Knowl Inf Syst (2014) 38:179–206
DOI 10.1007/s10115-012-0570-1

REGULAR PAPER

Analyzing the presence of noise in multi-class problems: alleviating its influence with the One-vs-One decomposition

José A. Sáez · Mikel Galar · Julián Luengo · Francisco Herrera

Received: 7 February 2012 / Revised: 12 August 2012 / Accepted: 6 October 2012 /
Published online: 6 November 2012

Abstract The presence of noise in data is a common problem that produces several negative consequences in classification problems. In multi-class problems, these consequences are aggravated in terms of accuracy, building time, and complexity of the classifiers. In these cases, an interesting approach to reduce the effect of noise is to decompose the problem into several binary subproblems, reducing the complexity and, consequently, dividing the effects caused by noise into each of these subproblems. This paper analyzes the usage of decomposition strategies, and more specifically the One-vs-One scheme, to deal with noisy multi-class datasets. In order to investigate whether the decomposition is able to reduce the effect of noise or not, a large number of datasets are created introducing different levels and types of noise, as suggested in the literature. Several well-known classification algorithms, with or without decomposition, are trained on them in order to check when decomposition is advantageous. The results obtained show that methods using the One-vs-One strategy lead to better performances and more robust classifiers when dealing with noisy data, especially with the most disruptive noise schemes.

Keywords Noisy data · Class noise · Attribute noise · One-vs-One · Decomposition strategies · Ensembles · Classification

J. A. Sáez (✉) · F. Herrera
Department of Computer Science and Artificial Intelligence, University of Granada, CITIC-UGR, 18071 Granada, Spain
e-mail: smja@decsai.ugr.es

F. Herrera
e-mail: herrera@decsai.ugr.es

M. Galar
Department of Automática y Computación, Universidad Pública de Navarra, 31006 Pamplona, Spain
e-mail: mikel.galar@unavarra.es

J. Luengo
Department of Civil Engineering, LSI, University of Burgos, 09006 Burgos, Spain
e-mail: jluengo@ubu.es

1 Introduction

Any classification problem [14,49] consists of m training examples, characterized by n attributes A_i, $i = 1, \ldots, n$ that are either numerical or categorical, with $\mathbb{D}_i$ their corresponding domains. Thus, an example $\mathbf{x}$ is represented as an n-dimensional attribute vector

$$\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{D} = \mathbb{D}_1 \times \cdots \times \mathbb{D}_n.$$

Each of these examples is labeled with one out of the M possible classes $\mathbb{L} = \{\lambda_1, \ldots, \lambda_M\}$. Many current real-world classification problems, such as the classification of cancer tissues [5] or the recognition of business documents [36], must distinguish between more than two classes, that is, $M > 2$. These problems are formally known as multi-class classification problems.

Classification algorithms aim to extract the implicit knowledge from previously known labeled examples of the problem creating a model, called a classifier, which should be capable of predicting the class for new unobserved examples. For this reason, the classification accuracy of a classifier is directly influenced by the quality of the training data used. Data quality depends on several components [50], for example, the source and the input procedure, inherently subject to errors. Real-world datasets usually contain corrupted data that may hinder the interpretations, decisions, and therefore, the classifiers built from that data.

Usually, the more classes in a problem, the more complex it is. In multi-class learning, the generated classifier must be able to separate the data into more than a pair of classes, which increases the chances of incorrect classifications (in a two-class balanced problem, the probability of a correct random classification is $1/2$, whereas in a multi-class problem it is $1/M$). Furthermore, in problems affected by noise, the boundaries, the separability of the classes, and therefore, the prediction capabilities of the classifiers may be severely hindered.

Given the loss of accuracy produced by noise, the need of techniques to deal with it has been proved in previous works [9,23,56]. In the specialized literature, two ways have been proposed in order to mitigate the effects produced by noise:

1. Adaptation of the algorithms to properly handle the noise [11,42]. These methods are known as *robust learners* and they are characterized by being less influenced by noisy data.
2. Preprocessing of the datasets aiming to remove or correct the noisy examples [8,18].

However, even though both techniques can provide good results, drawbacks exist. The former depends on the classification algorithm, and therefore, the same result is not directly extensible to other learning algorithms, since the benefit comes from the adaptation itself. Moreover, this approach requires to change an existing method, which neither is always possible nor easy to develop. However, the latter requires the usage of a previous preprocessing step, which is usually time-consuming. Furthermore, these methods are only designed to detect an specific type of noise and hence, the resulting data might not be perfect [53]. For these reasons, it is important to investigate other mechanisms, which could lead to decrease the effects caused by noise without neither needing to adapt each specific algorithm nor having to make assumptions about the type and level of noise present in the data.

When dealing with multi-class problems, several works [6,27] have demonstrated that decomposing the original problem into several binary subproblems is an easy, yet accurate way to reduce their complexity. These techniques are referred to as binary decomposition strategies [32]. The most studied schemes in the literature are: *One-vs-One* (OVO) [27], which trains a classifier to distinguish between each pair of classes, and *One-vs-All* (OVA) [6], which trains a classifier to distinguish each class from all other classes. Both strategies

can be encoded within the error-correcting output codes framework [4,13]. However, none of these works provide any theoretical nor empirical results supporting the common assumption that supposes a better behavior against noise of decomposition techniques (than not using decomposition). Neither they show what type of noise is better handled by decomposition techniques.

On this account, this paper analyzes the usage of the OVO strategy, which generally outstands over OVA [15,16,24,43,45], and checks its suitability with noisy training data. It should be mentioned that, in real situations, the existence of noise in the datasets is usually unknown—therefore, neither the type nor the quantity of noise in the dataset can be known or supposed a priori. Hence, tools which are able to manage the presence of noise in the datasets, despite its type or quantity (or unexistence), are of great interest. If the OVO strategy (which is a simple yet effective methology when clean[1] datasets are considered) is also able to properly (better than the baseline non-OVO version) handle the noise, its usage could be recommended in spite of the presence of noise and without taking into account its type. Furthermore, this strategy can be used with any of the existing classifiers which are able to deal with two-class problems. Therefore, the problems of algorithm level modifications and preprocessing techniques could be avoided, and if desired, they could also be combined.

In order to carry out the analysis, a thorough empirical study will be developed considering several well-known learning algorithms having a very different behavior with noisy data: two rule-based systems, which are considered robust to noise—C4.5 [42] and *Repeated Incremental Pruning to Produce Error Reduction* (RIPPER) [11]—and an instance-based learning method, which is considered very noise-sensitive—*k-Nearest Neighbors* (*k*-NN) [35]—will be studied. Even though the theoretical robustness of these methods has been previously studied [29,40,41], there is a lack of empirical studies analyzing the real behavior of such methods when dealing with noisy data, particularly if the OVO decomposition is used. Considering two different noise categories, class and attribute noise, 800 datasets will be created [56]. Several noise schemes presented in the literature will be used to introduce these types of noise [46,56–58] and a large number of noise levels—from 5 to 50 %, by increments of 5 %—will be also studied. The differences between the OVO and non-OVO (baseline) classifiers will be analyzed through an analysis of both the accuracy and the robustness achieved on these datasets. The results obtained will be contrasted using the proper statistical tests, as recommended in the specialized literature [12].

The experimental framework stated will allow us to extract a series of conclusions on the effect of noise in multi-class problems and the usage of OVO in this scenario. We will concrete the types of noise—class (random/pair) or attribute noise(random/Gaussian)—that are more detrimental for the classifier performance and those where OVO provides a higher advantage. We will also determine in which extent OVO helps robust and noise-sensitive learners to deal with noisy data and the reasons of the increase of the robustness of such methods when using OVO. All these conclusions will be presented in Sect. 7.

A web page with all the complementary material associated with this paper is available at http://sci2s.ugr.es/ovo_noise, including the basic information of this paper, all the datasets created, and the complete results obtained for each classification algorithm.

The rest of this paper is organized as follows. Section 2 presents an introduction to classification with noisy data. Section 3 is devoted to the motivations for the usage of binary decomposition strategies in multi-class classification problems and recalls the OVO decomposition scheme. Next, Sect. 4 describes the experimental framework. Section 5 includes the analysis

[1] We refer to clean and noise-free datasets to the original datasets without additional induced noise, despite they might have noise, but it is not quantifiable, and hence it cannot be used to evaluate the robustness of the methods against noise.

of the results obtained by the classifiers on data with class noise, whereas Sect. 6 focuses on attribute noise. Section 7 provides a summary including the conclusions that can be extracted from the analysis of the results. Finally, Sect. 8 presents the concluding remarks.

2 Classification with noisy data

Real-world data are never perfect and often suffers from corruptions that may hinder the analysis of the data and their interpretations, that is, the models extracted and hence the decisions made on their basis. In classification, noise can negatively affect the system performance in terms of classification accuracy, building time, size, and interpretability of the classifier [55,56]. The presence of noise in the data may affect the intrinsic characteristics of a classification problem, since these corruptions could introduce new properties in the problem domain. For example, noise can lead to the creation of small clusters of examples of a particular class in areas of the domain corresponding to another class, or it can cause the disappearance of examples located in key areas within a specific class. The boundaries of the classes and the overlapping between them are also factors that can be affected as a consequence of noise. All these alterations difficult the knowledge extraction from the data and spoil the models obtained using that noisy data when they are compared to the models learned from clean data, which represent the real implicit knowledge of the problem [56].

