data Mining and they cover the widest used paradigms in classification. In this way, we can study the validity of our proposal within different types of classifiers, thus being able to generalize our extracted conclusions.

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

1. **C4.5**
   For C4.5 we have set a confidence level of *0.25*, the minimum number of item-sets per leaf was set to 2 and the application of pruning was used to obtain the final tree.
2. **Chi et al.**
   We will apply a configuration consisting in product T-norm as conjunction operator, together with the Penalized Certainty Factor approach [25] for the rule weight, and winning rule as Fuzzy Reasoning Method [14]. Furthermore, we have selected the use of 5 labels per variable.

3. ***k*-NN**
   In this case we have selected 1 neighbour for determining the output class, applying the euclidean distance metric.
4. **SMO**
   The SMO algorithm was run using polynomial reference functions, with a value of 1 in the exponent of each kernel function and a penalty parameter of the error term of *1.0*.
5. **PDFC**
   The FRBCS part of this method applies a product T-norm as the fuzzy conjunction operator, addition for fuzzy rule aggregation, and centre of area defuzzification. For the SVM part we have chosen Gaussian functions for the kernels, with an internal parameter of *0.25* and the weight of the classification error set to *100.0*.

Regarding the SMOTE preprocessing technique, we will consider the *5-nearest neighbours of the positive class* to generate the synthetic samples, and *balancing both classes to the 50% distribution*.

We must also point out that all these algorithms are available within the KEEL software tool [4].

4.3. Statistical tests for performance comparison

The goodness of a given approach cannot be only measured in terms of the improvement for the mean performance. Significant differences must be found among the different algorithms for concluding the superior behaviour of the one that achieves the highest average result.

For this reason, in this paper we use the hypothesis testing techniques to provide statistical support for the analysis of the results [19,35]. Specifically, we will use non-parametric tests, due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility with these type of tests [17].

We apply the Wilcoxon signed-rank test [35] as a non-parametric statistical procedure for performing pairwise comparisons between two algorithms, as the analogous of the paired *t*-test. This procedure computes the differences between the performance scores of the two classifiers on i^{th} out of N_{ds} datasets. The differences are ranked according to their absolute values, from smallest to largest, and average ranks are assigned in case of ties. We call R^+ the sum of ranks for the datasets on which the second algorithm outperformed the first, and R^- the sum of ranks for the opposite. Let T be the smallest of the sums, $T = min(R^+, R^-)$. If T is less than or equal to the value of the distribution of Wilcoxon for N_{ds} degrees of freedom (Table B.12 in [51]), the null hypothesis of equality of means is rejected.

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

Non-parametrical tests are suggested in the studies presented in [17,19,20], where its use in the field of machine learning is highly recommended. Any interested reader can find additional information on the Website http://sci2s.ugr.es/sicidm/.

5. Experimental study

This section is devoted to identify the possible differences regarding the estimation of the performance with the standard SCV and the suggested DOB–SCV for imbalanced datasets.

Table 3 shows the average results for the five algorithms selected for our study, namely C4.5, FRBCS (Chi et al.), 1-NN, SMO and PDFC, grouped with respect to the IR. We must recall that, in order to address imbalance, these results are computed using SMOTE as preprocessing technique.

For each classification method, three values are given: first the average AUC performance together with its standard variation obtained in the test partitions for the SCV technique, then the average performance for DOB–SCV, and finally the relative difference between both values, i.e. $\frac{AUC_{DOB-SCV} - AUC_{SCV}}{AUC_{SCV}}$. In this manner, if the value is positive it means that the estimation of the performance for DOB–SCV is more optimistic than SCV; if the value is negative it refers to the contrary case; and the

Table 3
Average test results with AUC metric and percentage differences for the SCV and DOB–SCV techniques.

Algorithm	IR < 9			IR > 9			All		
	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.
C4.5	.8597 ± .0357	.8698 ± .0393	1.28	.8133 ± .0844	.8309 ± .0751	2.83	.8288 ± .0681	.8439 ± .0632	2.32
Chi	.8151 ± .0352	.8187 ± .0380	0.51	.7698 ± .1041	.7781 ± .0909	1.24	.7849 ± .0811	.7916 ± .0733	1.00
k-NN	.8478 ± .0342	.8616 ± .0340	1.96	.8272 ± .0937	.8395 ± .0855	1.74	.8341 ± .0739	.8468 ± .0683	1.81
SMO	.8573 ± .0317	.8644 ± .0253	0.96	.8425 ± .0695	.8427 ± .0606	0.23	.8474 ± .0569	.8500 ± .0488	0.47
PDFC	.8877 ± .0293	.8901 ± .0263	0.34	.8608 ± .0819	.8672 ± .0708	0.86	.8698 ± .0644	.8749 ± .0560	0.69

Table 4
Detailed test results with AUC metric and percentage differences for the SCV and DOB–SCV techniques. Values are grouped by classification algorithm.

