Knowledge-Based Systems 90 (2015) 153–164

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Using the One-vs-One decomposition to improve the performance of class noise filters via an aggregation strategy in multi-class classification problems

Luís P.F. Garcia [a,*], José A. Sáez [b], Julián Luengo [c], Ana C. Lorena [d], André C.P.L.F. de Carvalho [a], Francisco Herrera [e,f]

[a] Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, Trabalhador São-carlense Av. 400, São Carlos, São Paulo 13560-970, Brazil
[b] ENGINE Centre, Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, Wrocław 50–370, Poland
[c] Department of Civil Engineering, Computing Systems and Languages, University of Burgos, Burgos 09006, Spain
[d] Instituto de Ciência e Tecnologia, Universidade Federal de São Paulo, Talim St. 330, São José dos Campos, São Paulo 12231-280, Brazil
[e] Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, Granada 18071, Spain
[f] Faculty of Computing and Information Technology, King Abdulaziz University, North Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 12 March 2015
Revised 25 August 2015
Accepted 22 September 2015
Available online 9 October 2015

Keywords:
Noisy data
Class noise
Noise filters
Decomposition strategies
Classification

ABSTRACT

Noise filters are preprocessing techniques designed to improve data quality in classification tasks by detecting and eliminating examples that contain errors or noise. However, filtering can also remove correct examples and examples containing valuable information, which could be useful for learning. This fact usually implies a margin of improvement on the noise detection accuracy for almost any noise filter. This paper proposes a scheme to improve the performance of noise filters in multi-class classification problems, based on decomposing the dataset into multiple binary subproblems. Decomposition strategies have proven to be successful in improving classification performance in multi-class problems by generating simpler binary subproblems. Similarly, we adapt the principles of the One-vs-One decomposition strategy to noise filtering, making the noise identification process simpler. In order to integrate the filtering results achieved in the binary subproblems, our proposal uses a soft voting approach considering a reliability level based on the aggregation of the noise degree prediction calculated for each binary classifier. The experimental results show that the One-vs-One decomposition strategy usually increases the performance of the noise filters studied, which can detect more accurately the noisy examples.

1. Introduction

Real-world data usually contain errors or noise [1–4]. In classification problems, a classification model must be induced from labeled examples and this classifier should be capable of reliably predicting the true class of new examples. The correct assignment of class labels to the training examples has a strong impact on the predictive quality of the induced classifiers. Thus, errors in the class labeling of the training examples may severely harm the predictive performance and complexity of the induced classifiers [1,5,6]. This type of error is known in the literature as *class noise* or *label noise* [2].

In the case of multi-class classification problems, binary decomposition strategies [7] are usually employed to allow the usage of well-known algorithms originally proposed for binary classification problems, such as *Support Vector Machines* (SVM) [8], in multi-class tasks. These strategies decompose the original problem into several binary subproblems of a lower complexity. The most popular decomposition schemes are *One-vs-One* (OVO) [9], which induces a classifier to distinguish between each pair of classes, and *One-vs-All* (OVA) [9], which induces a classifier to distinguish each class from all other classes.

The behavior of the OVO strategy in presence of noise was studied by Sáez et al. in [10]. In order to analyze whether OVO was able to reduce the harmful effects of noise in the classification results, several classification algorithms with and without the usage of this decomposition were compared. The experimental results showed that, in the presence of noisy data, decomposition generally offers better classification performance than solving the original multi-class

* Corresponding author. Tel.: +551633738161; fax: +551633739633.
E-mail addresses: lpgarcia@icmc.usp.br, lpfgarcia@gmail.com (L.P.F. Garcia), jose.saezmunoz@pwr.edu.pl (J.A. Sáez), jluengo@ubu.es (J. Luengo), aclorena@unifesp.br (A.C. Lorena), andre@icmc.usp.br (A.C.P.L.F. de Carvalho), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.knosys.2015.09.023
0950-7051/

problem. These improvements are mainly attributed to the distribution of the noisy examples in the binary subproblems. Furthermore, the separability of the classes is increased, while it is also possible to collect information from different classifiers.

Another alternative to overcome the problems resulting from the presence of class noise is the usage of noise filtering techniques, which remove potentially noisy examples in a preprocessing step [11,12]. Several studies show the benefits from their usage regarding improvements in the classification predictive performance and the reduction in the complexity of the classifiers built [5,13–15]. Noise filters can use different information to detect noise, such as those employing neighborhood or density information [11,16,17], descriptors extracted from the data [13,18] and noise identification models induced by classifiers [13] or ensembles of classifiers [5,14,19,20]. In other papers, they are also used to remove predictive noise [21] and investigate the presence of noise in imbalanced datasets [22,23]. Since each filter has a bias, it may have a distinct performance depending on the data used [24,25]. Thus, it is common the existence of a margin of improvement on the noise detection accuracy of filtering methods.

This paper investigates a new approach to detect and remove label noise in multi-class classification tasks. This approach combines the OVO multi-class decomposition strategy with a group of noise filtering techniques. In this combination, each noise filter, instead of being applied to the original multi-class dataset, is applied to each binary subproblem produced by the OVO strategy. Each noise filter assigns to each training instance a degree of confidence of the example being noisy, named *noise degree prediction* (NDP), which is a real number. However, some noise filters only output two values: noisy and not noisy. If so, the noise filter is adapted to output NDPs. For each training instance, the NDPs obtained from all noise filters are combined using a soft voting strategy, producing a unique NDP for the instance. The strategy adopted in this paper is to remove a fixed number of the examples with highest NDP values.

The proposed approach has three main advantages: (i) it does not require any modification in the concept and the bias of the noise filters; (ii) it provides for each training instance a combined degree of confidence regarding noise identification and; (iii) it does not make any assumptions about the noise characteristics.

In order to evaluate the impact of using the OVO strategy for noise filtering in multi-class tasks, we present an empirical study using several well-known noise filters found in the literature that will be adapted for soft voting [5,13,14,16,20] and a large amount of datasets with different levels of class noise [1]. The differences between the filtering with and without decomposition will be analyzed based on the accuracy of the noise filters detecting the noisy examples in each scenario.

The rest of this paper is organized as follows. Section 2 points out the main motivations for this study, presenting an overview on noise filtering techniques and the motivations for the use of decomposition strategies in multi-class problems. Section 3 details the approach proposed for noise detection. Section 4 describes the experimental framework, whereas Section 5 analyzes the experimental results obtained by the noise filters with and without decomposition. Finally, Section 6 presents the main conclusions from this study. A website with additional information, such as the datasets employed and the results of each noise filter is available at http://www.biocom.icmc.usp.br/~lpfgarcia/ovo.

2. Preliminaries

This section presents the background to support our proposal. Section 2.1 describes the main aspects of class noise treatment with a brief overview of the noise filtering techniques employed. Then, Section 2.2 introduces the usage of binary decomposition strategies that are commonly employed in multi-class classification.

2.1. Class noise treatment by noise filtering

Noise filters [5,13–16,20] are preprocessing methods commonly used to identify and remove noise in a dataset [2]. Most of the existing filters focus on the elimination of examples with class noise, which has shown to be advantageous [18]. In contrast, the elimination of examples with feature noise is not as beneficial [1], since other attributes from these examples may be useful to build the classifier.

Most of the noise filters [5,14,20] adopt a *crisp decision* for noise identification, classifying each training example either as either noisy or safe. *Soft decision* strategies, on the other hand, assign a noise degree prediction to each example, NDP values. The soft decision helps to correctly identify examples, those whose identification as noisy is more difficult. Besides, it makes easier the combination of multiple filters, a strategy proposed in this paper.