The quality of a dataset is determined by a large number of components [50]. Among them, the class labels and the attribute values are two sources influencing the quality of a classification dataset. The quality of the class labels refers to whether the class of each example is correctly assigned; the quality of the attributes refers to the capability to characterize the examples for classification purposes. Two types of noise in a given dataset can be distinguished based on these two information sources [52]:

1. **Class noise** (or labeling errors). It occurs when an example is incorrectly labeled. Class noise can be attributed to several causes, such as subjectivity during the labeling process, data entry errors, or inadequacy of the information used to label each example. Two types of class noise can be distinguished:
   - *Contradictory examples.* There are duplicate examples in the dataset having different class labels [21].
   - *Misclassifications.* Examples are labeled with other class label different from the real one [57].
2. **Attribute noise**. It refers to corruptions in the values of one or more attributes. Examples of attribute noise are: erroneous attribute values, missing or unknown attribute values, and incomplete attributes or "do not care" values.

In this paper, class noise refers to misclassifications, whereas attribute noise refers to the erroneous attribute values, because they are the most common in real-world data [56]. Furthermore, erroneous attribute values, unlike other types of attribute noise, such as missing values [33] (which are easily detectable), have been less studied in the literature.

Treating class and attribute noise as corruptions of the class labels and attribute values, respectively, has been also considered in other works in the literature [37,56]; for instance, in [56], the authors reached a series of interesting conclusions, showing that attribute noise is more harmful than class noise or that eliminating or correcting examples in datasets with class and attribute noise, respectively, may improve classifier performance. They also showed that attribute noise is more harmful in those attributes highly correlated with the class labels.

In [37], the authors checked the robustness of methods from different paradigms, such as probabilistic classifiers, decision trees, instance-based learners or support vector machines, studying the possible causes of their behaviors.

However, most of the works found in the literature are only focused on class noise. In [7], the problem of multi-class classification in the presence of labeling errors was studied. The authors proposed a generative multi-class classifier to learn with labeling errors, which extends the multi-class quadratic normal discriminant analysis by a model of the mislabeling process. They demonstrated the benefits of this approach in terms of parameter recovery as well as improved classification performance. In [22], the problems caused by labeling errors occurring far from the decision boundaries in multi-class Gaussian process classifiers were studied. The authors proposed a robust multi-class Gaussian process classifier, introducing binary latent variables that indicate when an example is mislabeled. Similarly, the effect of mislabeled samples appearing in gene expression profiles was studied in [54]. A detection method for these samples was proposed, which take advantage of the measuring effect of data perturbations based on the support vector machine regression model. They also proposed three algorithms based on this index to detect mislabeled samples. An important common characteristic of these works, also considered in this paper, is that the suitability of the proposals was evaluated on both real-world and synthetic or noisy-modified real-world datasets, where the noise can be somehow quantified.

In order to model class and attribute noise, we consider four different synthetic noise schemes found in the literature, in such a way that we can simulate the behavior of the classifiers in the presence of noise:

1. **Class noise** usually occurs on the boundaries of the classes, where the examples have similar characteristics—although it might occur on any other areas of the domain. In this paper, class noise is introduced using a *random class noise scheme* [46] (randomly corrupting the class labels of the examples) and a *pairwise class noise scheme* (labeling examples of the majority class with the second majority class) [56,57]. Considering these two schemes, the similarities between any pair of classes and only between the two majority classes are simulated, respectively.
2. **Attribute noise** can proceed from several sources, such as transmission constraints, faults in sensor devices, irregularities in sampling, and transcription errors [47]. The erroneous attribute values can be totally unpredictable, that is, random, or they can imply a low variation with respect to the correct value. We use a *random attribute noise scheme* [56,58] and a *Gaussian attribute noise scheme* [44] in order to simulate each one of the possibilities, respectively. We introduce attribute noise in accordance with the hypothesis that interactions between attributes are weak [56]. As a result, the noise introduced into each attribute has a low correlation with the noise introduced into the rest of the attributes.

Robustness is the capability of an algorithm to build models that are insensitive to data corruptions and suffer less from the impact of noise [25]. Thus, a classification algorithm is said to be more robust than other one if the former builds classifiers which are less influenced by noise than the latter, that is, more robust. In order to analyze the degree of robustness of the classifiers in the presence of noise, we will compare the performance of the classifiers learned with the original (without induced noise) dataset with the performance of the classifiers learned using the noisy dataset. Therefore, those classifiers learned from noisy datasets being more similar (in terms of results) to the noise-free classifiers will be the most robust ones.

3 Addressing multi-class classification problems by decomposition

Multi-class classification problems are frequent in real-world classification tasks. Examples of such problems are the classification of micro-arrays [30], electroencephalogram signals [19] or texts [31], and audio streams [1]. These problems are more general and complex than the special case of two classes, that is, binary classification problems.

In the literature, multi-class classifier learning has been overcome in two different ways [32]: (1) adapting the internal operations of the learning algorithm and (2) decomposing the multi-class problem into a set of easier to solve binary subproblems. The former embeds the management of the multiple classes in the algorithm, whereas the latter aims to reduce the complexity of the original problem by decomposing it into simpler binary subproblems. In such a way, any binary classifier learning algorithm can be used as base learner, without needing to adapt its learning procedure. The first alternative may be a very complex issue [38]; therefore, it is common to use the decomposition alternative when the algorithms are not easily adaptable, that is, suppor vector machines [48], but also when adaptations exist, since its benefits have been proved [16].

3.1 Decomposition strategies for multi-class problems

Several motivations for the usage of binary decomposition strategies in multi-class classification problems can be found in the literature [15,16,24,43]:

- The separation of the classes becomes easier (less complex), since less classes are considered in each subproblem [15,34]. For example, in [28], the classes in a digit recognition problem were shown to be linearly separable when considered in pairs. A simpler alternative than learning a unique nonlinear classifier to separate all classes simultaneously.
- Classification algorithms, whose extension to multi-class problems is not easy, can address multi-class problems using decomposition techniques [15].
- In [39], the advantages of the usage of decomposition was pointed out when the classification errors for different classes have distinct costs. The binarization allows the binary classifiers generated to impose preferences for some of the classes.
- Decomposition allows one to easily parallelize the classifier learning, since the binary subproblems are independent and can be solved with different processors.

Dividing a problem into several new subproblems, which are then independently solved, implies the need of a second phase where the outputs of each problem need to be aggregated. Therefore, decomposition includes two steps:

1. *Problem division*. The problem is decomposed into several binary subproblems which are solved by independent binary classifiers, called *base classifiers* [15]. Different decomposition strategies can be found in the literature [32]. The most common one is OVO [27].
2. *Combination of the outputs* [16]. The different outputs of the binary classifiers must be aggregated in order to output the final class prediction. In [16], an exhaustive study comparing different methods to combine the outputs of the base classifiers in the OVO and OVA strategies is developed. Among these combination methods, the weighted voting [26] and the approaches in the framework of probability estimates [51] are highlighted.

This paper focuses the OVO decomposition strategy due to the several advantages shown in the literature with respect to OVA [15,16,24,43]:

– OVO creates simpler borders between classes than OVA.
– OVO generally obtains a higher classification accuracy and a shorter training time than OVA because the new subproblems are easier and smaller.
– OVA has more of a tendency to create imbalanced datasets which can be counterproductive [17,45].
– The application of the OVO strategy is widely extended and most of the software tools considering binarization techniques use it as default [3,10,20].

3.2 One-vs-One decomposition scheme

The OVO decomposition strategy consists of dividing a classification problem with M classes into $M(M-1)/2$ binary subproblems. A classifier is trained for each new subproblem only considering the examples from the training data corresponding to the pair of classes (λ_i, λ_j) with $i < j$ considered.

When a new instance is going to be classified, it is presented to all the binary classifiers. This way, each classifier discriminating between classes λ_i and λ_j provides a confidence degree $r_{ij} \in [0, 1]$ in favor of the former class (and hence, r_{ji} is computed by $1 - r_{ij}$). These outputs are represented by a score matrix R:

$$R = \begin{pmatrix} - & r_{12} & \cdots & r_{1M} \\ r_{21} & - & \cdots & r_{2M} \\ \vdots & & & \vdots \\ r_{M1} & r_{M2} & \cdots & - \end{pmatrix} \tag{1}$$

The final output is derived from the score matrix by different aggregation models. The most used and simplest combination, also considered in the experiments of this paper, is the application of a voting strategy:

$$\text{Class} = \arg\max_{i=1,\ldots,M} \sum_{1 \le j \ne i \le M} s_{ij} \tag{2}$$

where s_{ij} is 1 if $r_{ij} > r_{ji}$ and 0 otherwise. Therefore, the class with the largest number of votes will be predicted. This strategy has proved to be competitive with different classifiers obtaining similar results in comparison with more complex strategies [16].

4 Experimental framework

First, the base datasets used in the experiments are described (Sect. 4.1). Afterward, how the noise is induced into them is explained (Sect. 4.2). The algorithms used as base classifiers and their parameters are presented in Sect. 4.3. Finally, the methodology for the analysis of the results is explained in Sect. 4.4.

4.1 Base datasets

The experimentation is based on twenty real-world multi-class classification problems from the KEEL-dataset repository[2] [2]. Table 1 shows the datasets sorted by the number of classes (#CLA). Moreover, for each dataset, the number of examples (#EXA) and the number of attributes (#ATT), along with the number of real, integer and nominal attributes (R/I/N) are

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

Table 1 Summary description of the classification datasets

Dataset	#CLA	#EXA	#ATT (R/I/N)	Dataset	#CLA	#EXA	#ATT (R/I/N)
Balance	3	625	4 (4/0/0)	Flare	6	1,066	11 (0/0/11)
Contraceptive	3	1,473	9 (0/9/0)	Glass	7	214	9 (9/0/0)
Iris	3	150	4 (4/0/0)	Satimage	7	643	36 (0/36/0)
Newthyroid	3	215	5 (4/1/0)	Segment	7	2,310	19 (19/0/0)
Splice	3	319	60 (0/0/60)	Shuttle	7	2,175	9 (0/9/0)
Thyroid	3	720	21 (6/15/0)	Ecoli	8	336	7 (7/0/0)
Vehicle	4	846	18 (0/18/0)	Led7digit	10	500	7 (7/0/0)
Nursery	5	1,269	8 (0/0/8)	Penbased	10	1,099	16 (0/16/0)
Page-blocks	5	547	10 (4/6/0)	Yeast	10	1,484	8 (8/0/0)
Automobile	6	150	25 (15/0/10)	Vowel	11	990	13 (10/3/0)

presented. Some of the largest datasets (*nursery*, *page-blocks*, *penbased*, *satimage*, *splice*, and *led7digit*) were stratified at 10 % in order to reduce the computational time required for training, given the large amount of executions carried out. For datasets containing missing values (such as *automobile* or *dermatology*), instances with missing values were removed from the dataset before the partitioning.