Dataset	IR	C4.5			Chi			*k*-NN			SMO			PDFC		
		SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.
Glass1	1.82	.7577 ± .0379	.7416 ± .0413	−2.12	.6788 ± .0663	.6567 ± .0670	−3.26	.7738 ± .0561	.8000 ± .0454	3.38	.5692 ± .0676	.6091 ± .0133	7.01	.7072 ± .0259	.7303 ± .0215	3.26
Ecoli0vs1	1.86	.9761 ± .0190	.9806 ± .0178	0.46	.9570 ± .0498	.9516 ± .0320	−0.57	.9626 ± .0302	.9704 ± .0129	0.80	.9796 ± .0219	.9808 ± .0175	0.12	.9831 ± .0167	.9841 ± .0156	0.10
Wisconsin	1.86	.9545 ± .0199	.9585 ± .0116	0.42	.5734 ± .0215	.5699 ± .0284	−0.62	.9624 ± .0177	.9655 ± .0118	0.33	.9706 ± .0103	.9728 ± .0061	0.23	.9568 ± .0051	.9566 ± .0146	−0.01
Pima	1.90	.7145 ± .0388	.7451 ± .0366	4.27	.6714 ± .0251	.7010 ± .0545	4.41	.6808 ± .0505	.6940 ± .0432	1.93	.7412 ± .0397	.7424 ± .0240	0.16	.7508 ± .0351	.7482 ± .0515	−0.35
Iris0	2.00	.9900 ± .0224	.9900 ± .0224	0.00	1.0000 ± .0000	1.0000 ± .0000	0.00	1.0000 ± .0000	1.0000 ± .0000	0.00	1.0000 ± .0000	1.0000 ± .0000	0.00	1.0000 ± .0000	1.0000 ± .0000	0.00
Glass0	2.06	.7856 ± .0234	.7709 ± .1189	−1.87	.6826 ± .0223	.6865 ± .0333	0.56	.8595 ± .0598	.8171 ± .0768	−4.93	.7117 ± .0298	.7183 ± .0123	0.93	.7576 ± .0821	.7722 ± .0650	1.92
Yeast1	2.46	.7113 ± .0295	.7117 ± .0424	0.05	.6994 ± .0142	.6974 ± .0374	−0.28	.6533 ± .0318	.6596 ± .0121	0.97	.7038 ± .0273	.7146 ± .0372	1.54	.7152 ± .0347	.7176 ± .0179	0.34
Vehicle1	2.52	.7468 ± .0125	.7222 ± .0451	−3.30	.6348 ± .0185	.6858 ± .0324	8.04	.6323 ± .0204	.6862 ± .0395	8.52	.7470 ± .0254	.7588 ± .0245	1.59	.8732 ± .0193	.8598 ± .0180	−1.53
Vehicle2	2.52	.9476 ± .0160	.9547 ± .0174	0.76	.8735 ± .0312	.8648 ± .0166	−1.00	.9539 ± .0202	.9299 ± .0149	−2.52	.9287 ± .0103	.9247 ± .0202	−0.42	.9811 ± .0074	.9806 ± .0097	−0.06
Vehicle3	2.52	.7015 ± .0281	.7290 ± .0460	3.92	.7212 ± .0123	.6946 ± .0225	−3.69	.6835 ± .0235	.6562 ± .0642	−3.99	.7282 ± .0376	.7376 ± .0370	1.30	.8401 ± .0152	.8329 ± .0237	−0.85
Haberman	2.68	.6309 ± .0407	.6521 ± .0227	3.37	.6185 ± .0266	.6123 ± .0935	−0.99	.5394 ± .0525	.6096 ± .0869	13.01	.6161 ± .0612	.6199 ± .0460	0.61	.6120 ± .0587	.6256 ± .0588	2.23
Glass0123vs456	3.19	.8832 ± .0605	.9256 ± .0363	4.80	.8640 ± .0140	.8662 ± .0655	0.25	.9224 ± .0154	.9395 ± .0393	1.85	.8819 ± .0714	.9173 ± .0331	4.02	.9292 ± .0512	.9374 ± .0386	0.88
Vehicle0	3.23	.9143 ± .0237	.9465 ± .0073	3.52	.8495 ± .0160	.8697 ± .0209	2.38	.9106 ± .0191	.9301 ± .0245	2.14	.9562 ± .0159	.9587 ± .0128	0.26	.9764 ± .0095	.9813 ± .0065	0.50
Ecoli1	3.36	.9162 ± .0485	.8661 ± .0358	−5.47	.8791 ± .0487	.8793 ± .0190	0.02	.8298 ± .0783	.8934 ± .0285	7.66	.8933 ± .0452	.8931 ± .0133	−0.03	.8967 ± .0546	.8854 ± .0153	−1.26
New-thyroid2	4.92	.9631 ± .0456	.9833 ± .0181	2.10	.9659 ± .0612	.9746 ± .0275	0.90	.9774 ± .0279	.9690 ± .0325	−0.85	.9774 ± .0296	.9944 ± .0076	1.75	.9917 ± .0076	.9917 ± .0124	0.00