Next, the noise filters used in the experiments performed for this study are briefly presented. Since they were all proposed for crisp noise detection, their adaptation to allow soft decision is also discussed. The following filtering methods were used in this study, each belonging to a different filtering paradigm:

1. **All-*k*-NN** (AENN) [16]. Distance-based approaches uses the *k*-NN decision rule [16,26] to identify noisy data. Techniques following this approach assume that an example is likely to be noisy if it is located close to other examples from a different class. These noise filters are able to remove examples with class noise and examples lying on the decision border, which increases the margin of separation between the classes. A well known technique from this group is *All-k-NN* (AENN) [16]. This filter applies, iteratively, the *k*-NN classifier with several increasing values of *k*. Examples misclassified by their neighbors are marked as noisy and eliminated from the dataset. The soft version of this technique estimates the NDP of an example as the percentage of times it is labeled as noisy in different iterations.
2. **Prune Saturation Filter** (PruneSF) [13]. Complexity-based approaches extract complexity measures from the training data [13,18]. For instance, the *Saturation Filter* (SF) [13] exhaustively looks for examples that reduce a metric called *Complexity of the Least Correct Hypothesis* (CLCH) associated with a dataset. The size of a Decision Tree (DT) without pruning is used to estimate the CLCH value [13]. If the removal of an example reduces the CLCH value, it is marked as noisy. Next, the method carries out a new search in the dataset without this example and repeats the same procedure until no example is marked as noisy or a stopping criterion is reached. PruneSF [13] is based on SF. It uses a DT with pruning in a previous step to overcome computation time restrictions. Therein, first a pruning step removes all examples misclassified by a pruned DT, which are regarded as noisy. Afterwards, the iterative procedure described for SF is performed. In our work, a soft decision is obtained by firstly ranking all examples removed in the pruning step as noisy with a probability of 1. Next, the examples are ranked according to their CLCH values, which are normalized to give their probability of being noisy.
3. **High Agreement Random Forest** (HARF) [20]. This is a well-known classifier-based filter that uses a *Random Forest* classifier [27]. This technique considers the rate of disagreement in the predictions made by the individual trees in the forest to detect the noisy examples: if this rate is high, the example is probably noisy; otherwise, it is considered to be clean. A soft decision for this filter can be obtained by the percentage of base trees that disagree on their predictions for a particular instance.
4. **Static Ensemble Filter** (SEF) [5]. Ensemble-based approaches employ ensembles of classifiers to identify the noisy examples [5,14,20]. Their motivation is that different classification models provide a better alternative for detecting mislabeled examples than using information from a single model only [5]. SEF

[5] uses a set of three learning algorithms (C4.5 [28], *k*-Nearest Neighbor (*k*-NN) [29] and SVM [8]) to identify and remove the potentially noisy examples. The training data is classified using *k*-fold cross-validation and the noisy examples are those misclassified by more than half of the classifiers (majority voting). The soft decision for SEF is computed as the percentage of disagreements between the predictions of the classifiers.

5. **Dynamic Ensemble Filter** (DEF) [14]. By using a fixed set of classifiers, the predictive performance of SEF may be affected by the bias of the classifiers employed. To overcome this problem, DEF [14] dynamically selects the most suitable set of classifiers for a given dataset. The selected classifiers are those that obtain the best predictive performance on the training data using *k*-fold cross-validation. Finally, similarly to SEF, a majority vote is used to determine whether an example is noisy or not. The adaptation of DEF for soft voting is similar to that of DEF. Therefore, examples misclassified by more classifiers will be considered unsafe and, as a result, will be assigned a higher probability of being noisy.

2.2. Binary decomposition strategies in classification problems

Many real-world classification tasks, such as text classification [30], medical diagnosis [31] and intrusion detection [32], are characterized by having more than two class labels. They are known as multi-class classification problems. Usually, it is easier to build a classifier to distinguish only between two classes (called *binary classifiers*) than among a higher number of classes, since the dataset conformations and decision boundaries for multi-class problems tend to be more complex.

In order to be able to use binary classifiers in multi-class problems, two different approaches are found in the literature [7]: (1) adaptation of a learning algorithm to manage more than two classes and (2) decomposition of the multi-class problem into a set of easier to solve binary subproblems. The former requires the adaptation of the learning procedure of an existing method, which may be a difficult task [33]. The second alternative is usually an easier, yet accurate way, to efficiently deal with the original problem [9]. These techniques are referred to as binary decomposition strategies [7].

Galar et al. [9] list various benefits of using decomposition strategies. Although they are more frequently used to allow binary classification techniques to address multi-class problems, these strategies can also make the separation of the classes less complex. The decomposition also allows to parallelize the classifiers learning, since the binary subproblems are independent and can be solved in different processors.

Decomposition strategies have two steps. At the first stage, the problem is decomposed into several binary subproblems which are solved by independent binary classifiers, called *base classifiers* [34]. In a second phase, the outputs obtained for each subproblem need to be aggregated. Even though different decomposition strategies can be found in the literature, the most widely used ones are the following [7]:

1. The *One-vs-One* (OVO) decomposition induces a classifier for each pair of classes, dividing a classification problem with M classes into $M(M-1)/2$ binary subproblems. The induction of the classifier for each pair of classes uses only training examples from these classes.
2. The *One-vs-All* (OVA) scheme induces a different classifier to distinguish each class from all the other classes. Thus, it divides a classification problem with M classes into M binary subproblems considering all the training examples, which are then used to induce M different classifiers.

At the second stage of a binary decomposition, the outputs from the binary classifiers are combined into a single output, the predicted class. Galar et al. [9] provided an exhaustive study comparing different methods to combine the outputs from the base classifiers in the OVO and OVA strategies. The weighted voting [35] and the methods from a framework of probability estimates [36] presented the best predictive performances. However, a voting strategy, where the class with the largest number of votes is selected, is the most used and simplest decision combination strategy, with predictive performance similar to those of the most complex strategies [9]. When classifying an example using the OVO approach, it is also possible to use a tournament or decision on *Directed Acyclic Graphs* (DAG) [37]. Therein, an initial two-class classifier is consulted and one of the classes is eliminated from further analysis, while the predicted class is tested against another class. This process is repeated until one single class remains. Although it is a relevant strategy and has been successfully used in some multi-class applications, this combination is not suitable for our filtering scenario, where the objective is to classify an example as noisy or clean. In this case, the decomposition will give more attention to the noisy cases, instead of selecting one of the multiple classes.

A decomposition strategy frequently compared with OVO is the OVA strategy. There are several advantages in the use of OVO instead of OVA [9,34,38]. The main benefits of the OVO decomposition strategy discussed in the literature are: the construction of simpler decision borders between the classes and the increase of classification performance with less training time, since the complexity of the subproblems generated is smaller. Besides, this is the binarization technique mostly used as default by learning algorithms, applications and software tools in Machine Learning and Data Mining [39–41]. Moreover, OVA tends to produce imbalanced classification tasks, which can harm the base classifiers performance for some classes [9,42]. The same behavior can be expected for the noise filters, where the proportion of examples in the minority classes can be further reduced, so they may be considered as noisy cases. Finally, Sáez et al. [10] pointed out additional advantages of using the OVO decomposition instead of the OVA decomposition for noisy data.

Therefore, this paper proposes to use OVO decomposition strategy, not for classification purposes as they are traditionally employed, but for noise preprocessing. In the same way that decomposition helps to improve the performance of classifiers in multi-class problems, one may expect that they can help to improve the performance of noise filtering methods when detecting noisy examples in multi-class datasets, since filters will work over simpler binary subproblems where the noisy examples can be more easily identified. Next section introduces this proposal.

3. A noise filtering scheme based on the OVO decomposition strategy

This section describes the noise filtering scheme proposed, which is based on the usage of OVO to improve the accuracy of noise filters, establishing a parallelism with the standard usage of decomposition strategies in classification. The proposal is composed of three main steps:

1. **Problem decomposition.** In this phase, the multi-class classification problem is decomposed into p binary subproblems using the OVO decomposition strategy.
2. **Filtering of each binary subproblem.** When using decomposition for classification purposes, a classifier is built for each one of the p binary subproblems. Similarly, our proposal applies a noise filter to each one of the subproblems created in the previous step. Since the noise filters are adapted to output a confidence level regarding the noise predictions (the NDP values), this step results in p different lists of NDP values $(N_1, \ldots, N_p)$, each one with the NDP values of the examples belonging to the binary subproblem in which the filter is applied.