4.2 Introducing noise into datasets

In the datasets presented in the previous section, the initial amount and type of noise present on them are unknown. Therefore, no assumptions about the base noise type and level can be made. For this reason, these datasets are considered to be noise-free, in the sense that no new noise has been induced. In order to control the amount of noise in each dataset and to check how it affects the classifiers, noise is introduced into each dataset in a supervised manner. Four different noise schemes, which are proposed in the specialized literature, are used in order to introduce a noise level of x % into each dataset. The following procedures are followed in order to induce the different noise schemes:

1. **Introduction of class noise.**
   - **Random class noise** [46]. It supposes that exactly x % of the examples are corrupted. The class labels of these examples are randomly changed by other one out of the M classes.
   - **Pairwise class noise** [56,57] . Being X the majority class and Y the second majority class, an example with the label X has a probability of $x/100$ of being incorrectly labeled as Y.
2. **Introduction of attribute noise.**
   - **Random attribute noise** [56,58]. x % of the values of each attribute in the dataset are corrupted. To corrupt an attribute A_i, approximately x % of the examples in the data set are chosen, and their A_i value is assigned a random value from $\mathbb{D}_i$. A uniform distribution is used either for numerical or nominal attributes.
   - **Gaussian attribute noise** [44]. This scheme is similar to the random attribute noise, but in this case, the A_i values are corrupted adding a random value to them following a Gaussian distribution of $mean = 0$ and $standard\ deviation = (max - min)/5$,

being *max* and *min* the upper and lower limits of $\mathbb{D}_i$, respectively. Nominal attributes are treated as in the case of the random attribute noise.

In order to create a noisy dataset from an original noise-free dataset, the noise is introduced into the training partitions as follows:

1. A unique identifier, that is, an index, is assigned to each example of the full original dataset.
2. A level of noise x %, of either class noise (random or pairwise) or attribute noise (random or Gaussian), is introduced into a copy of the full original dataset. Each example maintains its identifier in this noisy copy.
3. Both datasets, the original one and the noisy copy, are partitioned into fivefolds. Each one of the folds in the original dataset must have examples with the same identifiers that the corresponding fold in the noisy copy.
4. The training partitions are built from the noisy copy (using 4 of the fivefolds), whereas the test partitions are formed of instances from the original dataset (using the fold with examples whose identifiers have not been considered in the training set).

Introducing noise into the training partitions while keeping the test partitions noise-free, as performed in other works in the literature [56], allows one to observe how noisy data affect the training process, observing how the test results are degraded depending on the type and level of noise introduced. Furthermore, the robustness of the methods can be better studied since the effects of noise are isolated in the training process.

The accuracy estimation of the classifiers in a dataset is obtained by means of 5 runs of a stratified fivefolds cross-validation (5-fcv). Hence, a total of 25 runs per dataset, noise type, and level are averaged. Each fold has a larger number of examples considering 5 partitions than considering a higher number of partitions, for example, 10, which is desirable in multi-class problems where some of the classes might be not represented in the test sets. Therefore, the performance of each classifier built is evaluated with a larger number of examples in each test set of the 5-fcv. This fact lets that little modifications in the classifier due to the effect of noise on training sets to be shown better in test sets because we consider a larger number of examples. Furthermore, performing 5 runs of each 5-fcv, the final results obtained are stabilized.

A large collection of new noisy datasets are created from the aforementioned 20 base datasets. Both types of noise are independently considered: class and attribute noise. For each type of noise, the noise levels ranging from $x = 0\,\%$ (base datasets) to $x = 50\,\%$, by increments of $5\,\%$, are studied. Therefore, 200 noisy datasets are created for each of the four noise schemes. The total number of datasets in the experimentation is 820. Hence, considering the 5×5fcv of the 820 datasets, 20,500 executions are carried out for each classifier (which are repeated for the OVO and non-OVO versions). All these datasets are available on the web page associated with this paper.

4.3 Algorithms and parameters

The choice of the learning algorithms—C4.5 [42], RIPPER [11], and k-NN [35]—has been made on the basis of their good behavior in a large number of real-world problems and their different characteristics against noise. They have been also considered in previous works focused on noisy data [37,56]. Moreover, notice that all these learning methods are capable of handling multiple classes inherently, which is needed in order to be comparable against the usage of the OVO strategy.

Table 2 Setup of the parameters for the classification algorithms

Rule-based learning		Instance-based learning
C4.5	RIPPER	k-NN
Confidence: $c = 0.25$	Folds: $f = 3$	Neighbors: $k = 3, 5$
Min. instances per leaf: $i = 2$	Optimizations: $r = 2$	Distance: HVDM
Pruned tree		

C4.5 and RIPPER are considered robust learners tolerant to noisy data. Both use pruning strategies to reduce the chances of classifiers to be affected by noisy instances from the training data [40,41]. However, when the noise level is relatively high, even these robust learners may obtain a poor performance. Regarding k-NN, it is known to be more sensitive to noise than other learning algorithms. Furthermore, the value of k determines a higher or lower sensitivity to noise [29], since a larger value of k usually implies a lower influence on the prediction of the closest potential noisy examples. In this manner, this paper studies the effect of noise on the performance of robust and noise-sensitive learners, and more specifically focusing on multi-class problems, it compares their baseline results with respect to the usage of the OVO strategy. Hence, we check whether the advantages usually attributed to OVO are maintained in the presence of noise or not; in such a way, we provide an in-depth study of these cases (Sects. 5, 6) followed by a thorough explanation of the results (Sect. 7).

The classification algorithms have been executed using the default parameters recommended by the authors, which are shown in Table 2.

4.4 Methodology of analysis

In order to check the suitability of methods using OVO when dealing with noisy data, the results of C4.5, RIPPER, 3-NN, and 5-NN with and without decomposition are compared one another using three distinct properties:

1. The performance of the classification algorithms on the test sets for each level of induced noise defined as its accuracy rate. For the sake of brevity, only averaged results are shown (the rest can be found on the web page associated with this paper), but it must be taken into account that our conclusions are based on the proper statistical analysis, which considers all the results (not averaged).
2. The *relative loss of accuracy* (RLA) (Eq. 3) is used to measure the percentage of variation of the accuracy of the classifiers in a concrete noise level with respect to the original case with no additional noise:

$$\mathrm{RLA}_{x\,\%} = \frac{\mathrm{Acc}_{0\,\%} - \mathrm{Acc}_{x\,\%}}{\mathrm{Acc}_{0\,\%}}, \tag{3}$$

   where $\mathrm{RLA}_{x\,\%}$ is the relative loss of accuracy at a noise level $x\,\%$, $\mathrm{Acc}_{0\,\%}$ is the test accuracy in the original case, that is, with $0\,\%$ of induced noise, and $\mathrm{Acc}_{x\,\%}$ is the test accuracy with a noise level $x\,\%$.
3. Box-plots are used to easily analyze the distribution of the RLA values. The values of the median and the interquartile range, along with its size, can provide a good approximation about the robustness of the methods over all the datasets. Thus, a method with a lower median and a lower and more compact interquartile range will be always preferable, since

its behavior with new noisy datasets is more similar in accuracy to that obtained with the original dataset.

In order to properly analyze the performance and RLA results, the Wilcoxon's signed rank statistical test is used, as suggested in the literature [12]. This is a nonparametric pairwise test that aims to detect significant differences between two sample means, that is, the behavior of the two algorithms involved in each comparison. For each type and noise level, the OVO and non-OVO versions will be compared using Wilcoxon's test and the p values associated with these comparisons will be obtained. The p value represents the lowest level of significance of a hypothesis that results in a rejection and it allows one to know whether two algorithms are significantly different and the degree of their difference. We will consider a difference to be significant if the p value obtained is lower than 0.1—even though p values slightly higher than 0.1 might be showing important differences. We study both, performance and robustness, because the conclusions reached with one of these metrics necessary not imply the same conclusions with the other one.

5 Analysis of the OVO strategy with class noise

In this section, the performance and robustness of the classification algorithms using the OVO decomposition with respect to its baseline results when dealing with data suffering from class noise are analyzed. Section 5.1 is devoted to the study of the random class noise scheme, whereas Sect. 5.2 analyzes the pairwise class noise scheme. The results obtained for each single dataset can be found on the web page associated with this paper.

5.1 Random class noise scheme

Table 3 shows the test accuracy and RLA results for each classification algorithm at each noise level along with the associated p-values between the OVO and the non-OVO version from the Wilcoxon's test. The few exceptions where the baseline classifiers obtain more ranks than the OVO version in the Wilcoxon's test are indicated with a star next to the p value.

From these results, the following points can be highlighted:

- The test accuracy of the methods using OVO is higher at all the noise levels. Moreover, the low p values show that this advantage in favor of OVO is significant.
- The RLA values of the methods using OVO are lower than those of the baseline methods at all noise levels. These differences are also statistically significant as reflected by the low p values. Only at some very low noise levels—5 and 10 % for C4.5 and 5 % for 5-NN—the results between the OVO and the non-OVO version are statistically equivalent, but notice that the OVO decomposition does not hinder the results, simply the loss is not lower.