New-thyroid1	5.14	.9802 ± .0371	.9690 ± .0473	−1.13	.9548 ± .0859	.9603 ± .0664	0.58	.9774 ± .0279	.9806 ± .0124	0.32	.9861 ± .0170	.9889 ± .0152	0.28	.9944 ± .0076	.9917 ± .0124	−0.28
Ecoli2	5.46	.8921 ± .0715	.8834 ± .0500	−0.97	.9170 ± .0490	.9061 ± .0400	−1.19	.9343 ± .0505	.9272 ± .0414	−0.77	.9085 ± .0469	.9046 ± .0427	−0.43	.9381 ± .0419	.9311 ± .0405	−0.75
Segment0	6.01	.9927 ± .0060	.9912 ± .0076	−0.15	.9590 ± .0121	.9649 ± .0066	0.61	.9949 ± .0066	.9934 ± .0038	−0.15	.9917 ± .0090	.9917 ± .0054	0.00	.9960 ± .0033	.9990 ± .0017	0.30
Glass6	6.38	.8450 ± .0750	.8896 ± .0839	5.28	.7969 ± .0679	.8396 ± .0834	5.36	.8686 ± .0867	.9365 ± .0664	7.82	.9057 ± .0552	.9365 ± .0719	3.40	.8938 ± .0813	.9176 ± .0714	2.66
Yeast3	8.11	.8869 ± .0344	.9086 ± .0363	2.45	.8942 ± .0337	.8881 ± .0281	−0.68	.8607 ± .0134	.8693 ± .0297	1.00	.9040 ± .0128	.9003 ± .0314	−0.41	.9224 ± .0213	.9301 ± .0246	0.83
Ecoli3	8.19	.7755 ± .0787	.8677 ± .1067	11.90	.8665 ± .0801	.8681 ± .0423	0.19	.7777 ± .0482	.8139 ± .0469	4.65	.8874 ± .0418	.8758 ± .0678	−1.31	.8798 ± .0554	.8797 ± .0462	−0.01
Page-blocks0	8.77	.9484 ± .0153	.9472 ± .0140	−0.12	.8744 ± .0185	.8752 ± .0189	0.09	.8953 ± .0155	.9135 ± .0147	2.03	.8729 ± .0215	.8774 ± .0169	0.52	.9335 ± .0101	.9300 ± .0122	−0.37
Ecoli034vs5	9.00	.8583 ± .0806	.8694 ± .1118	1.29	.8194 ± .1343	.8222 ± .1055	0.34	.8472 ± .1361	.8639 ± .1323	1.97	.8944 ± .1037	.9000 ± .0669	0.62	.8833 ± .1139	.8889 ± .1080	0.63
Yeast2vs4	9.08	.8620 ± .0589	.8716 ± .0358	1.11	.8607 ± .0492	.8727 ± .0282	1.39	.8807 ± .0655	.8905 ± .0504	1.11	.8863 ± .0287	.8963 ± .0244	1.13	.9154 ± .0608	.9201 ± .0344	0.51
Ecoli067vs35	9.09	.8125 ± .2097	.8225 ± .0945	1.23	.7925 ± .1660	.7850 ± .1084	−0.95	.8625 ± .1495	.8675 ± .1037	0.58	.8550 ± .1509	.8500 ± .0824	−0.58	.8650 ± .1687	.8800 ± .0873	1.73
Ecoli0234vs5	9.10	.8974 ± .1051	.8528 ± .0871	−4.97	.8114 ± .1577	.8725 ± .1043	7.53	.8530 ± .1261	.8808 ± .1102	3.26	.8946 ± .1109	.9029 ± .0972	0.92	.9056 ± .1135	.8862 ± .0964	−2.15
Glass015vs2	9.12	.7444 ± .1152	.6411 ± .0694	−13.87	.5583 ± .0848	.5126 ± .1752	−8.18	.6573 ± .1287	.6290 ± .1018	−4.29	.5344 ± .0400	.5737 ± .0968	7.34	.8043 ± .1182	.7793 ± .1021	−3.11
Yeast0359vs78	9.12	.7222 ± .0537	.7022 ± .0874	−2.77	.7040 ± .0631	.7063 ± .0527	0.32	.7543 ± .0384	.7188 ± .0856	−4.72	.7428 ± .0415	.7495 ± .0635	0.90	.7170 ± .0377	.7028 ± .0786	−1.99
Yeast02579vs368	9.14	.9171 ± .0164	.9044 ± .0325	−1.39	.8871 ± .0380	.8813 ± .0421	−0.65	.9044 ± .0282	.8927 ± .0493	−1.28	.9035 ± .0366	.9027 ± .0336	−0.09	.9021 ± .0319	.9037 ± .0395	0.19
Yeast0256vs3789	9.14	.7543 ± .0242	.7771 ± .0585	3.02	.7798 ± .0763	.7837 ± .0233	0.49	.7807 ± .0556	.8068 ± .0475	3.33	.7940 ± .0510	.8095 ± .0376	1.96	.8189 ± .0528	.8142 ± .0274	−0.58
Ecoli046vs5	9.15	.8729 ± .0993	.8342 ± .1094	−4.44	.8394 ± .1434	.8533 ± .1645	1.65	.8642 ± .1427	.8918 ± .1073	3.20	.8979 ± .1086	.8978 ± .1092	−0.01	.8507 ± .0926	.9086 ± .1061	6.81