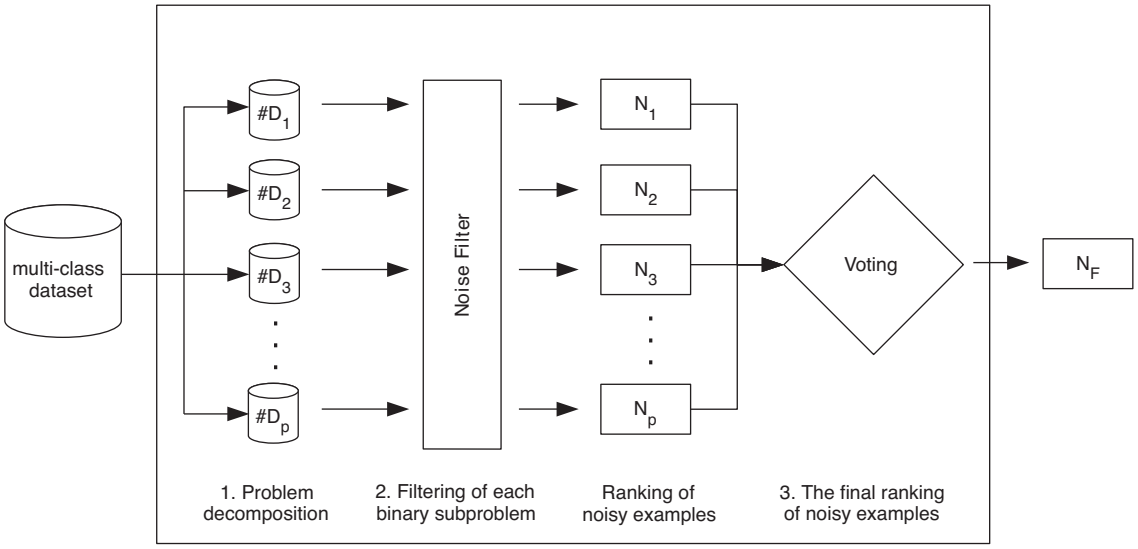


Fig. 1. Noise filtering scheme using decomposition strategies.

3. **Combination of the lists of noisy examples.** In the same way that the predictions of the different classifiers must be combined when using decomposition for classification, our proposal also requires a last step where the different lists of NDP values $(N_1, \ldots, N_p)$ are combined. Thus, a unique final list of NDP values (N_F), which will be ordered from the highest to lowest value (from the noisiest example to the cleanest example), is computed in this last step. For the combination of these lists, the average NDP of each example is computed.

Fig. 1 describes the noise filtering scheme proposed. The first step employs the OVO decomposition in the multi-class dataset. The next step is responsible of applying a filter technique to each binary subproblem and return the lists of NDP values. The third step constructs the final list of NDP values considering the lists obtained in the previous step.

The following sections describe each of these steps in depth. Section 3.1 is devoted to the problem decomposition, Section 3.2 describes the application of the noise filter to each binary subproblem, whereas Section 3.3 shows how the different lists of NDP values are combined in order to obtain the final list of NDP values of each training example. Finally, Section 3.4 discusses the computational cost of the decomposition strategy proposed for noise filtering.

3.1. Problem decomposition

Decomposition has shown to be advantageous when building classifiers from noisy datasets [10]. This fact is mainly attributed to the distribution of the noisy examples in each subproblem, which reduces the complexity of the original problem, while increasing the separability of the classes. It also allows to combine information from different models, where the failure of some models can be corrected by the remaining models.

The usage of binary decomposition strategies can also help to filter noisy examples in multi-class problems. Usually, a higher number of classes in a dataset implies in a higher complexity due to the need to consider more relationships between the classes. Since the decomposition of the multi-class problem can create simpler subproblems (with a higher degree of separation between the classes) and distributes the noisy examples in several subproblems, the noise filters can improve their detection capabilities when compared with preprocessing the original multi-class dataset. Thus, the use of OVO is expected to increase the accuracy of the noise filters in multi-class data. Therefore, the first step of the proposed method decomposes the original multi-class classification problem into p binary subproblems $D_1, \ldots, D_p$. When using OVO decomposition to fulfill this task, $p = M(M-1)/2$.

The artificial multi-class dataset shown in Fig. 2 illustrates these issues. Fig. 2a shows the original multi-class artificial dataset, composed of 3 classes (•, ▲ and ■). The possible borders between the classes are also shown. Fig. 2b shows the same artificial dataset with three potential noisy examples. Relabeling these examples changes the decision borders, which became more complex. Fig. 2c illustrate the effect of the OVO decomposition strategy in this noisy dataset. It is possible to check the simplification of the class borders due to the decomposition. A noise filter applied to these datasets is able to easily identify the noisy examples with a high confidence (○, △ and □) in each one of the two-class datasets.

3.2. Filtering of each binary subproblem

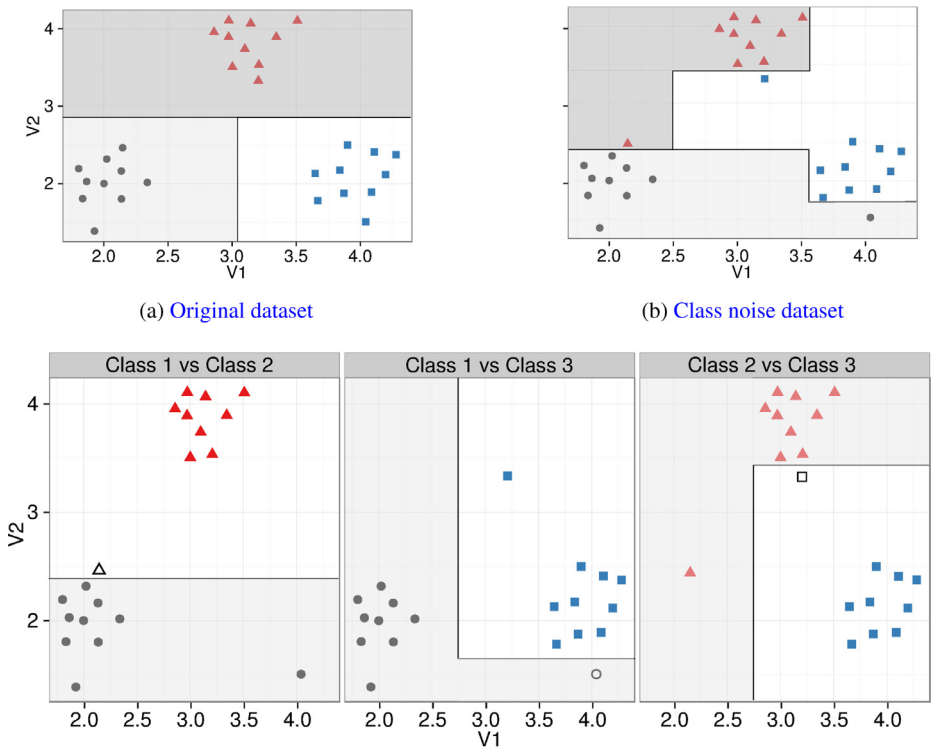
Once the p binary subproblems have been created, the second step applies a noise filter to each of them. This filtering method should be adapted to provide a soft decision on noise prediction (NDP values) to each one of the examples belonging to these subproblems. Since a NDP value represents the probability of an example being noisy, it must be in the interval [0,1].

Thus, the aforementioned process results in p different lists of NDP values $N_1, \ldots, N_p$, each one referring to the examples belonging to each one of the binary subproblems $D_1, \ldots, D_p$. The NDP values v_i^j of each list N_i are normalized applying the transformation $v_i^j \leftarrow (v_i^j - min_i)/(max_i - min_i)$, where min_i and max_i are the minimum and maximum NDP values provided by the noise filter in the subproblem D_i.

The p normalized lists must be combined in the last step to produce the final list of NDP values of all the training examples (N_F). It must be observed that our proposal can employ any existing filtering method in this step that provides a soft decision or can be adapted for such, so it should be simple to employ various filters for noise identification and removal.

3.3. Final ranking of noisy examples

The last step of the noise filtering scheme proposed builds the final list of NDP values (N_F) for the multi-class problem. Thus, the p normalized lists of NDP values obtained in the previous step must be combined to form the final list N_F.