Figure 1 shows the distribution of the RLA results of each algorithm at each noise level on datasets with random class noise. For all the classification algorithms, these graphics show that the medians of the RLA results of the OVO approach are much lower with respect to those of non-OVO. Moreover, the interquartile range is generally lower and more compact for OVO. Therefore, when noise randomly affects the class labels, the suitability of the OVO decomposition is proved to be advantageous. The binary decomposition of the problem provides better predictions. Hence, OVO is more robust against this type of class noise, obtaining a greater performance and a lower RLA result. This may be attributed to the

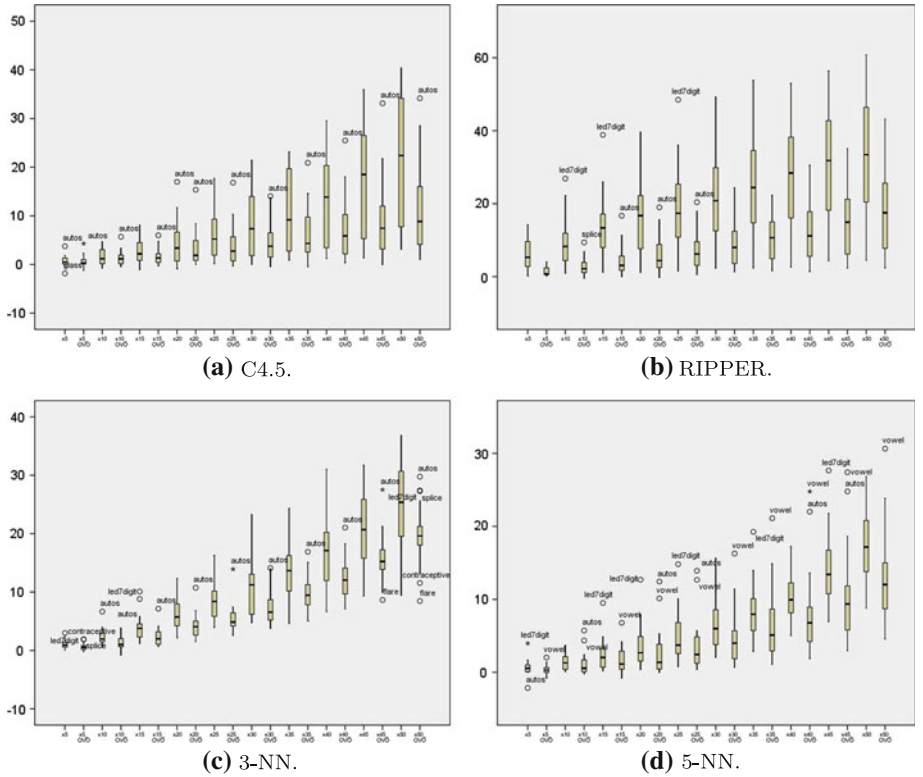
Table 3 Test accuracy, RLA results, and p values on datasets with random class noise

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
Test accuracy								
Results (%)								
0	81.66	82.70	77.92	82.15	81.79	83.39	82.10	83.45
5	81.13	82.14	73.51	80.82	81.00	82.93	81.65	83.14
10	80.50	81.71	71.30	79.86	80.00	82.29	81.01	82.56
15	79.37	81.39	68.52	78.41	78.76	81.49	80.39	82.13
20	78.13	80.27	66.71	77.35	76.91	80.01	79.55	81.36
25	76.96	79.54	64.26	76.25	75.15	78.99	78.43	80.58
30	75.22	78.87	62.91	74.98	73.24	77.39	77.21	79.82
35	73.35	77.89	60.48	73.40	70.82	75.30	75.48	78.28
40	71.10	76.88	58.32	72.12	68.18	73.08	73.82	76.83
45	67.96	75.74	56.18	69.98	64.90	70.27	70.86	74.93
50	64.18	73.71	53.79	67.56	61.66	66.98	68.04	72.73
p values (%)								
0	–		–		–		–	
5	0.0206		0.0001		0.0003		0.0033	
10	0.0124		0.0001		0.0001		0.0036	
15	0.0008		0.0001		0.0001		0.0137	
20	0.0028		0.0001		0.0004		0.0017	
25	0.0010		0.0001		0.0002		0.0032	
30	0.0002		0.0001		0.0003		0.0013	
35	0.0003		0.0001		0.0008		0.0028	
40	0.0002		0.0001		0.0005		0.0111	
45	0.0003		0.0001		0.0019		0.0019	
50	0.0001		0.0001		0.0028		0.0008	
RLA value								
Results (%)								
0	–		–		–		–	
5	0.66	0.73	6.04	1.59	1.07	0.57	0.63	0.38
10	1.56	1.28	9.35	2.79	2.27	1.33	1.46	1.12
15	3.01	1.65	13.13	4.56	3.96	2.34	2.37	1.67
20	4.63	3.15	15.44	5.86	6.20	4.15	3.48	2.67
25	6.12	3.98	18.89	7.23	8.37	5.33	4.87	3.61
30	8.36	4.90	20.74	8.82	10.78	7.27	6.40	4.53
35	10.77	6.11	23.97	10.76	13.48	9.74	8.46	6.38
40	13.46	7.41	26.72	12.35	16.79	12.35	10.30	8.11
45	17.30	8.86	29.70	15.05	20.74	15.79	14.12	10.48
50	21.87	11.29	32.74	18.10	24.51	19.64	17.47	13.12
p values (%)								
0	–		–		–		–	
5	0.7369*		0.0003		0.0040		0.6012	

Table 3 continued

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
10	0.5257		0.0001		0.0017		0.1354	
15	0.0025		0.0001		0.0012		0.0731	
20	0.0304		0.0001		0.0004		0.0479	
25	0.0017		0.0001		0.0005		0.0522	
30	0.0006		0.0001		0.0005		0.0124	
35	0.0003		0.0001		0.0025		0.0276	
40	0.0001		0.0001		0.0012		0.0333	
45	0.0002		0.0001		0.0022		0.0057	
50	0.0001		0.0001		0.0036		0.0015	

Those cases where the baseline classifiers obtain more ranks than the OVO version in the Wilcoxon's test are indicated with a *



(a) C4.5. **(b)** RIPPER. **(c)** 3-NN. **(d)** 5-NN.

Fig. 1 Box-plots representing the distribution of the RLA results on datasets with random class noise

division of the mislabeled examples, hindering in this way only some classifiers. As these classifiers are only a part of the global system, they do not affect as much as in the original case.

5.2 Pairwise class noise scheme

The pairwise class noise results are shown in Table 4. The test accuracy and RLA of each classification algorithm at each noise level are presented. The associated p values between the OVO and non-OVO version of each algorithm are also shown. The following points can be concluded:

- The test accuracy of the methods using OVO is statistically better—as shown by the p values—than those of non-OVO at almost all noise levels. C4.5 and RIPPER at a noise level of 50 % are exceptions: with C4.5, both OVO and non-OVO, are statistically equivalent; with RIPPER, the non-OVO version is statistically better. In the last part of this subsection, we try to obtain an explanation to these results.
- Attending to the RLA results:
  - C4.5 with OVO is only statistically better at intermediate noise levels 15–20 %. Both methods are statistically equivalent in the rest of noise levels—except at the maximum noise level, 50 %, where non-OVO is statistically better. But having equivalent RLA, OVO performs statistically better in most of the cases.
  - Both versions of RIPPER, with and without OVO, are statistically equivalent at all noise levels—except at the maximum noise level 50 % where non-OVO is statistically better.
  - 3-NN and 5-NN with OVO present better RLA results. The lower p-values are generally obtained from 20 to 25 % of noise. Other levels of noise are also remarkable with 5-NN, as 5 %.

Figure 2 shows the distribution of RLA results with pairwise class noise. These graphics show similar conclusions to those obtained from the analysis of the RLA results and the corresponding p-values:

- At the lowest noise levels, C4.5 and RIPPER using OVO are slightly better than non-OVO (attending to their medians and interquartile ranges). However, from 30 % on (C4.5) and from 25 % on (RIPPER), the methods using OVO are more detrimental than those not using it.
- 3-NN and 5-NN with and without OVO are much more similar, but OVO is better at some noise levels.

These results show that OVO achieves more accurate predictions when dealing with this type of noise; however, it is not so advantageous with C4.5 or RIPPER as with k-NN in terms of robustness when noise only affects one class. For example, the behavior of RIPPER with this noise scheme can be related to the hierarchical way in which the rules are learned: it starts learning rules of the class with the lowest number of examples and continues learning those classes having more examples. When introducing this type of noise, RIPPER might change its training order, but the remaining part of the majority class can still be properly learned, since it has now more priority. Moreover, the original second majority class, now with noisy examples, will probably be the last one to be learned and it would depend on how the rest of the classes have been learned. Decomposing the problem with OVO, a considerable number of classifiers will have a notable quantity of noise—those of the majority and the second majority classes and hence, the tendency to predict the original majority class decreases—when the noise level is high, it strongly affects the accuracy, since the majority has more influence on it.