Ecoli01vs235	9.17	.8041 ± .1660	.8377 ± .1359	4.18	.7441 ± .0805	.8209 ± .0691	1.32	.8286 ± .1507	.8850 ± .1069	6.80	.8577 ± .0923	.8764 ± .0838	2.17	.8868 ± .1512	.9214 ± .0720	3.90
Ecoli0267vs35	9.18	.7704 ± .1082	.8606 ± .0869	11.71	.7753 ± .0752	.7881 ± .1398	1.65	.8976 ± .0985	.8928 ± .0918	−0.53	.8731 ± .0776	.8730 ± .0840	−0.01	.8426 ± .1085	.8804 ± .0533	4.49
Glass04vs5	9.22	.9816 ± .0168	.9706 ± .0294	−1.12	.7210 ± .1989	.7224 ± .1422	0.20	.9691 ± .0383	.9206 ± .1156	−5.01	.9629 ± .0408	.9581 ± .0155	−0.50	.9636 ± .0254	.9706 ± .0416	0.72
Ecoli0346vs5	9.25	.8703 ± .0517	.8784 ± .1196	0.93	.8568 ± .0986	.8176 ± .1156	−4.57	.8838 ± .0986	.8561 ± .0933	−3.13	.8953 ± .0587	.8953 ± .0589	0.00	.9169 ± .0707	.9115 ± .0627	−0.59
Ecoli0347vs56	9.28	.8368 ± .1514	.8992 ± .0443	7.45	.8196 ± .1107	.8306 ± .1124	1.34	.8834 ± .1215	.8764 ± .0930	−0.79	.9191 ± .0888	.8905 ± .0813	−3.11	.9055 ± .0838	.9185 ± .0809	1.44
Yeast05679vs4	9.35	.7682 ± .1009	.7954 ± .0827	3.55	.7989 ± .0625	.8011 ± .0481	0.28	.7753 ± .0599	.8003 ± .0597	3.23	.7885 ± .0849	.8005 ± .0182	1.51	.7900 ± .0930	.8014 ± .0392	1.45
Ecoli067vs5	1.00	.8250 ± .0862	.8875 ± .0690	7.58	.8275 ± .0958	.7875 ± .0631	−4.83	.8675 ± .0577	.8800 ± .0677	1.44	.8675 ± .0855	.8450 ± .0841	−2.59	.8700 ± .0473	.8650 ± .1088	−0.57
Vowel0	1.10	.9433 ± .0483	.9750 ± .0155	3.36	.9789 ± .0183	.9933 ± .0015	1.47	1.0000 ± .0000	.9989 ± .0015	−0.11	.9566 ± .0117	.9599 ± .0092	0.35	.9989 ± .0015	.9994 ± .0012	0.06
Glass016vs2	1.29	.6367 ± .1255	.6752 ± .1478	6.06	.6002 ± .0841	.5140 ± .1171	−14.36	.6814 ± .1793	.6976 ± .1674	2.38	.5379 ± .1120	.5819 ± .0771	8.19	.7605 ± .1208	.7769 ± .1316	2.16
Glass2	1.39	.5424 ± .1401	.7498 ± .1155	38.25	.5206 ± .1120	.6241 ± .0982	19.89	.6447 ± .0987	.7331 ± .1316	13.72	.5985 ± .1570	.5989 ± .0841	0.07	.7688 ± .1486	.7789 ± .1081	1.31
Ecoli0147vs2356	1.59	.8461 ± .0453	.8426 ± .0669	−0.41	.7894 ± .0606	.8043 ± .0728	1.89	.8507 ± .0309	.8857 ± .1008	4.11	.8844 ± .0767	.8862 ± .0461	0.20	.9025 ± .0542	.8823 ± .0158	−2.23
Led7digit02456789vs1	1.97	.8832 ± .0962	.8207 ± .0995	−7.08	.8302 ± .0749	.7983 ± .0810	−3.84	.8108 ± .0333	.8652 ± .0432	6.70	.8875 ± .0531	.8248 ± .0549	−7.07	.8852 ± .0923	.8611 ± .0954	−2.73
Glass06vs5	11.00	.9147 ± .1186	.9600 ± .0285	4.95	.7500 ± .2215	.7850 ± .1876	4.67	.9400 ± .1207	.9200 ± .1242	−2.13	.9439 ± .0344	.9492 ± .0260	0.56	.9745 ± .0358	.9597 ± .0133	−1.51
Ecoli01vs5	11.00	.8227 ± .1074	.8523 ± .0114	3.59	.8386 ± .1447	.8500 ± .1445	1.36	.8545 ± .1525	.8909 ± .0973	4.26	.8932 ± .0756	.8977 ± .0663	0.51	.8795 ± .1018	.9114 ± .0660	3.62
Glass0146vs2	11.06	.7564 ± .1089	.7361 ± .1509	−2.68	.5146 ± .1054	.5527 ± .1197	7.39	.6453 ± .0884	.7445 ± .1301	15.37	.6157 ± .0732	.6185 ± .0496	0.45	.8029 ± .1359	.7838 ± .0650	−2.38
Ecoli0147vs56	12.28	.8641 ± .0565	.8474 ± .0425	−1.93	.8441 ± .1129	.8458 ± .0535	0.19	.8756 ± .0622	.8740 ± .0717	−0.19	.9093 ± .0353	.8928 ± .0760	−1.82	.8907 ± .0755	.9124 ± .0831	2.44