(a) Original dataset

(b) Class noise dataset

(c) OVO decomposition for class noise dataset

Fig. 2. The effect of the OVO decomposition in the reduction of complexity of the borders between the classes.

The average value of all the occurrences of each example in the different subproblems is considered as its final NDP value in the list N_F. This is a simple yet effective strategy which allows obtaining a combined NDP value. Finally, all the training examples are ordered from highest NDP value (which is most likely to be noisy) to the lowest NDP value (which is most likely to be a clean example).

The final removal of noisy examples can be made following different approaches. For example, a domain expert can fix a threshold in order to remove all those examples which exceeds it. A second alternative is to determine different thresholds to define which examples should be removed, each defining different percentages of the noisy examples to be removed. A third alternative, adopted in this paper, is to remove a fixed number of examples, the r examples with highest NDP values. Since we will introduce the noisy examples in a controlled way in the dataset (in order to know the exact number of noisy examples in each dataset), this third alternative can help us to better estimate the performance of the noise filters in the detection of the artificially introduced noisy examples.

3.4. Analysis of computational cost

There are two main components in the computational cost of the proposed approach regarding the application of the noise filter over the original multi-class dataset: (i) the application of the filtering using the OVO decomposition and (ii) the combination of the outputs from the different noise filters.

For the first component, since the OVO decomposition strategy is applied to the multi-class problem, the filtering occurs for $M(M-1)/2$ datasets, where M is the number of classes. Therefore, the cost of each filter is multiplied by $O(M^2)$. The second component, the combination of NDPs for each training example into one NDP has a cost of $O(n)$, where n is the number of training examples.

It should be observed that each OVO binary subproblem has less examples than the original problem. Therefore, even though the cost of the filter is multiplied by $O(M^2)$, each filter application is usually much faster than the application of the same filter to the original dataset.

Finally, although the application of the proposed approach can be slower than the application of the noise filters to the original multi-class dataset, the time should not be a strong concern, since the application occurs only once (both approaches are offline preprocessing methods). Moreover, the overall time cost of the proposal can be reduced if its internal filters are implemented in a parallel architecture, which can take advantage of the distributed nature of the proposed approach.

4. Experimental framework

This section describes the experiments carried out in this paper to evaluate the behavior of the noise filtering scheme based on the usage of the OVO binary decomposition strategy. First, Section 4.1 describes the datasets used. Section 4.2 presents the noise filters considered. Finally, Section 4.3 describes the methodology followed to analyze the results.

4.1. Datasets

The experimentations were carried out using 28 multi-class classification datasets taken from the UCI and KEEL-dataset repositories [43,44]. Table 1 summarizes the main characteristics of these datasets, organized according to their number of examples, number of attributes (in parenthesis, showing the number of numerical/categorical features), imbalanced ratio (IR) measure and number of classes [45]. The examples containing missing values were

Table 1
Characteristics of the real-world datasets.

# Instances	# Attributes	# IR	# Classes		
			$M < 5$	$5 \leq M < 10$	$10 \leq M < 100$
$n < 100$	$10 \leq a < 100$	$1 < IR < 5$	Zoo(1/15)		
$100 \leq n < 1000$	$a < 10$	$IR = 1$	Iris (4/0)		
			Tae (3/2)		
		$1 < IR < 5$	Hayes-roth (4/0)		Led7digit (7/0)
		$5 \leq IR < 10$	Balance (4/0)	Ecoli (7/0)	
			Newthyroid (5/0)		
	$10 \leq a < 100$	$IR = 1$	Vehicle (18/0)		Movement-libras (90/0)
					Vowel (10/0)
		$1 < IR < 5$	Wine (13/0)	Breast-tissue (9/0)	Collins (20/1)
				Flags (2/26)	
		$5 \leq IR < 10$		Glass (9/0)	
		$IR \geq 10$		Expgen (79/0)	
$1000 \leq n < 10,000$	$a < 10$	$IR \geq 10$	Car (0/6)	Yeast (8/0)	Abalone (7/1)
	$10 \leq a < 100$	$IR = 1$		Segmentation (18/0)	
		$1 < IR < 5$	Cmc (2/7)	Landsat (36/0)	
		$5 \leq IR < 10$		Flare (0/11)	
		$IR \geq 10$		Page-blocks (10/0)	Cardiotocography (20/0)
				Wine-quality (11/0)	
$n \geq 10,000$	$a < 10$	$IR \geq 10$	Nursery (0/8)		

removed from the datasets. The datasets were grouped into distinct categories according to their characteristics: from small ($n < 100$) to very large datasets ($n \geq 10{,}000$); from a low dimensionality ($a < 10$) to a medium/high dimensionality ($10 \leq a < 100$); from balanced ($IR = 1$), to highly imbalanced ($IR \geq 10$); and finally from a small number of classes ($M < 5$) to a high number of classes ($10 \leq M < 100$).

In order to control the amount of noise in each dataset and verify how it affects the noise filtering methods, noise is introduced into each dataset in a supervised manner. In this paper we use the uniform random noise method to noise imputation, where each example has the same probability of having its label exchanged by another label [46]. Noise was injected at the rates of 5, 10, 20 and 40%. As a result, we are able to check the influence of increasingly noise levels in the detection results achieved. For each dataset and noise level, we generated 10 different noisy versions. Thus, 1120 noisy datasets with class noise were created from the aforementioned 28 base datasets. All these multi-class datasets are available on the website associated with this paper.

4.2. Noise filters

The proposal presented in this paper can use any existing noise filter providing a soft decision on noise identification or that can be adapted for such. For the sake of generality, we will evaluate the behavior of the proposal using five different up-to-date noise filtering techniques described in Section 2.1, which are well-known representatives of the field and present different biases [2]. All of them were adapted to output a NDP value. They are HARF, SEF, DEF, PruneSF and AENN. SEF and DEF combine three classifiers, PruneSF estimates the CLCH values using an unpruned DT induced by C4.5 [28] and the AENN technique is run varying the k value from 1 to 9.

4.3. Methodology

In order to assess the performance of the noise filtering scheme proposed, the behavior of each one of the five aforementioned noise filters for the multi-class and decomposed problems was measured. The ability of the filters in noise detection is recorded in both scenarios. Finally, the performance achieved by the filters in noise retrieval for the multi-class and decomposed subproblems are compared.

To perform these comparisons, the *precision at N* ($p@N$) metric is used, as suggested in [47]. Thereby, N is a threshold on the number of examples in N_F that will be regarded as noisy. We set N to be the number of noisy examples artificially introduced in the datasets, as in [47]. The precision in noise detection is then defined as the number of correctly identified noisy cases (*#correct_noisy*) divided by the number of examples identified by the filter as noisy (the threshold N):

$$p@N = \frac{\#correct\_noisy}{N} \tag{1}$$

Three types of analyzes are performed:

1. **Evaluation of the performance of the filters in noise identification for the original multi-class problems and for the OVO decomposition strategy** (Section 5.1). The first analysis considers the average of the $p@N$ performance of each filter over all noise levels in a dataset. The average ranking of the filters performance before and after decomposition in all datasets are also compared. The objective in this case is to identify for which filters the OVO decomposition strategy shows more improvements in noise identification.
2. **Analysis of the performance of the filtering scheme for different noise levels** (Section 5.2). The second analysis considers the performance of the filters for each noise level. The purpose is to analyze the behavior of the decomposition scheme for different noise levels.
3. **Analysis of the filtering scheme performance in imbalanced datasets** (Section 5.3). The third analysis investigates the effect of noise filtering in the most imbalanced datasets. Since each example is now preprocessed by multiple filters, the minority class could be impaired and reduced even further. This last analysis identifies how the minority classes are affected by the decomposition scheme, by monitoring the IR values before and after the employment of the decomposition.

The Wilcoxon signed-rank statistical test [48–50] was applied in the first and second analysis to compare the predictive performance of OVO against the original multi-class approach. The R^+ (sum of the rankings for the positive differences), R^- (sum of the rankings for the negative differences) and p-values in this test are obtained to confirm the results at a confidence level of 95% [48]. High R^+ values with low R^- values indicate a superior performance of the OVO decomposition strategy, while the opposite behavior indicates better results for the original multi-class approach. Next section presents these experimental results.