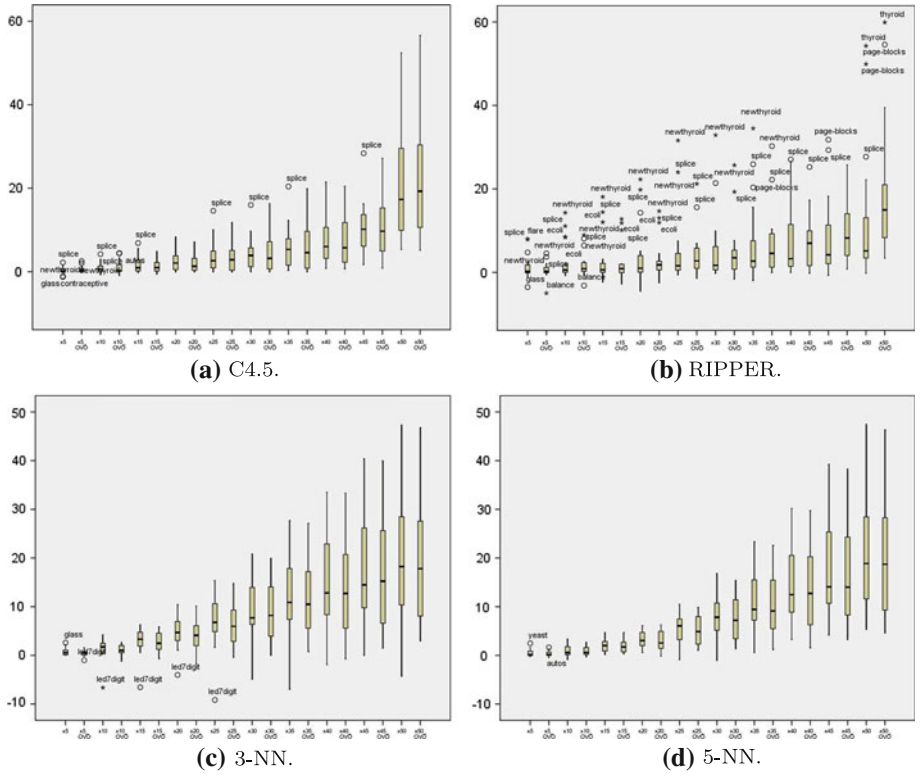
Table 4 Test accuracy, RLA results, and p values on datasets with pairwise class noise

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
Test accuracy								
Results (%)								
0	81.66	82.70	77.92	82.15	81.79	83.39	82.10	83.45
5	81.40	82.24	76.94	81.71	81.24	83.02	81.78	83.19
10	80.94	81.86	75.94	80.71	80.65	82.36	81.42	82.82
15	80.43	81.71	75.64	80.32	79.25	80.97	80.49	81.94
20	79.82	81.03	74.77	79.62	77.75	79.65	79.41	81.01
25	78.96	80.28	73.89	78.67	75.88	77.71	77.55	79.08
30	78.49	79.26	73.38	78.05	73.53	75.47	75.29	76.81
35	77.41	78.28	71.89	76.42	71.24	73.18	72.92	74.50
40	76.17	76.91	71.60	76.19	68.77	70.89	69.89	71.65
45	73.45	74.26	70.73	74.04	66.55	68.48	67.13	68.83
50	63.63	63.52	67.11	65.78	64.11	65.86	64.02	65.52
p values (%)								
0	0.0070		0.0002		0.0522		0.0930	
5	0.025		0.0001		0.0152		0.0228	
10	0.0033		0.0003		0.0019		0.0137	
15	0.0022		0.0003		0.0036		0.0090	
20	0.0017		0.0002		0.0005		0.0022	
25	0.0008		0.0005		0.0012		0.0015	
30	0.0090		0.0001		0.0045		0.0100	
35	0.0366		0.0013		0.0002		0.0032	
40	0.0276		0.0003		0.0015		0.0040	
45	0.0333		0.0333		0.0022		0.0072	
50	0.5016*		0.0930*		0.0008		0.0057	
RLA value								
Results (%)								
0	–		–		–		–	
5	0.30	0.56	1.17	0.48	0.74	0.44	0.48	0.34
10	0.91	1.01	2.38	1.72	1.25	1.22	0.89	0.82
15	1.62	1.25	2.58	2.17	2.88	2.81	2.05	1.88
20	2.39	2.13	3.52	3.03	4.69	4.33	3.29	2.93
25	3.52	3.13	4.58	4.22	6.75	6.58	5.48	5.22
30	4.16	4.49	5.13	4.84	9.57	9.09	8.05	7.81
35	5.50	5.69	6.77	6.82	12.12	11.76	10.76	10.41
40	7.01	7.38	7.25	7.12	15.03	14.29	14.37	13.68
45	10.38	10.65	8.28	9.70	17.57	17.01	17.53	16.86
50	21.03	22.28	12.22	18.75	20.23	20.00	21.06	20.67
p values (%)								
0	–		–		–		–	
5	0.2043*		0.2790		0.2954		0.1354	

Table 4 continued

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
10	0.8721*		0.3317		0.3507		0.3507	
15	0.0304		0.3507		0.2043		0.5503	
20	0.0859		0.4781		0.2959		0.0674	
25	0.3317		0.9702		0.1005		0.0620	
30	0.6813		0.6542		0.2471		0.3507	
35	0.7652		0.6542*		0.0674		0.1913	
40	0.6274		0.6274*		0.1169		0.0793	
45	0.7369		0.1169*		0.1259		0.0522	
50	0.0400*		0.0001*		0.0731		0.0620	

Those cases where the baseline classifiers obtain more ranks than the OVO version in the Wilcoxon's test are indicated with a *



(a) C4.5. **(b)** RIPPER. **(c)** 3-NN. **(d)** 5-NN.

Fig. 2 Box-plots representing the distribution of the RLA results on datasets with pairwise class noise

In contrast with the rest of noise schemes, with this noise scheme, all the datasets have different real percentages of noisy examples at the same noise level of x %. This is because each dataset has a different number of examples of the majority class, and thus a noise level of x % does not affect all the datasets in the same way. In this case, the percentage of

Table 5 Noise levels—remarked with gray—where the RLA results of OVO are better that those of the non-OVO version

		C4.5										RIPPER										3-NN										5-NN									
Dataset	N_{maj}	5	10	15	20	25	30	35	40	45	50	5	10	15	20	25	30	35	40	45	50	5	10	15	20	25	30	35	40	45	50	5	10	15	20	25	30	35	40	45	50
vowel	9.09																																								
penbased	10.42																																								
led7digit	11.50																																								
segment	14.29																																								
satimage	23.82																																								
vehicle	25.89																																								
autos	30.71																																								
flare	30.99																																								
yeast	31.17																																								
iris	33.33																																								
nursery	33.33																																								
glass	35.09																																								
ecoli	42.54																																								
contraceptive	42.70																																								
balance	46.20																																								
splice	51.88																																								
newthyroid	69.77																																								
shuttle	78.39																																								
page-blocks	89.79																																								
thyroid	92.59																																								

noisy examples with a noise level of x % is computed as $(x \cdot N_{\text{maj}})/100$, where N_{maj} is the percentage of examples of the majority class.

Therefore, in order to deepen in the analysis of the robustness results with this noise scheme, the RLA results on each single dataset must be studied. With this aim, Table 5 shows for each dataset, remarked in gray, those noise levels where the OVO version obtains better RLA results than the non-OVO version. The datasets appearing in this table are sorted by their percentage of examples of the majority class (N_{maj}). Noise affects adjacent datasets similarly—since close datasets have similar percentages of noisy examples at the same noise level of x %. Thus, those datasets and noise levels where OVO works better are easily identified.

From this table, several conclusions can be drawn:

- C4.5 and RIPPER with OVO have better RLA results on those dataset having a large number of examples of the majority class.
- 3-NN and 5-NN with OVO have better RLA results, without being affected by the number of examples of the majority class.

Therefore, following the previous analysis, we can state that when the noise only affects one class, OVO clearly provides advantages in terms of the accuracy of the classifiers built. However, methods using OVO are affected by noise in a similar way to those that do not use OVO, that is, both have similar robustness, although OVO outstands at some noise levels. The robustness of C4.5 and RIPPER with OVO is remarkable with those datasets having a large number of majority class examples, that is, with those dataset where the noise percentage is the highest. The robustness of k-NN methods is not so dependent on the number of examples of the majority class, and the methods with OVO have better RLA results, which are distributed more homogeneously.

6 Analysis of the OVO strategy with attribute noise

In this section, the performance and robustness of the classification algorithms using OVO in comparison with its non-OVO version when dealing with data with attribute noise are

analyzed. Section 6.1 is devoted to the study of the random attribute noise scheme, whereas Sect. 6.2 analyzes the Gaussian attribute noise scheme.

6.1 Random attribute noise scheme

The test accuracy, RLA results, and p values of each classification algorithm at each noise level are shown in Table 6. From these results, the following points can be highlighted:

- The test accuracy of the methods using OVO is always statistically better at all the noise levels.
- The RLA values of the methods using OVO are lower than those of the baseline methods at all noise levels—except in the case of C4.5 with a 5 % of noise level. Regarding the p values, a clear tendency is observed, the p-value decreases when the noise level increases with all the algorithms.
- With most of the methods—C4.5, RIPPER, and 5-NN—the p values of the RLA results at the lowest noise levels (up to 20–25 %) show that the robustness of OVO and non-OVO methods is statistically equivalent. From that point on, the OVO versions statistically outperform the non-OVO ones. The case of 3-NN is even more favorable to OVO, since only a high p value is found at the lowest noise level, i.e., 5 %.

Figure 3 shows the box-plots of RLA results. For all the classification algorithms and noise levels, these graphics show that the medians of the RLA results of OVO are much lower with respect those of non-OVO. Moreover, the interquartile range is also generally lower and more compact for the methods using OVO.

Therefore, the usage of OVO is clearly advantageous in terms of accuracy and robustness when noise affects the attributes in a random and uniform way. This behavior is particularly notable with the highest noise levels, where the effects of noise are expected to be more detrimental.