(*continued on next page*)

Table 4 (*continued*)

Dataset	IR	C4.5			Chi			*k*-NN			SMO			PDFC		
		SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.	SCV	DOB–SCV	% Diff.
Cleveland0vs4	12.62	.7210 ± .1259	.7719 ± .1180	7.05	.1188 ± .0538	.1188 ± .0580	−0.08	.8543 ± .1430	.7042 ± .0652	−17.57	.9076 ± .0619	.9010 ± .0932	−0.72	.8929 ± .0765	.8188 ± .1538	−8.30
Ecoli0146vs5	13.00	.8981 ± .0975	.8538 ± .0797	−4.93	.8481 ± .1215	.8712 ± .1330	2.72	.8481 ± .1171	.9154 ± .1120	7.94	.8846 ± .0947	.8962 ± .1101	1.30	.8750 ± .1088	.9096 ± .1127	3.96
Ecoli4	13.84	.8044 ± .1388	.8980 ± .0732	11.64	.9230 ± .0813	.9152 ± .0771	−0.85	.9171 ± .0689	.9608 ± .0527	4.77	.9481 ± .0590	.8997 ± .0632	−5.10	.9060 ± .0724	.9012 ± .0695	−0.53
Yeast1vs7	13.87	.7064 ± .0671	.6711 ± .1027	−5.00	.6524 ± .1047	.6671 ± .0913	2.26	.7479 ± .1279	.6610 ± .0746	−11.62	.7691 ± .0642	.7477 ± .0431	−2.78	.6881 ± .0521	.7071 ± .0832	2.76
Shuttle0vs4	13.87	.9997 ± .0007	.9991 ± .0008	−0.06	.9872 ± .0117	.9874 ± .0281	0.02	.9960 ± .0089	.9957 ± .0088	−0.03	.9960 ± .0089	.9960 ± .0089	0.00	.9960 ± .0089	.9960 ± .0089	0.00
Glass4	15.47	.8508 ± .0935	.8986 ± .1376	5.61	.8618 ± .1105	.8762 ± .1459	1.67	.8917 ± .1162	.9085 ± .1491	1.88	.8928 ± .1161	.8713 ± .1429	−2.41	.9251 ± .1052	.9344 ± .0786	1.01
Page-blocks13vs4	15.85	.9955 ± .0047	.9565 ± .0752	−3.91	.8928 ± .1067	.8684 ± .0810	−2.74	.9977 ± .0051	.9876 ± .0074	−1.01	.7223 ± .1226	.8096 ± .0648	12.09	.9752 ± .0124	.9741 ± .0129	−0.11
Abalone9vs18	16.68	.6201 ± .0514	.7854 ± .0794	26.66	.6744 ± .0988	.6937 ± .0938	2.86	.6820 ± .0814	.7457 ± .0669	9.34	.8458 ± .0564	.7977 ± .0524	−5.68	.8969 ± .0227	.8373 ± .0577	−6.65
Glass016vs5	19.44	.9714 ± .0143	.9686 ± .0120	−0.29	.8486 ± .2191	.8514 ± .1435	0.34	.8771 ± .2191	.9329 ± .1118	6.35	.9343 ± .0329	.9371 ± .0192	0.31	.8771 ± .2274	.9214 ± .1229	5.05
Shuttle2vs4	2.50	.9958 ± .0093	.9877 ± .0185	−0.82	.8838 ± .2160	.8840 ± .2161	0.02	1.0000 ± .0000	.9958 ± .0093	−0.42	.9960 ± .0089	.9960 ± .0089	0.00	.9960 ± .0089	.9960 ± .0089	0.00
Yeast1458vs7	22.10	.5230 ± .0170	.5889 ± .0623	12.59	.5713 ± .0830	.6061 ± .0390	6.10	.6390 ± .0778	.6290 ± .0625	−1.56	.6570 ± .0612	.6539 ± .0745	−0.46	.6569 ± .0439	.7024 ± .0548	6.92
Glass5	22.81	.8829 ± .1331	.9829 ± .0139	11.33	.7463 ± .2052	.8439 ± .1281	13.07	.8829 ± .2148	.9232 ± .1182	4.56	.9341 ± .0318	.9390 ± .0228	0.52	.8732 ± .1145	.9256 ± .0984	6.01
Yeast2vs8	23.10	.8066 ± .1122	.7490 ± .0980	−7.13	.8066 ± .0694	.7099 ± .0566	−12.00	.8055 ± .1425	.7501 ± .1096	−6.88	.7664 ± .0960	.7663 ± .0495	−0.01	.7924 ± .1055	.7892 ± .0713	−0.41
Yeast4	28.41	.7004 ± .0565	.7823 ± .0786	11.69	.8325 ± .0239	.8303 ± .0209	−0.27	.7242 ± .0593	.7668 ± .0899	5.88	.8217 ± .0430	.8352 ± .0629	1.64	.8090 ± .0774	.8155 ± .0819	0.80
Yeast1289vs7	3.56	.7051 ± .0697	.6037 ± .0724	−14.38	.6770 ± .0853	.7027 ± .0665	3.80	.6444 ± .0713	.6503 ± .0877	0.92	.7216 ± .0514	.7227 ± .0713	0.15	.6964 ± .0938	.7126 ± .0506	2.31
Yeast5	32.78	.9337 ± .0400	.9389 ± .0266	0.56	.9372 ± .0272	.9465 ± .0256	1.00	.9326 ± .0413	.9514 ± .0333	2.01	.9656 ± .0068	.9653 ± .0069	−0.04	.9611 ± .0290	.9396 ± .0302	−2.24
Ecoli0137vs26	39.15	.8136 ± .2171	.8780 ± .1215	7.92	.7917 ± .1981	.8598 ± .1340	8.60	.8281 ± .2087	.8836 ± .1263	6.69	.8490 ± .1969	.8489 ± .1209	−0.01	.8118 ± .1957	.8744 ± .1266	7.72
Yeast6	39.15	.8280 ± .1277	.7996 ± .1199	−3.44	.8820 ± .0855	.8796 ± .0488	−0.27	.7998 ± .1200	.8361 ± .1274	4.54	.8751 ± .0712	.8744 ± .0494	−0.08	.8684 ± .0610	.8562 ± .0730	−1.41
Abalone19	128.87	.5203 ± .0443	.5827 ± .0811	11.99	.6748 ± .1077	.6976 ± .0424	3.38	.5176 ± .0385	.5763 ± .0653	11.34	.7894 ± .0463	.7908 ± .0729	0.18	.6777 ± .0529	.7280 ± .1019	7.42
Average		.8288 ± .0681	.8439 ± .0632	2.32	.7849 ± .0811	.7916 ± .0733	1.00	.8341 ± .0739	.8468 ± .0683	1.81	.8474 ± .0569	.8500 ± .0488	0.47	.8698 ± .0644	.8749 ± .0560	0.69