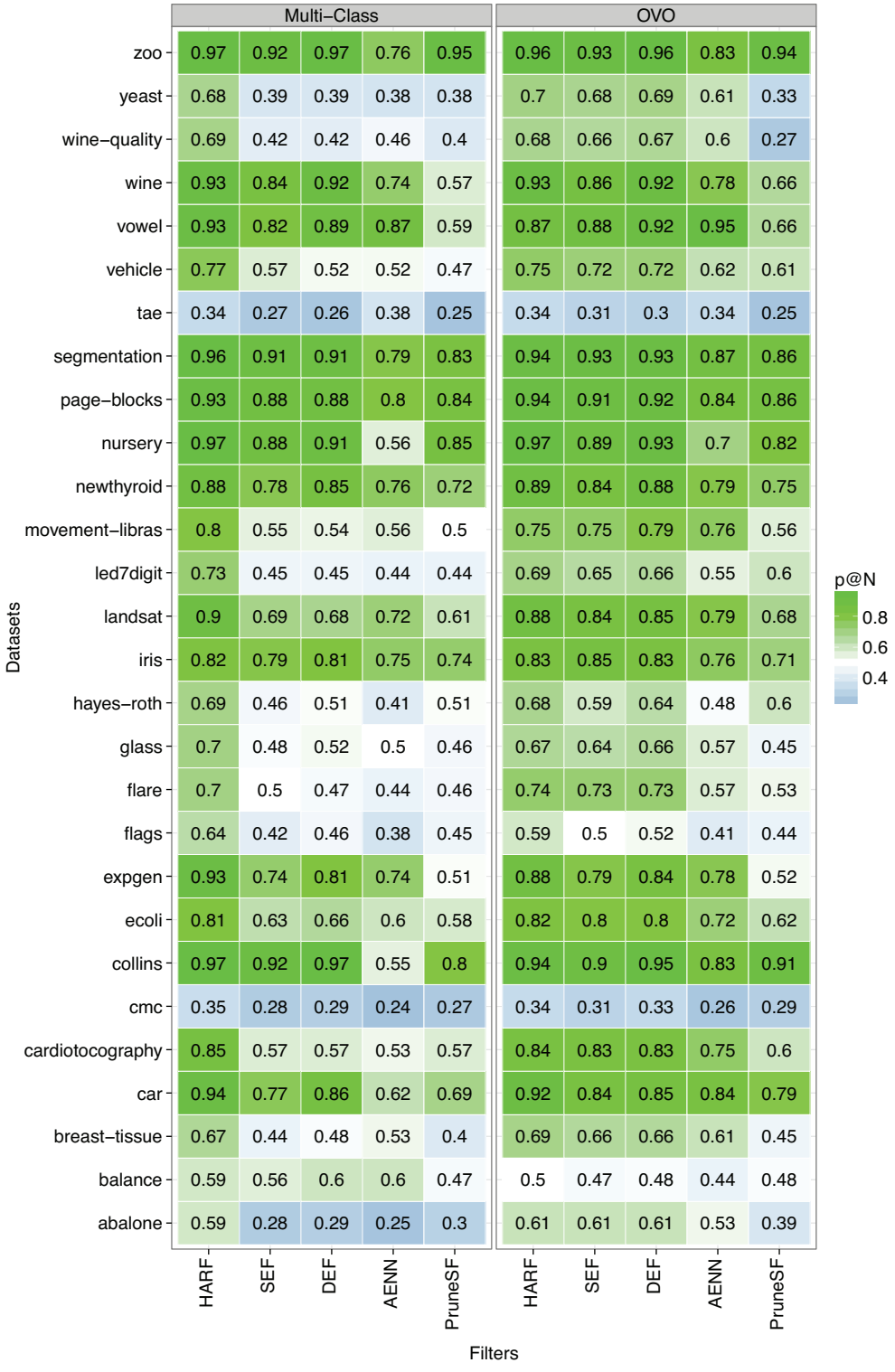


Fig. 3. *p@N* of filters for each dataset and type of noise filtering scheme.

5. Experimental results

This section presents the experimental results obtained in this study. Section 5.1 reports the average *p@N* values of the decomposition and original multi-class strategies and the average ranking of each noise filter. Section 5.2 reports the performance of the filters at each noise level. Finally, Section 5.3 presents how the filters behave in imbalanced datasets.

5.1. *Average* p@N *performance*

This analysis considers the average predictive performance of each filter, independently of the noise level introduced in the data. Therefore, this analysis considers the average results for all noisy versions of the datasets, despite their noise levels. The average *p@N* values obtained by the filters in the identification of the artificial noise inserted are shown in the *heatmap* of Fig. 3. There are two groups of

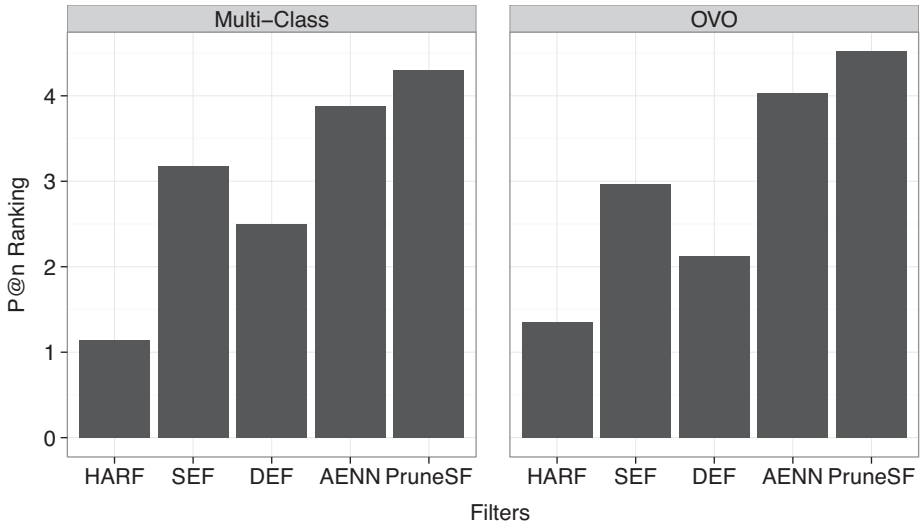


Fig. 4. Ranking of the *p@N* average for each filter in each filtering strategy.

Table 2
The *p*-values of Wilcoxon signed-rank test for each filter.

Filter	R^+	R^-	*p*-value
HARF	108	292	$2.89E-02$
SEF	387	19	$2.92E-05$
DEF	379	26	$6.29E-05$
AENN	378	28	$7.04E-05$
PruneSF	355.5	49.5	$4.84E-04$

columns in this figure, one for each type of strategy: (1) using the original multi-class datasets and (2) using the datasets decomposed by the OVO strategy. Each column represents one filter, while each row corresponds to a specific dataset. While higher *p@N* values are colored in green scale, lower *p@N* score levels are colored in blue scale.

It is possible to observe in Fig. 3 that higher *p@N* values were obtained for the datasets *car, collins, expgen, iris, landsat, newthyroid, nursery, page-blocks, segmentation, vowel, wine* and *zoo*. On the other hand, lower *p@N* values were verified for the datasets *abalone, breast-tissue, cmc, flags, flare, led7digit, tae, wine-quality* and *yeast*. The other datasets had intermediate predictive performance, with *p@N* values ranking between 0.5 and 0.7 for almost all the filters.

In all datasets, for at least one filter, the OVO decomposition was able to improve the filtering performance when compared to the original multi-class scheme. Most of the improvements were obtained for the SEF, DEF, AENN and PruneSF techniques. The HARF filter had an increased performance for some specific cases, while it maintained or decreased its performance for others.

The performance of the OVO strategy in the datasets with higher *p@N* values was better or at least similar to that of the original multi-class strategy. The highest improvements were obtained for datasets with intermediate and low *p@N* values, such as *yeast, wine-quality, vehicle, movement-libras, cardiotocography* and *breast-tissue*. In cases where the multi-class scheme had a low *p@N*, like *cmc* and *abalone*, the OVO decomposition was able to improve the performance further.