6.2 Gaussian attribute noise scheme

In Table 7, the test accuracy and RLA results of each classification algorithm at each noise level, along with the associated p value between the OVO and non-OVO version of each algorithm, are shown. From these results, the following conclusions can be pointed out:

- The test accuracy of the methods using OVO is better at all the noise levels. Moreover, the low p values show that this advantage in favor of OVO is statistically significant.
- Regarding the RLA results, the p values show a clear decreasing tendency when the noise level increases with all the algorithms. In the case of C4.5, OVO is statistically better from a 35 % of noise level on, and in 3-NN from 20 % on. RIPPER and 5-NN are statistically equivalent at all noise levels—although 5-NN with OVO obtains higher Wilcoxon's ranks.

It is important to note that in some cases, particularly in the comparisons involving RIPPER, some RLA results show that OVO is better than the non-OVO version in average but the latter obtains more ranks in the statistical test—even though these differences are not significant. This is due to the extreme results of some individual datasets, such as *led7digit* or *flare*, in which the RLA results of the non-OVO version are much worse than those of the OVO version. Anyway, we should notice that average results themselves are not meaningful and the corresponding nonparametric statistical analysis must be carried out in order to extract meaningful conclusions, which reflects the real differences between algorithms.

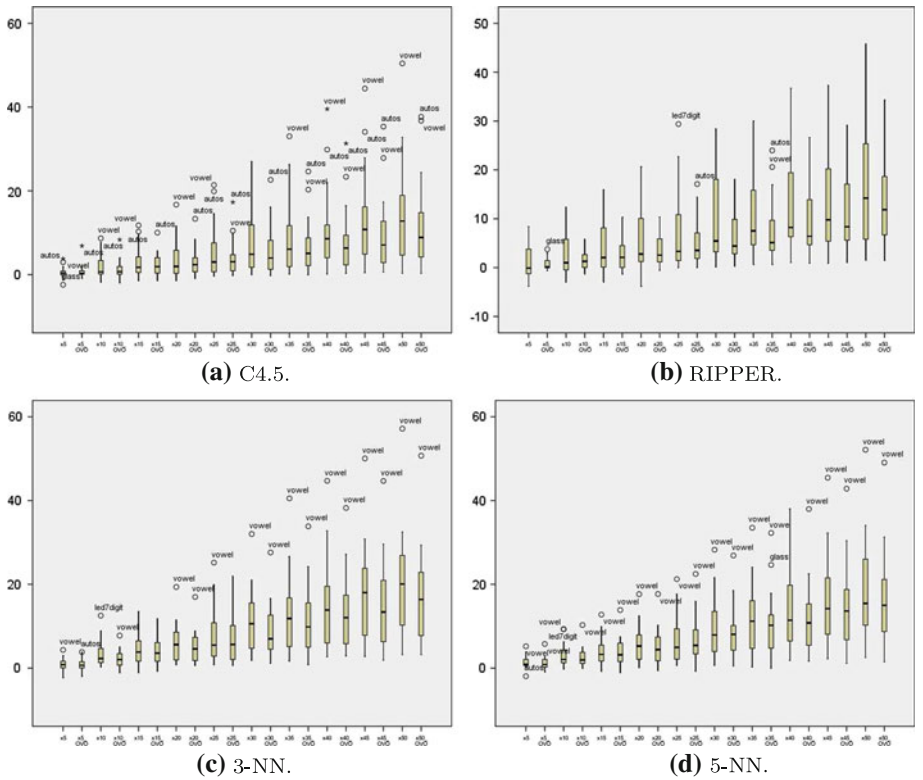
Table 6 Test accuracy, RLA results, and p values on datasets with random attribute noise

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
Test accuracy								
Results (%)								
0	81.66	82.70	77.92	82.15	81.79	83.39	82.10	83.45
5	81.26	82.10	77.18	81.50	80.90	82.53	81.01	82.45
10	80.31	81.65	76.08	80.85	79.23	81.52	79.81	81.34
15	79.39	80.83	74.83	80.03	78.31	80.22	78.97	80.31
20	78.71	80.27	73.95	79.15	76.99	79.20	77.63	79.38
25	77.54	79.64	72.77	78.11	75.36	77.71	76.58	77.96
30	76.01	78.25	71.25	77.06	73.37	76.05	74.68	76.46
35	74.55	77.42	70.05	76.15	71.62	74.28	73.05	75.01
40	73.58	76.19	68.66	74.56	69.62	72.66	71.29	73.65
45	71.79	75.21	67.64	73.35	67.56	70.56	69.26	71.53
50	70.49	73.51	65.50	71.66	65.88	69.15	67.72	70.07
p values (%)								
0		0.0070		0.0002		0.0522		0.0930
5		0.0400		0.0012		0.0038		0.0100
10		0.0169		0.0003		0.0004		0.0910
15		0.0169		0.0001		0.0022		0.0707
20		0.0057		0.0003		0.0008		0.0015
25		0.0007		0.0005		0.0017		0.0080
30		0.0043		0.0001		0.0005		0.0112
35		0.0004		0.0001		0.0019		0.0032
40		0.0032		0.0001		0.0005		0.0006
45		0.0009		0.0002		0.0008		0.0012
50		0.0036		0.0007		0.0001		0.0011
RLA value								
Results (%)								
0		–		–		–		–
5	0.50	0.74	0.97	0.80	0.98	0.94	1.39	1.20
10	1.82	1.32	2.56	1.62	3.45	2.23	3.03	2.62
15	3.02	2.41	4.24	2.69	4.31	3.82	3.97	3.87
20	3.88	3.11	5.46	3.77	5.85	4.99	5.72	5.03
25	5.48	3.87	7.10	5.13	7.96	6.87	6.94	6.80
30	7.54	5.77	9.20	6.42	10.71	9.01	9.57	8.76
35	9.38	6.70	10.89	7.57	12.65	11.12	11.57	10.49
40	10.64	8.25	12.81	9.64	15.22	12.98	13.73	12.08
45	13.04	9.60	14.13	11.24	17.91	15.72	16.34	14.85
50	14.74	11.74	17.21	13.33	19.96	17.40	18.14	16.55
p values (%)								
0		–		–		–		–
5		0.4115*		0.5016*		0.8813*		0.5755

Table 6 continued

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
10	0.4781		0.5755		0.1560		1.0000*	
15	0.2471		0.3507		0.2627		0.9108	
20	0.2471		0.1454		0.0930		0.1354	
25	0.0930		0.2322		0.0620		0.5257	
30	0.0304		0.0438		0.0124		0.1672	
35	0.0015		0.0064		0.0333		0.0731	
40	0.0569		0.0036		0.0100		0.0111	
45	0.0137		0.0251		0.0064		0.0152	
50	0.0152		0.0064		0.0008		0.0228	

Those cases where the baseline classifiers obtain more ranks than the OVO version in the Wilcoxon's test are indicated with a *



(a) C4.5. **(b)** RIPPER. **(c)** 3-NN. **(d)** 5-NN.

Fig. 3 Box-plots representing the distribution of the RLA results on datasets with random attribute noise

Figure 4 shows the distribution of the RLA results by means of box-plots. As in the case of the random attribute noise, these graphics show that the medians and interquartile ranges of the RLA results of OVO are much lower with respect to those of non-OVO.

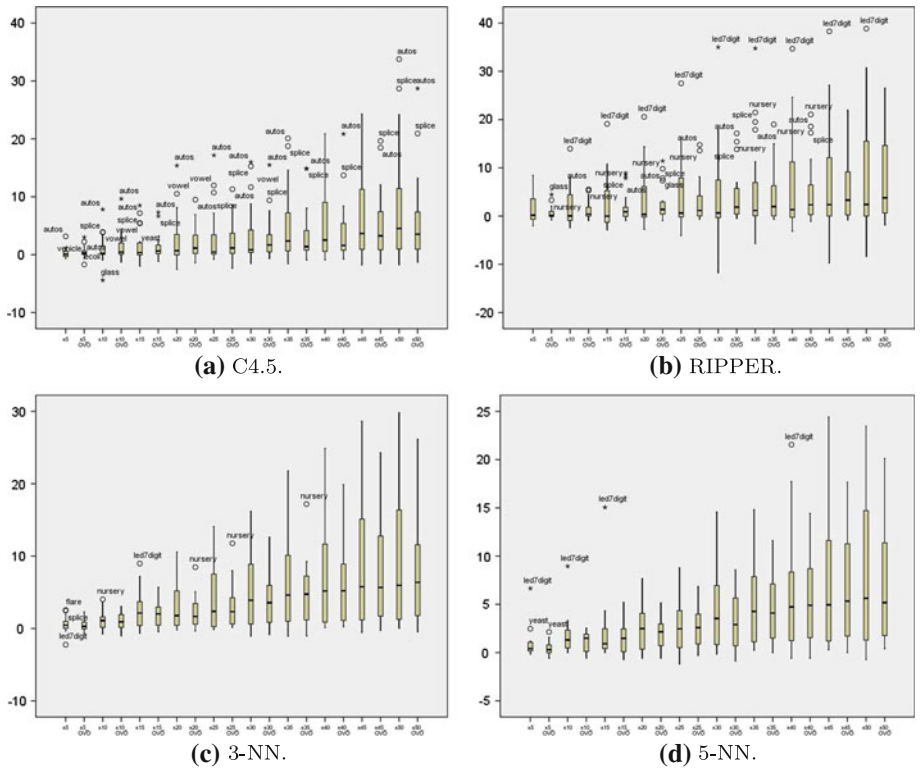
Table 7 Test accuracy, RLA results and p values on datasets with Gaussian attribute noise