Table 5
Wilcoxon's tests to compare the results with the DOB–SCV versus the standard SCV. R^+ corresponds to the sum of the ranks for the DOB–SCV partitioning approach and and R^- to the original SCV partitioning.

Comparison	R^+	R^-	p-value
C4.5[DOB–SCV] vs C4.5[SCV]	1391	754	0.0371
Chi[DOB–SCV] vs Chi[SCV]	1411	734	0.0267
k-NN[DOB–SCV] vs k-NN[SCV]	1536	609	0.0024
SMO[DOB–SCV] vs SMO[SCV]	1395	816	0.0639
PDFC[DOB–SCV] vs PDFC[SCV]	1366	845	0.0955

higher the obtained number, the most significant the selection of the validation approach is. Additionally, we show the detailed test results for all datasets in Table 4.

From these tables of results we may observe that for all five algorithms, the DOB–SCV validation technique achieves a higher estimation of the performance for most datasets, therefore being more robust for analyzing the quality of the models learned in imbalanced data.

Furthermore, we must point out that the degree of imbalance of the dataset has a direct impact on the diverse results over the different folds in the obtained results, i.e. the higher the IR is, the greater the differences between the standard SCV and the DOB–SCV are. In addition to the former, the standard deviation computation supports this perception: these values for both partitioning techniques are similar when the degree of imbalance is low; however, when the IR is higher we may observe that the standard deviation is much higher in contrast with low imbalanced datasets. Additionally, DOB–SCV has lower standard deviation values than SCV, therefore sustaining the reduction of the gap between training and test partitions.

This issue may arise due to the fact that, the lower the number of positive instances we have in a dataset with respect to the negative ones, the more significant is to maintain the data distribution to avoid the differences in performance between training and test.

The characteristics of specific datasets do not pose a source of knowledge when trying to observe if there is a group of them where DOB–SCV performs better than SCV. In general, DOB–SCV obtains a better performance for most of the algorithms for each dataset, however, only few of the datasets considered are able to provide a clear trend for all the algorithms: the cases where DOB–SCV obtains a better estimation than SCV (for instance, *Abalone*19 or *Glass*2) are more numerous than the contrary case (*Ecoli*2 or *Yeast*2*vs*8) and the improvement is much greater than the loss.