Table 2 shows the results of the statistical comparison between the OVO and the original multi-class filtering strategies. The R^+, R^- and *p*-values obtained in the Wilcoxon's test for each filter are shown. At 95% of confidence level, there are significant differences for all filters. OVO was superior in all of the tests, except for the HARF filter, where the multi-class strategy was superior. Next section analyzes these results further, by separating them according to the noise level.

Fig. 4 presents the average ranking of the filters concerning the *p@N* values reported in all datasets. The groups of columns correspond to the multi-class and OVO strategies. While each column represents one filter, the *y*-axis corresponds to the ranking average. Better filters will show a lower ranking.

It is possible to observe in Fig. 4 that the HARF technique was the best performing filter for both OVO and multi-class scenarios. DEF comes next, followed by SEF. AENN and PruneSF were the worst performing filters in both filtering schemes. It is interesting to notice that the order of the filters according to their overall *p@N* performance was maintained after the application of the OVO decomposition. The ensembles DEF and SEF seem to have benefited more from the OVO decomposition according to this analysis, since their ranking was improved after the application of the decomposition.

5.2. Performance per noise level

The previous analysis on the average *p@N* performance hinders the behavior of the techniques for specific noise levels. Fig. 5 shows the difference in the predictive performance achieved by the multi-class and OVO strategies in each dataset, for each noise level. The *x*-axis represents the noise levels while the *y*-axis corresponds to improvements or decreases of *p@N* achieved when using the OVO strategy. HARF is shown by black dots, SEF by red triangles, DEF by blue squares, AENN by green crosses and PruneSF by purple dashed squares. The gray area in the plots highlights improvements achieved by the filters when using the OVO strategy with respect to not performing any decomposition.

For several datasets, it is possible to notice improvements of the *p@N* performance achieved by the OVO decomposition. For SEF, DEF and PruneSF we have a negative slope (as the noise level increases, the performance decreases), with more improvements of performance for low noise rates like 5, 10 and 20%. For AENN we have few cases with positive slope (better results for higher noise levels) and improvements of performance for high noise levels like 40%. For the HARF filter, the predictive performance was similar to the performance of the multi-class strategy or impaired, but there are also improvements, in few cases, for high noise levels.

In general, for all datasets there are at least three filters with improved performance. While SEF, DEF and PruneSF had their performance improved for low noise levels, AENN showed improved results for high noise levels. The HARF filter had improved performance for high noise levels, but mostly it was not significant. There is a specific dataset where most of the filters achieved a low performance when

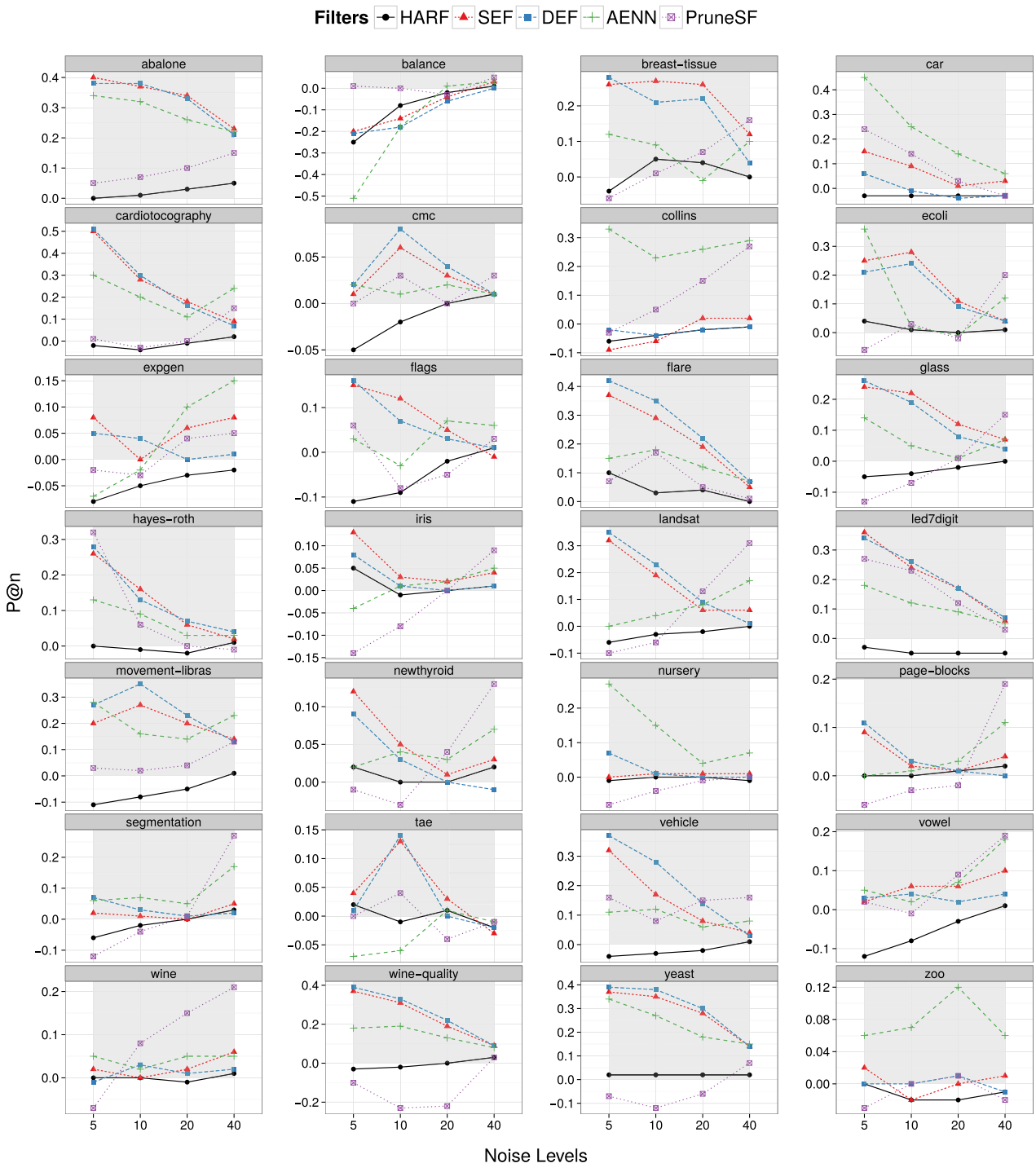


Fig. 5. Differences of *p@N* values between the OVO and the multi-class strategies.

OVO was employed: *balance*. In the datasets *collins, expgen, flags* and *tae* the results were impaired for specific noise levels. Except for the dataset *expgen*, most of the previous datasets are imbalanced.

Table 3 shows for different noise levels the R^+, R^- and *p*-values of the Wilcoxon's test comparing the pair OVO vs multi-class, respectively. In most of the cases, there are statistical differences in favor of OVO. However, there are no statistical differences between some specific noise filters for some noise levels: for HARF there is no statistical difference at 20 and 40% of noise level and for PruneSF this happens for 5 and 10% of noise level. The multi-class strategy was superior to OVO only when using the HARF filter for 5 and 10% of noise level.

Considering Table 3 and the results illustrated in Figs. 3 and 5, the OVO strategy was able to improve the *p@N* values for the SEF, DEF and AENN filters. The same happened with the PruneSF filter in almost all noise levels but without statistical difference for some specific noise levels. For the HARF filter, the multi-class strategy was superior, except for 40% of noise level.

5.3. Filtering noise in imbalanced datasets

The use of the OVO decomposition strategy improved the performance of noise filters for several multi-class datasets, even those with high IR. Nonetheless, it is important to keep all safe examples

Table 3
p-values of Wilcoxon signed-rank test for each filter and noise level.