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
Test accuracy								
Results (%)								
0	81.66	82.70	77.92	82.15	81.79	83.39	82.10	83.45
5	81.46	82.33	76.82	81.64	81.29	83.02	81.52	83.09
10	80.93	81.67	76.53	81.12	80.88	82.54	80.91	82.52
15	80.51	81.69	76.16	80.64	79.87	81.72	80.61	82.21
20	79.77	81.11	75.35	80.06	79.56	81.41	80.16	81.74
25	79.31	80.98	74.69	79.72	78.87	80.90	79.85	81.21
30	79.03	80.40	74.46	78.93	77.98	80.29	78.84	80.77
35	77.95	79.94	73.85	78.76	77.05	79.35	78.12	79.75
40	77.36	79.51	72.94	78.10	76.27	78.60	77.53	79.11
45	76.38	78.64	72.37	77.34	75.11	77.48	76.58	78.25
50	75.29	78.03	71.57	76.27	74.90	77.10	76.02	77.72
p values (%)								
0	0.0070		0.0002		0.0522		0.0930	
5	0.0442		0.0003		0.0033		0.0050	
10	0.1262		0.0004		0.0089		0.0064	
15	0.0152		0.0003		0.0033		0.0169	
20	0.0048		0.0002		0.0033		0.0036	
25	0.0019		0.0002		0.0100		0.0187	
30	0.0051		0.0003		0.0008		0.0025	
35	0.0007		0.0002		0.0036		0.0040	
40	0.0019		0.0003		0.0025		0.1262	
45	0.0004		0.0004		0.0017		0.0364	
50	0.0004		0.0008		0.0019		0.0251	
RLA value								
Results (%)								
0	–		–		–		–	
5	0.28	0.47	1.53	0.62	0.57	0.44	0.88	0.46
10	0.92	1.27	1.90	1.26	1.11	0.98	1.68	1.13
15	1.53	1.25	2.38	1.84	2.57	2.02	2.10	1.56
20	2.40	1.99	3.49	2.59	2.74	2.32	2.51	2.09
25	3.05	2.13	4.44	2.97	3.73	2.98	2.91	2.75
30	3.42	2.91	4.66	3.95	5.00	3.74	4.34	3.32
35	4.86	3.43	5.52	4.18	6.27	4.87	5.19	4.61
40	5.67	4.03	6.74	4.96	7.23	5.85	6.03	5.38
45	6.95	5.14	7.46	5.99	8.79	7.26	7.25	6.51
50	8.37	5.87	8.50	7.35	8.80	7.64	7.82	7.16
p values (%)								
0	–		–		–		–	
5	0.3144*		0.5503		0.6274		0.4781	

Table 7 continued

	C4.5		RIPPER		3-NN		5-NN	
	Base	OVO	Base	OVO	Base	OVO	Base	OVO
10	0.0766*		0.8519*		0.9702		0.4115	
15	0.5755		0.3507*		0.5016		0.4115	
20	0.8405		0.9108*		0.1913		0.3905	
25	0.3547		0.9702*		0.2627		0.9108	
30	0.6542		0.2627*		0.1005		0.2627	
35	0.1354		1.0000*		0.1169		0.3507	
40	0.1169		0.3905		0.0620		0.9405	
45	0.0930		0.9405*		0.0859		0.2471	
50	0.0090		0.6542*		0.0731		0.2180	

Those cases where the baseline classifiers obtain more ranks than the OVO version in the Wilcoxon's test are indicated with a *



(a) C4.5. **(b)** RIPPER. **(c)** 3-NN. **(d)** 5-NN.

Fig. 4 Box-plots representing the distribution of the RLA results on datasets with Gaussian attribute noise

Hence, the OVO approach is also suitable considering the accuracy achieved with this type of attribute noise. The robustness results are similar between the OVO and non-OVO versions with RIPPER and 5-NN. However, in C4.5 and 3-NN, there are statistical differences in favor of OVO at the highest noise levels. The box-plots show that methods using OVO

have a better and more homogeneous behavior with all the datasets, that is, methods using OVO have a more similar behavior with noisy problems, whereas the robustness results are much more unpredictable with the non-OVO methods. Hence, the behavior of the non-OVO methods is much better with some particular datasets, whereas with others is much worse. Nevertheless, regardless of the dataset considered, OVO is more stable with respect to its robustness results.

7 OVO Decomposition and Noise Schemes: Lessons Learned

Attending to the accuracy and robustness results analyzed in the previous sections, several conclusions can be extracted about the degree of disruptiveness of the different types of noise:

1. **Class Noise.** The random class noise scheme is much more disruptive than the pairwise class noise scheme.
2. **Attribute Noise.** The random attribute noise scheme is more disruptive than the Gaussian attribute noise scheme.
3. **Class vs. Attribute Noise**. The random class noise is clearly more disruptive than the random attribute noise. The ranking of disruptiveness follows with the pairwise class noise and the Gaussian attribute noise.

Regarding the behavior of the methods using the OVO decomposition when dealing with the different noise types, the following points can be pointed out:

1. **OVO & Class Noise.** The methods using OVO have better classification accuracies at the different noise levels. The robustness of the methods using OVO is more notable with the random class noise scheme, although it also outstands with the pairwise class noise scheme on those datasets with the highest percentages of noisy examples.
2. **OVO & Attribute Noise.** The usage of the OVO approach produces better accuracies with both attribute noise schemes. The robustness results of OVO are remarkable with the random attribute noise scheme, where the differences are larger due to its higher disruptiveness.
3. **OVO & Homogeneity of the Robustness Results.** The box-plots representing the distribution of RLA results show that methods using OVO are expected to have a more similar behavior with problems suffering from noise, being generally more robust than methods not using OVO.

The following remarks can be made about how the methods with a different noise tolerance benefit from the usage of OVO:

1. **OVO & Robust Learners.** In spite of being robust learners, the performance of C4.5 and RIPPER with the more disruptive noise schemes—the random class noise scheme and the random attribute noise scheme—is much more deteriorated as the noise level increases if they do not use OVO. Therefore, the good behavior of both methods with OVO considering standard datasets [16] remains with noisy datasets. Indeed, their differences with respect to the baseline classifiers are increased.
2. **OVO & Noise-sensitive Learners.** k-NN methods also benefits from the usage of OVO. The differences of robustness between the OVO and non-OVO version, although they are generally in favor of OVO, are not so accentuated as in the case of the robust learners. With the less disruptive noise schemes—the pairwise class noise and the Gaussian attribute noise—RLA results of OVO and non-OVO are affected more similarly than in the case of the random noise schemes, where the differences increases along with the noise level.

Therefore, the methods using OVO obtain better performances than baseline methods when noise is present in the data. Methods using OVO also generally create more robust classifiers than baseline classifiers when the noise level increases, and particularly, with the more disruptive noise schemes—random class noise and random attribute noise. We can conclude that these results may be supported by the following hypotheses:

1. **Distribution of the noisy examples in the subproblems.** When decomposing the problem into several binary subproblems with OVO, the complexity of the original problem decreases. As a consequence, noisy examples are divided into each subproblem, which also decreases the effect of noise in each binary classifier, thereby having a lower influence in the final performance.
2. **Increase of the separability of the classes.** The decomposition increases the separability of the classes, since only one boundary must be established. The corruptions of noise in these regions is less notable and the classifiers are less influenced.
3. **Collecting information from different classifiers.** The aggregation of the outputs from the base classifiers produces more robust classifiers, since some fails can be corrected. Besides, if a noisy example does not belong to one of both classes involved in the learning of a classifier, the classifier will not be affected by that example, and its predictions will not be hindered.

8 Concluding remarks

This paper analyzes the suitability of the usage of the OVO decomposition when dealing with noisy training datasets in multi-class problems. A large number of noisy datasets have been created considering different types, schemes, and levels of noise, as proposed in the literature. The C4.5 and RIPPER robust learners and the noise-sensitive k-NN method have been evaluated on these datasets, with and without the usage of OVO.

The results obtained have shown that the OVO decomposition improves the baseline classifiers in terms of accuracy when data are corrupted by noise in all the noise schemes studied. The robustness results are particularly notable with the more disruptive noise schemes—the random class noise scheme and the random attribute noise scheme—where a larger amount of noisy examples and with higher corruptions are available, which produces greater differences (with statistical significance).

Three hypotheses have been introduced aiming to explain the better performance and robustness of the methods using OVO when dealing with noisy data: (1) the distribution of the noisy examples in the subproblems, (2) the increase of the separability of the classes, and (3) the possibility of collecting information from different classifiers.

As final remark, we must emphasize that one usually does not know the type and level of noise present in the data of the problem that is going to be addressed. Decomposing a problem suffering from noise with OVO has shown a better accuracy, higher robustness, and homogeneity with all the classification algorithms tested. For this reason, the usage of the OVO decomposition strategy in noisy environments can be recommended as an easy to applicate, yet powerful tool to overcome the negative effects of noise in multi-class problems.

In future works, the synergy between OVO in combination with noise preprocessing techniques will be studied in order to check its suitability to deal with noisy data. Moreover, the behavior of OVO in different noisy frameworks must be studied, that is, where both the training and the test sets are affected by noise.

Acknowledgments Supported by the Spanish Ministry of Science and Technology under Project TIN2011-28488, and also by the Regional Projects P10-TIC-06858 and P11-TIC-9704. José A. Sáez holds an FPU scholarship from the Spanish Ministry of Education and Science.