When trying to find a group of data with the highest differences between DOB–SCV and SCV, it is not possible to do so without also considering the algorithm underneath. For instance, if we try to observe where the greatest improvements or losses are obtained for each algorithm, we realize that the datasets obtained for one algorithm are completely different from the datasets obtained for the rest.

In order to give statistical support to the findings previously extracted, we will carry out a Wilcoxon test to compare both validation techniques with the five classification algorithms. This analysis is shown in Table 5 where the algorithms are compared by rows.

The conclusions from this test are clear, from which significant differences are found between DOB–SCV and SCV in all cases with a low p-value. Furthermore, the higher sums of the ranks for DOB–SCV tell us about the goodness of this approach.

To summarize, we must stress that DOB–SCV is a suitable methodology for contrasting the performance of the classification algorithms in imbalanced data. When the distribution of the classes is skewed, using standard estimation models may lead to misleading conclusions on the quality of the prediction. The proposed use of this model addresses the handicap of losing the generalization ability because of the way data is distributed among the different folds.

6. Concluding remarks

In this work we have proposed the use of a novel partition-based methodology, named as DOB–SCV, which aims at obtaining a better estimation of a classifier's performance by carrying out an heterogeneous organization of the instances of the classes among the different folds.

We have identified this validation technique as a very suitable procedure in the framework of imbalanced datasets. It is straightforward to realize that, in the case that one of the classes of the problem contains a fewer number of examples, and regarding to the evaluation metrics used in this scenario, introducing covariate shift between training and test will unequivocally lead to high differences in performance in the learning and validation stages.

The stable performance estimation of DOB–SCV has been contrasted versus the classical k-fold SCV, detecting significant differences between both techniques for several classifiers often used in imbalanced tasks such as C4.5, FRBCSs, k-NN and SVMs. We must highlight that avoiding different data distribution inside each fold will allow researchers on imbalanced data to concentrate their efforts on designing new learning models based only on the skewed data, rather than seeking for complex solutions when trying to overcome the gaps between training and test results. Nevertheless, neither SCV nor DOB–SCV can unequivocally guarantee to obtain the best estimate of the true error for a given problem. This can only be achieved by having infinite data or, at least, that the input data covers the whole problem space, which is not usually the case.