	HARF			SEF			DEF			AENN			PruneSF		
Rates	R^+	R^-	*p*-value	R^+	R^-	*p*-value	R^+	R^-	*p*-value	R^+	R^-	*p*-value	R^+	R^-	*p*-value
5%	87.5	303.5	$1.49E-2$	380.5	24.5	$6.61E-05$	381.5	23.5	$5.42E-5$	346	57	$1.37E-3$	192.5	210.5	$8.68E-1$
10%	76.0	320	$5.27E-3$	374	29	$1.13E-04$	375.5	29.5	$9.88E-5$	358.5	47.5	$4.13E-4$	237.5	165.5	$3.94E-1$
20%	116.5	268.5	$9.23E-2$	391	12	$2.75E-05$	352	39	$5.54E-4$	400	6	$7.51E-6$	307.5	88.5	$1.09E-2$
40%	263.5	132.5	$1.87E-1$	395.5	10.5	$1.21E-05$	347.5	52.5	$7.76E-4$	404.5	1.5	$4.68E-6$	386.5	18.5	$2.87E-5$

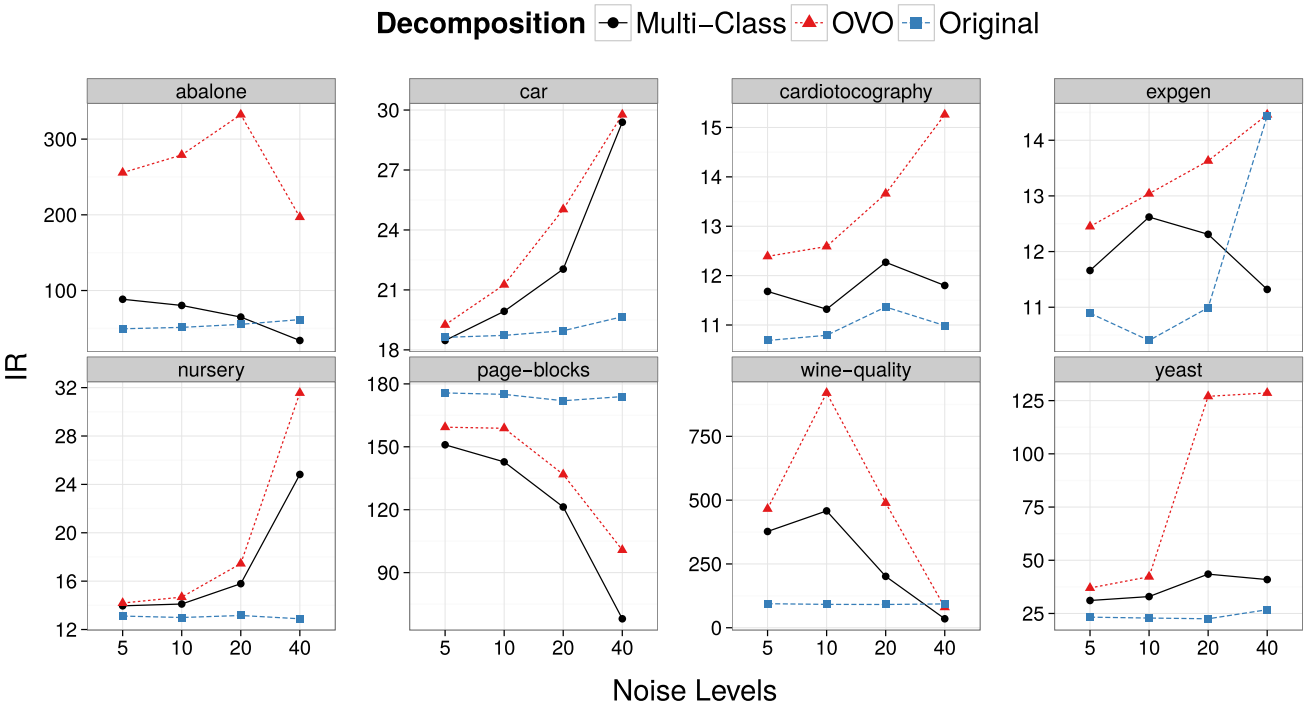


Fig. 6. IR achieved by the OVO and multi-class strategies in datasets with $IR \geq 10$.

from the minority class, avoiding their removal by the filters. Fig. 6 shows the IR of the datasets filtered by the multi-class and OVO strategies, for each noise level. The IR values for the original datasets when all the noisy examples are correctly identified are also shown. These graphs are plotted only for datasets with a high IR ($IR \geq 10$) and they consider the best performing filters in the multi-class setting (the HARF filter, in which the multi-class setting generally outperforms decomposition). The *x*-axis represents the noise levels and the *y*-axis corresponds to the *IR* values. The multi-class IR and the OVO IR are represented with black circles and red triangles, respectively, while the IR for the original noisy datasets are illustrated with blue squares. These plots show how preprocessing affects the minority classes. The results for perfect noise preprocessing (original) remain the same for different noise rates, since a uniform random noise imputation method was used, which tends to affect all classes uniformly.

The HARF filter obtained the best performance in these datasets. For the *car, cardiotocography* and *expgen* datasets, the best *p@N* performance was obtained by the multi-class strategy. For the *abalone, page-blocks, wine-quality* and *yeast* datasets, which mainly have numerical attributes, the OVO strategy presented the best *p@N* performance. For the *nursery* dataset, the two strategies had the same *p@N* performance.

Regarding the IR values, both multi-class and OVO filtering schemes tend to produce more imbalanced datasets compared with if perfect filtering, except in the *page-blocks* dataset. They both seem to have eliminated safe examples from the minority classes. Overall, the multi-class filtering strategy always produces more balanced datasets than the OVO strategy. The OVO strategy presents the largest changes of IR for the datasets *abalone*, for all noise levels, *wine-quality* for 10% of noise level, and *yeast* for 20 and 40% of noise level. Even when improved the performance, OVO always increased the imbalance in the datasets and tended to remove more minority class examples. This is a harmful effect that can be due to the several filter application to each example in the OVO strategy, increasing its probability of being labeled as noisy. Nonetheless, the increase of IR seems to be a harmful effect of noise pre-processing, despite the filtering scheme employed. Some strategies can minimize these effects, such as weight the NDP of an example by the proportion of training examples from its class. This could decrease the reduction of the minority class examples by the filters.

6. Conclusion

This paper investigated the performance of five well-known noise filtering techniques when multi-class datasets are decomposed using the OVO decomposition strategy. Several benchmark public datasets were used in the experiments and different levels of artificially imputed noise data were considered. In Sáez et al. [10], the OVO strategy improved the robustness of the classifiers in the presence of noise when dealing with multi-class datasets. In this study, we reinforced the benefits of the same strategy under a different perspective: to use the OVO strategy to improve the performance of various preprocessing techniques in label noise identification.

The OVO decomposition presented a better performance than the multi-class strategy in almost all analysis carried out. This may

have occurred because, while in the original multi-class dataset the information has a complex structure representation, the simpler structure of the binary subproblems after the OVO decomposition helps the identification of the noisy data. Considering all results obtained by the OVO strategy, the SEF, DEF, AENN and PruneSF filters had their *p@N* values improved for almost all datasets. For the HARF filter, the multi-class strategy remained the best for most of the noise levels.

Results from a separate analysis in imbalanced datasets showed that filtering tends to increase the imbalance, regardless of the filtering scheme employed. OVO intensified this effect. These results should be further investigated, taking into account the performance of the filters in the individual classes. This analysis also reinforced the importance of the presence of a domain specialist to analyze the results, even when the noise filters show a good performance in the overall noise identification task.

As future work, we would like to evaluate other decomposition strategies and study the use and combination of additional filters. This can improve the low performance seen for some datasets. For example, the use of dynamic OVO strategies [51,52] to improve the noise detection is a promising alternative. We would like also to analyze whether the multi-class datasets have an intrinsic noise level, which was not considered in the reported experiments because it is usually not possible to guarantee that an example is noisy. We also plan to investigate other strategies able to improve the filters performance in imbalanced data, specially for the minority classes. It is also relevant to develop a method able to automatically set the threshold for the NDP value to define whether an example is noisy. Possible alternatives are to use complexity measures or cumulative suns of probabilities of NDP until an abrupt change in percentages obtained.

Acknowledgment

The authors would like to thank FAPESP, CNPq and CAPES for their financial support. F. Herrera was supported by the National Project TIN2014-57251-P, and also by the Regional Projects P10-TIC-06858 and P11-TIC-7765. J.A. Sáez was supported by EC under FP7, Coordination and Support Action, grant agreement number 316097, ENGINE European Research Centre of Network Intelligence for Innovation Enhancement (http://engine.pwr.wroc.pl/).