References

1. Aggarwal CC (2009) On classification and segmentation of massive audio data streams. Knowl Inf Syst 20(2):137–156
2. Alcalá-Fdez J, Fernández A, Luengo J, Derrac J, García S, Sánchez L, Herrera F (2011) KEEL data-mining software tool: data set repository, integration of algorithms and experimental analysis framework. J Multiple Valued Logic Soft Comput 17(2–3):255–287
3. Alcalá-Fdez J, Sánchez L, García S, del Jesus M, Ventura S, Garrell J, Otero J, Romero C, Bacardit J, Rivas V, Fernández J, Herrera F (2009) KEEL: a software tool to assess evolutionary algorithms for data mining problems. Soft Comput Fusion Found Methodol Appl 13:307–318
4. Allwein EL, Schapire RE, Singer Y, Kaelbling P (2000) Reducing multiclass to binary: a unifying approach for margin classifiers. J Mach Learn Res 1:113–141
5. Anand A, Suganthan PN (2009) Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates. J Theor Biol 259(3):533–540
6. Anand R, Mehrotra K, Mohan CK, Ranka S (1995) Efficient classification for multiclass problems using modular neural networks. IEEE Trans Neural Netw 6(1):117–124
7. Bootkrajang J, Kabán A (2011) Multi-class classification in the presence of labelling errors. In: European symposium on artificial neural networks 2011 (ESANN 2011), pp 345–350
8. Brodley CE, Friedl MA (1999) Identifying mislabeled training data. J Artif Intell Res 11:131–167
9. Cao J, Kwong S, Wang R (2012) A noise-detection based AdaBoost algorithm for mislabeled data. Pattern Recognit 45(12):4451–4465
10. Chang CC, Lin CJ (2011) LIBSVM: A library for support vector machines. ACM Trans Intell Syst Technol 2:27:1–27:27
11. Cohen WW (1995) Fast effective rule induction. In: Proceedings of the twelfth international conference on machine learning. Morgan Kaufmann Publishers, pp 115–123
12. Demšar J (2006) Statistical comparisons of classifiers over multiple data sets. J Mach Learn Res 7:1–30
13. Dietterich TG, Bakiri G (1995) Solving multiclass learning problems via error-correcting output codes. J Artifl Intell Res 2:263–286
14. Duda RO, Hart PE, Stork DG (2001) Pattern classification, 2nd edn. Wiley, New York
15. Furnkranz J (2002) Round Robin classification
16. Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F (2011) 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 44:1761–1776
17. Galar M, Fernández A, Barrenechea E, Bustince H, Herrera F (2012) A review on ensembles for the class imbalance problem: bagging, boosting, and hybrid-based approaches. IEEE Trans Syst Man Cybern Part C Appl Rev 42(4):463–484
18. Gamberger D, Boskovic R, Lavrac N, Groselj C (1999) Experiments with noise filtering in a medical domain. In: Proceedings of the sixteenth international conference on machine learning. Morgan Kaufmann Publishers, pp 143–151
19. Guler I, Ubeyli ED (2007) Multiclass support vector machines for EEG-signals classification. IEEE Trans Inf Technol Biomed 11(2):117–126
20. Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. SIGKDD Explor 11:10–18
21. Hernández MA, Stolfo SJ (1998) Real-world data is dirty: data cleansing and the merge/purge problem. Data Min Knowl Discov 2:9–37
22. Hernández-Lobato D, Hernández-Lobato JM, Dupont P (2011) Robust multi-class Gaussian process classification. In: Annual conference on neural information processing systems (NIPS 2011), pp 280–288
23. Hido S, Tsuboi Y, Kashima H, Sugiyama M, Kanamori T (2011) Statistical outlier detection using direct density ratio estimation. Knowl Inf Syst 26(2):309–336
24. Hsu CW, Lin CJ (2002) A comparison of methods for multiclass support vector machines. IEEE Trans Neural Netw 13(2):415–425
25. Huber PJ (1981) Robust statistics. Wiley, New York
26. Hüllermeier E, Vanderlooy S (2010) Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting. Pattern Recognit 43(1):128–142

27. Knerr S, Personnaz L, Dreyfus G (1990) A stepwise procedure for building and training a neural network. In: Fogelman Soulié F, Hérault J (eds) Neurocomputing: algorithms, architectures and applications. Springer, Berlin, pp 41–50
28. Knerr S, Personnaz L, Dreyfus G, Member S (1992) Handwritten digit recognition by neural networks with single-layer training
29. Kononenko I, Kukar M (2007) Machine learning and data mining: introduction to principles and algorithms. Horwood Publishing Limited, New York
30. Liu KH, Xu CG (2009) A genetic programming-based approach to the classification of multiclass microarray datasets. Bioinformatics 25(3):331–337
31. Liu L, Liang Q (2011) A high-performing comprehensive learning algorithm for text classification without pre-labeled training set. Knowl Inf Syst 29(3):727–738
32. Lorena A, de Carvalho A, Gama J (2008) A review on the combination of binary classifiers in multiclass problems. Artif Intell Rev 30:19–37
33. Luengo J, García S, Herrera F (2012) On the choice of the best imputation methods for missing values considering three groups of classification methods. Knowl Inf Syst 32(1):77–108
34. Mayoraz E, Moreira M (1996) On the decomposition of polychotomies into dichotomies
35. McLachlan GJ (2004) Discriminant analysis and statistical pattern recognition. Wiley, New York
36. Ménard PA, Ratté S (2011) Classifier-based acronym extraction for business documents. Knowl Inf Syst 29(2):305–334
37. Nettleton D, Orriols-Puig A, Fornells A (2010) A study of the effect of different types of noise on the precision of supervised learning techniques. Artif Intell Rev 33(4):275–306
38. Passerini A, Pontil M, Frasconi P (2004) New results on error correcting output codes of kernel machines. IEEE Trans Neural Netw 15:45–54
39. Pimenta E, Gama J (2005) A study on error correcting output codes. In: Portuguese conference on artificial intelligence EPIA 2005, pp 218–223
40. Quinlan JR (1986) Induction of decision trees. In: Machine learning, pp 81–106
41. Quinlan JR (1986) The effect of noise on concept learning. In: Machine learning: an artificial intelligence approach, chap. 6. Morgan Kaufmann Publishers, pp 149–166
42. Quinlan JR (1993) C4.5: programs for machine learning. Morgan Kaufmann Publishers, San Francisco
43. Rifkin R, Klautau A (2004) In defense of one-vs-all classification. J Mach Learn Res 5:101–141
44. da Silva I, Adeodato P (2011) PCA and gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets. In: Neural networks (IJCNN), the 2011 international joint conference on, pp 2664–2669
45. Sun Y, Wong AKC, Kamel MS (2009) Classification of imbalanced data: a review. Int J Pattern Recognit Artif Intell 687–719
46. Teng CM (1999) Correcting noisy data. In: Proceedings of the sixteenth international conference on machine learning. Morgan Kaufmann Publishers, San Francisco, pp 239–248
47. Teng CM (2004) Polishing blemishes: Issues in data correction. IEEE Intell Syst 19:34–39
48. Vapnik V (1998) Statistical learning theory. Wiley, New York
49. Verikas A, Guzaitis J, Gelzinis A, Bacauskiene M (2011) A general framework for designing a fuzzy rule-based classifier. Knowl Inf Syst 29(1):203–221
50. Wang RY, Storey VC, Firth CP (1995) A framework for analysis of data quality research. IEEE Trans Knowl Data Eng 7(4):623–640
51. Wu TF, Lin CJ, Weng RC (2004) Probability estimates for multi-class classification by pairwise coupling. JMach Learn Res 5:975–1005
52. Wu X (1996) Knowledge acquisition from databases. Ablex Publishing Corp, Norwood
53. Wu X, Zhu X (2008) Mining with noise knowledge: error-aware data mining. IEEE Trans Syst Man Cybern Part A Syst Humans 38(4):917–932
54. Zhang C, Wu C, Blanzieri E, Zhou Y, Wang Y, Du W, Liang Y (2009) Methods for labeling error detection in microarrays based on the effect of data perturbation on the regression model. Bioinformatics 25(20):2708–2714
55. Zhong S, Khoshgoftaar TM, Seliya N (2004) Analyzing software measurement data with clustering techniques. IEEE Intell Syst 19(2):20–27
56. Zhu X, Wu X (2004) Class noise vs. attribute noise: a quantitative study. Artif Intell Rev 22:177–210
57. Zhu X, Wu X, Chen Q (2003) Eliminating class noise in large datasets. In: Proceeding of the twentieth international conference on machine learning, pp 920–927
58. Zhu X, Wu X, Yang Y (2004) Error detection and impact-sensitive instance ranking in noisy datasets. In: Proceedings of the nineteenth national conference on artificial intelligence. AAAI Press, pp 378–383

Author Biographies

José A. Sáez received his M.Sc. in Computer Science from the University of Granada, Granada, Spain, in 2009. He is currently a Ph.D. student in the Department of Computer Science and Artificial Intelligence in the University of Granada. His main research interests include noisy data in classification, discretization methods and imbalanced learning.

Mikel Galar received the M.Sc. and Ph.D. degrees in Computer Science in 2005 and 2010, both from the Public University of Navarra, Pamplona, Spain. He is currently a teaching assistant in the Department of Automatics and Computation at the Public University of Navarra. His research interests are data-minig, classification, multi-classification, ensemble learning, evolutionary algorithms and fuzzy systems.

Julián Luengo received the M.S. degree in Computer Science and the Ph.D. degree from the University of Granada, Granada, Spain, in 2006 and 2011, respectively. His research interests include machine learning and data mining, data preparation in knowledge discovery and data mining, missing values, data complexity and fuzzy systems.

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain. He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has had more than 200 papers published in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001). He currently acts as Editor in Chief of the international journal "Progress in Artificial Intelligence" (Springer) and serves as Area Editor of the Journal Soft Computing (area of evolutionary and bioinspired algorithms) and International Journal of Computational Intelligence Systems (area of information systems). He acts as Associated Editor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Knowledge and Information Systems, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, Swarm and Evolutionary Computation. He received the following honors and awards: ECCAI Fellow 2009, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", and International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010). His current research interests include computing with words and decision making, data mining, bibliometrics, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.