Acknowledgments

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

References

[1] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[2] R. Alaiz-Rodríguez, N. Japkowicz, 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, 2008.
[3] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multi-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.
[4] J. Alcalá;-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J.C. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (2009) 307–318.
[5] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (3) (2003) 849–851.
[6] G.E.A.P.A. Batista, R.C. Prati, M.C. Monard, A study of the behaviour of several methods for balancing machine learning training data, SIGKDD Explorations 6 (1) (2004) 20–29.
[7] A.P. Bradley, The use of the area under the ROC curve in the evaluation of machine learning algorithms, Pattern Recognition 30 (7) (1997) 1145–1159.
[8] J.Q. Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[9] J.R. Cano, F. Herrera, M. Lozano, Evolutionary stratified training set selection for extracting classification rules with trade off precision-interpretability, Data and Knowledge Engineering 60 (2007) 90–108.
[10] N.V. Chawla, K.W. Bowyer, L.O. Hall, W.P. Kegelmeyer, SMOTE: synthetic minority over-sampling technique, Journal of Artificial Intelligent Research 16 (2002) 321–357.
[11] N.V. Chawla, N. Japkowicz, A. Kotcz, Editorial: special issue on learning from imbalanced data sets, SIGKDD Explorations 6 (1) (2004) 1–6.
[12] Y. Chen, J. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 716–728.
[13] Z. Chi, H. Yan, T. Pham, Fuzzy Algorithms with Applications to Image Processing and Pattern Recognition, World Scientific, 1996.
[14] O. Cordón, M.J. del Jesus, F. Herrera, A proposal on reasoning methods in fuzzy rule-based classification systems, International Journal of Approximate Reasoning 20 (1) (1999) 21–45.
[15] C. Cortes, V. Vapnik, Support vector networks, Machine Learning 20 (1995) 273–297.
[16] T. Cover, P. Hart, Nearest neighbor pattern classification, IEEE Transactions on Information Theory 13 (1967) 21–27.
[17] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[18] P. Domingos, Metacost: a general method for making classifiers cost-sensitive, in: Proceedings of the 5th International Conference on Knowledge Discovery and Data Mining (KDD'99), 1999.
[19] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.
[20] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2607–2624.
[21] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.
[22] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.
[23] Y.-M. Huang, C.-M. Hung, H.C. Jiau, Evaluation of neural networks and data mining methods on a credit assessment task for class imbalance problem, Nonlinear Analysis: Real World Applications 7 (4) (2006) 720–747.
[24] H. Ishibuchi, T. Nakashima, M. Nii, Classification and Modeling with Linguistic Information Granules: Advanced Approaches to Linguistic Data Mining, Springer-Verlag, 2004.
[25] H. Ishibuchi, T. Yamamoto, Rule weight specification in fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 13 (2005) 428–435.
[26] N. Japkowicz, S. Stephen, The class imbalance problem: a systematic study, Intelligent Data Analysis Journal 6 (5) (2002) 429–450.
[27] T. Lane, C.E. Brodley, Approaches to online learning and concept drift for user identification in computer security, in: KDD, 1998.
[28] V. López, A. Fernández, J.G. Moreno-Torres, F. Herrera, Analysis of preprocessing vs. cost-sensitive learning for imbalanced classification. Open problems on intrinsic data characteristics, Expert Systems with Applications 39 (7) (2012) 6585–6608.
[29] J.G. Moreno-Torres, F. Herrera, A preliminary study on overlapping and data fracture in imbalanced domains by means of genetic programming-based feature extraction, in: Proceedings of the 10th International Conference on Intelligent Systems Design and Applications (ISDA'10), 2010.
[30] J.G. Moreno-Torres, T. Raeder, R. Aláiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recognition 45 (1) (2012) 521–530.
[31] J.G. Moreno-Torres, J.A. Sáez, F. Herrera, Study on the impact of partition-induced dataset shift on k-fold cross-validation, IEEE Transactions on Neural Networks and Learning Systems 23 (8) (2012) 1304–1313.
[32] A. Orriols-Puig, E. Bernadó-Mansilla, Evolutionary rule-based systems for imbalanced datasets, Soft Computing 13 (3) (2009) 213–225.
[33] A. Orriols-Puig, E. Bernadó-Mansilla, D.E. Goldberg, K. Sastry, P.L. Lanzi, Facetwise analysis of XCS for problems with class imbalances, IEEE Transactions on Evolutionary Computation 13 (2009) 260–283.
[34] J. Quinlan, C4.5: Programs for Machine Learning, Morgan Kauffman, 1993.
[35] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, Chapman & Hall/CRC, 2006.
[36] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log-likelihood function, Journal of Statistical Planning and Inference 90 (2) (2000) 227–244.
[37] A. Storkey, When training and test sets are different: characterizing learning transfer, in: J.Q. Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence (Eds.), Dataset Shift in Machine Learning, MIT Press, 2009, pp. 3–28.
[38] Y. Sun, A.K.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, International Journal of Pattern Recognition and Artificial Intelligence 23 (4) (2009) 687–719.
[39] Y. Tang, Y.-Q. Zhang, N.V. Chawla, S. Kresser, SVMs modeling for highly imbalanced classification, IEEE Transactions on Systems, Man and Cybernetics, Part B 39 (1) (2009) 281–288.
[40] K.M. Ting, An instance-weighting method to induce cost-sensitive trees, IEEE Transactions on Knowledge and Data Engineering 14 (3) (2002) 659–665.
[41] M. Wasikowski, X.-W. Chen, Combating the small sample class imbalance problem using feature selection, IEEE Transactions on Knowledge and Data Engineering 22 (10) (2010) 1388–1400.
[42] G.M. Weiss, Mining with rarity: a unifying framework, SIGKDD Explorations 6 (1) (2004) 7–19.
[43] G.M. Weiss, F.J. Provost, Learning when training data are costly: the effect of class distribution on tree induction, Journal of Artificial Intelligence Research 19 (2003) 315–354.

[44] G.M. Weiss, Y. Tian, Maximizing classifier utility when there are data acquisition and modeling costs, Data Mining and Knowledge Discovery 17 (2) (2008) 253–282.
[45] G. Widmer, M. Kubat, Learning in the presence of concept drift and hidden contexts, Machine Learning 23 (1) (1996) 69–101.
[46] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques, second ed., Morgan Kaufmann, San Mateo, CA, 2005.
[47] K. Yamazaki, M. Kawanabe, S. Watanabe, M. Sugiyama, K.-R. Müller, Asymptotic bayesian generalization error when training and test distributions are different, in: Z. Ghahramani (Ed.), ICML, ACM International Conference Proceeding Series, vol. 227, ACM, 2007.
[48] Q. Yang, X. Wu, 10 Challenging problems in data mining research, International Journal of Information Technology and Decision Making 5 (4) (2006) 597–604.
[49] B. Zadrozny, C. Elkan, Learning and making decisions when costs and probabilities are both unknown, in: Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining (KDD'01), 2001.
[50] B. Zadrozny, J. Langford, N. Abe, Cost-sensitive learning by cost-proportionate example weighting, in: Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM'03), 2003.
[51] J.H. Zar, Biostatistical Analysis, Prentice Hall, Upper Saddle River, New Jersey, 1999.
[52] X. Zeng, T.R. Martinez, Distribution-balanced stratified cross validation for accuracy estimation, Journal of Experimental and Theoretical Artificial Intelligence 12 (1) (2000) 1–12.