References

[1] X. Zhu, X. Wu, Class noise vs. attribute noise: a quantitative study, Artif. Intell. Rev. 22 (3) (2004) 177–210.
[2] B. Frenay, M. Verleysen, Classification in the presence of label noise: a survey, IEEE Trans. Neural Netw. Learn. Syst. 25 (5) (2014) 845–869.
[3] S. García, J. Luengo, F. Herrera, Data Preprocessing in Data Mining, Springer, 2015.
[4] R.Y. Toledo, Y.C. Mota, L. Martínez, Correcting noisy ratings in collaborative recommender systems, Knowl.-Based Syst. 76 (2015) 96–108.
[5] C.E. Brodley, M.A. Friedl, Identifying mislabeled training data, J. Artif. Intell. Res. 11 (1999) 131–167.
[6] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, Tackling the problem of classification with noisy data using multiple classifier systems: analysis of the performance and robustness, Inf. Sci. 247 (2013) 1–20.
[7] A.C. Lorena, A.C. Carvalho, J.a.M. Gama, A review on the combination of binary classifiers in multiclass problems, Artif. Intell. Rev. 30 (1-4) (2008) 19–37.
[8] V. Vapnik, Statistical Learning Theory, Wiley, 1998.
[9] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recognit. 44 (2011) 1761–1776.
[10] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition, Knowl. Inf. Syst. 38 (1) (2014) 179–206.
[11] D.L. Wilson, Asymtoptic properties of nearest neighbor rules using edited data, IEEE Trans. Syst., Man Cybernet. SMC 2 (3) (1972) 408–421.
[12] X. Zhu, X. Wu, Q. Chen, Eliminating class noise in large datasets, in: Proceeding of the Twentieth International Conference on Machine Learning, 2003, pp. 920–927.
[13] B. Sluban, D. Gamberger, N. Lavrac, Ensemble-based noise detection: noise ranking and visual performance evaluation, Data Mining Knowl. Discov. 28 (2) (2014) 265–303.
[14] L.P.F. Garcia, A.C. Lorena, A.C.P.L.F. Carvalho, A study on class noise detection and elimination, in: A.C. Lorena, C.E. Thomaz, A.T.R. Pozo (Eds.), Proceedings of the 2012 Brazilian Symposium on Neural Networks (SBRN), IEEE, 2012, pp. 13–18.
[15] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, INFFC: an iterative class noise filter based on the fusion of classifiers with noise sensitivity control, Inf. Fusion 27 (2016) 19–32.
[16] I. Tomek, An experiment with the edited nearest-neighbor rule, IEEE Trans. Syst., Man Cybernet SMC. 6 (6) (1976) 448–452.
[17] L.P. Garcia, A.C. de Carvalho, A.C. Lorena, Effect of label noise in the complexity of classification problems, Neurocomputing 160 (2015) 108–119.
[18] D. Gamberger, N. Lavrac, C. Groselj, Experiments with noise filtering in a medical domain, in: I. Bratko, S. Dzeroski (Eds.), ICML, Morgan Kaufmann, 1999, pp. 143–151.
[19] S. Verbaeten, A.V. Assche, Ensemble methods for noise elimination in classification problems, in: T. Windeatt, F. Roli (Eds.), Multiple Classifier Systems, Lecture Notes in Computer Science, 2709, Springer Berlin Heidelberg, 2003, pp. 317–325.
[20] B. Sluban, D. Gamberger, N. Lavrac, Advances in class noise detection, in: H. Coelho, R. Studer, M. Wooldridge (Eds.), ECAI, Frontiers in Artificial Intelligence and Applications, 215, IOS Press, 2010, pp. 1105–1106.
[21] A. Sahu, D.W. Apley, G.C. Runger, Feature selection for noisy variation patterns using kernel principal component analysis, Knowl.-Based Syst. 72 (2014) 37–47.
[22] J.A. Sáez, J. Luengo, J. Stefanowski, F. Herrera, Smoteipf: addressing the noisy and borderline examples problem in imbalanced classification by a re-sampling method with filtering, Inf. Sci. 291 (2015) 184–203.
[23] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, L.I. Kuncheva, Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl.-Based Syst. 85 (2015) 96–111.
[24] X. Wu, X. Zhu, Mining with noise knowledge: error-aware data mining, IEEE Trans.n Syst., Man, Cybernet. - Part A: Syst. Hum. 38 (4) (2008) 917–932.
[25] J.A. Sáez, J. Luengo, F. Herrera, Predicting noise filtering efficacy with data complexity measures for nearest neighbor classification, Pattern Recognit. 46 (1) (2013) 355–364.
[26] D.R. Wilson, T.R. Martinez, Reduction techniques for instance-based learning algorithms, Mach. Learn. 38 (3) (2000) 257–286.
[27] L. Breiman, Random forests, Mach. Learn. 45 (1) (2001) 5–32.
[28] J.R. Quinlan, Induction of decision trees, Mach. Learn. 1 (1) (1986) 81–106.
[29] T.M. Mitchell, Machine learning, McGraw Hill Series in Computer Science, 1st, McGraw-Hill, 1997.
[30] L. Liu, Q. Liang, A high-performing comprehensive learning algorithm for text classification without pre-labeled training set, Knowl. Inf. Syst. 29 (3) (2011) 727–738.
[31] P. Melin, J. Amezcua, F. Valdez, O. Castillo, A new neural network model based on the LVQ algorithm for multi-class classification of arrhythmias, Inf. Sci. 279 (2014) 483–497.
[32] G. Kou, Y. Peng, Z. Chen, Y. Shi, Multiple criteria mathematical programming for multi-class classification and application in network intrusion detection, Inf. Sci. 179 (4) (2009) 371–381.
[33] A. Passerini, M. Pontil, P. Frasconi, New results on error correcting output codes of kernel machines, IEEE Trans. Neural Netw. (2004) 45–54.
[34] J. Fürnkranz, Round Robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.
[35] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recognit. 43 (1) (2010) 128–142.
[36] T.-F. Wu, C.-J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.
[37] J.C. Platt, N. Cristianini, J. Shawe-taylor, Large margin dags for multiclass classification, in: Advances in Neural Information Processing Systems, MIT Press, 2000, pp. 547–553.
[38] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[39] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten, The weka data mining software: an update, SIGKDD Explor. Newsl. 11 (1) (2009) 10–18.
[40] J. Alcalá-Fdez, L. Sánchez, S. García, M.J.D. 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, Softw. Comput. 13 (3) (2008) 307–318.
[41] C.-C. Chang, C.-J. Lin, Libsvm: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (3) (2011) 27:1–27:27.
[42] Q. Li, Y. Mao, A review of boosting methods for imbalanced data classification, Pattern Anal. Appl. 17 (4) (2014) 679–693.
[43] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Mult.-Valued Logic Soft Comput. 17 (2-3) (2011) 255–287.
[44] K. Bache, M. Lichman, UCI machine learning repository, 2013, http://archive.ics.uci.edu/ml, (accessed December 2014).
[45] A. Tanwani, M. Farooq, Classification potential vs. classification accuracy: a comprehensive study of evolutionary algorithms with biomedical datasets, in: J. Bacardit, W. Browne, J. Drugowitsch, E. Bernad-Mansilla, M. Butz (Eds.), Learning Classifier Systems, 6471, Springer Berlin Heidelberg, 2010, pp. 127–144.
[46] C.-M. Teng, Correcting noisy data, in: I. Bratko, S. Dzeroski (Eds.), ICML, Morgan Kaufmann, 1999, pp. 239–248.
[47] E. Schubert, R. Wojdanowski, A. Zimek, H.-P. Kriegel, On Evaluation of Outlier Rankings and Outlier Scores, Anaheim, CA, 2012, pp. 1047–1058.
[48] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[49] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.
[50] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18.
[51] M. Galar, A. Fernández, E. Barrenechea, F. Herrera, DRCW-OVO: distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems, Pattern Recognit. 48 (1) (2015) 28–42.
[52] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for One-vs-One strategy: avoiding non-competent classifiers, Pattern Recognit. 46 (12) (2013) 3412–3424.