Pattern Recognition 44 (2011) 1761–1776

Contents lists available at ScienceDirect

Pattern Recognition

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

An overview of ensemble methods for binary classifiers in multi-class problems: Experimental study on one-vs-one and one-vs-all schemes

Mikel Galar [a,*], Alberto Fernández [b], Edurne Barrenechea [a], Humberto Bustince [a], Francisco Herrera [c]

[a] Departamento de Automática y Computación, Universidad Pública de Navarra, Campus Arrosadía s/n, P.O. Box 31006, Pamplona, Spain
[b] Department of Computer Science, University of Jaén, P.O. Box 23071, Jaén, Spain
[c] Department of Computer Science and Artificial Intelligence, University of Granada, P.O. Box 18071, Granada, Spain

ARTICLE INFO

Article history:
Received 22 June 2010
Received in revised form
7 January 2011
Accepted 24 January 2011
Available online 1 February 2011

Keywords:
Multi-classification
Pairwise learning
One-vs-one
One-vs-all
Decomposition strategies
Ensembles

ABSTRACT

Classification problems involving multiple classes can be addressed in different ways. One of the most popular techniques consists in dividing the original data set into two-class subsets, learning a different binary model for each new subset. These techniques are known as binarization strategies.

In this work, we are interested in ensemble methods by binarization techniques; in particular, we focus on the well-known one-vs-one and one-vs-all decomposition strategies, paying special attention to the final step of the ensembles, the combination of the outputs of the binary classifiers. Our aim is to develop an empirical analysis of different aggregations to combine these outputs. To do so, we develop a double study: first, we use different base classifiers in order to observe the suitability and potential of each combination within each classifier. Then, we compare the performance of these ensemble techniques with the classifiers' themselves. Hence, we also analyse the improvement with respect to the classifiers that handle multiple classes inherently.

We carry out the experimental study with several well-known algorithms of the literature such as Support Vector Machines, Decision Trees, Instance Based Learning or Rule Based Systems. We will show, supported by several statistical analyses, the goodness of the binarization techniques with respect to the base classifiers and finally we will point out the most robust techniques within this framework.

1. Introduction

Supervised Machine Learning consists in extracting knowledge from a set of n input examples $x_1, \ldots, x_n$ characterized by i features $a_1, \ldots, a_i \in \mathbb{A}$, including numerical or nominal values, where each instance has associated a desired output y_j and the aim is to learn a system capable of predicting this output for a new unseen example in a reasonable way (with good generalization ability). This output can be a continuous value $y_j \in \mathbb{R}$ or a class label $y_j \in \mathbb{C}$ (considering an m class problem $\mathbb{C} = \{c_1, \ldots, c_m\}$). In the former case, it is a regression problem, while in the latter it is a classification problem [22]. In classification, the system generated by the learning algorithm is a mapping function defined over the patterns $\mathbb{A}^i \rightarrow \mathbb{C}$ and it is called a classifier.

Classification tasks are widely used in real-world applications, many of them are classification problems that involve more than two classes, the so-called multi-class problems. Their application domain is diverse, for instance, in the field of bioinformatics, classification of microarrays [51] and tissues [71], which operate with several class labels. Computer vision multi-classification techniques play a key role within objects [72], fingerprints [41] and sign language [8] recognition tasks, whereas in medicine, multiple categories are considered in problems such as cancer [6] or electroencephalogram signals [38] classification.

Usually, it is easier to build a classifier to distinguish only between two classes than to consider more than two classes in a problem, since the decision boundaries in the former case can be simpler. This is why binarization techniques have come up to deal with multi-class problems by dividing the original problem into easier to solve binary classification problems that are faced by binary classifiers. These classifiers are usually referred to as *base learners* or *base classifiers* of the system [30].

Different decomposition strategies can be found in the literature [52]. The most common strategies are called "one-vs-one" (OVO) [47] and "one-vs-all" (OVA) [17,7].

- OVO consists in dividing the problem into as many binary problems as all the possible combinations between pairs of classes, so one classifier is learned to discriminate between each pair, and then the outputs of these base classifiers are combined in order to predict the output class.
- OVA approach learns a classifier for each class, where the class is distinguished from all other classes, so the base classifier giving a positive answer indicates the output class.

* Corresponding author.
E-mail addresses: mikel.galar@unavarra.es (M. Galar), alberto.fernandez@ujaen.es (A. Fernández), edurne.barrenechea@unavarra.es (E. Barrenechea), bustince@unavarra.es (H. Bustince), herrera@decsai.ugr.es (F. Herrera).

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.01.017

In the recent years, different methods to combine the outputs of the base classifiers from these strategies have been developed, for instance, new approaches in the framework of probability estimates [76], binary-tree based strategies [23], dynamic classification schemes [41] or methods using preference relations [44,24], in addition to more classical well-known combinations such as Pairwise Coupling [39], Max-Wins rule [29] or Weighted Voting (whose robustness has been recently proved in [46]).

In the specialized literature, there exist few works comparing these techniques, neither between OVO and OVA, nor between different aggregation strategies. In [42] a study of OVO, OVA and Error Correcting Output Codes (ECOC) [21] is carried out, but only within multi-class Support Vector Machine (SVM) framework, whereas in [52] an enumeration of the different existing binarization methodologies is presented, but also without comparing them mutually. Fürnkranz [31] compared the suitability of OVO strategies for decision trees and decision lists with other ensemble methods such as boosting and bagging, showing also the improvement of using confidence estimates in the combination of the outputs. In [76], a comparison in the framework of probability estimates is developed, but no more possible aggregations for the outputs of the classifiers are considered.

Our aim is to carry out an exhaustive empirical study of OVO and OVA decompositions, paying special attention to the different ways in which the outputs of the base classifiers can be combined. The main novelties of this paper with respect to the referred previous studies [42,31,76,52] consist in the following points:

- We develop a study of the state-of-the-art on the aggregation strategies for OVO and OVA schemes. To do so, we will present an overview of the existing combination methods and we will compare their performances over a set of different real-world problems. Whereas a previous comparison exists between probability estimates by pairwise coupling [76], to the best of our knowledge, a comparison among the whole kind of aggregation methods is missing.
- We analyse the behaviour of the OVO and OVA schemes with different base learners, studying the suitability of these techniques in each base classifier.
- Since binarization techniques have been already proven as appropriate strategies to deal with multi-class problems [30,31,42,63] where the original classifiers do not naturally handle multiple class labels, we analyse whether they also improve the behaviour of the classifiers that have a built-in multi-class support.

Thus, our intention is to make a thorough analysis of the framework of binarization, answering two main questions:

1. Given that we want or have to use binarization, how should we do it? This is the main objective of this paper; to show the most robust aggregation techniques within the framework of binarization, which is still an unanswered question. Therefore, we analyse empirically which is the most appropriate binarization technique and which aggregation should be used in each case.
2. But, should we do binarization? This is an essential question when we can overcome multi-class problems in different ways (the base classifier is able to manage multiple classes). Previous works have been done showing the goodness of binarization techniques [30,31,42,63], although we develop a complementary study to stress their suitability with a complete statistical analysis among different learning paradigms that support multi-class data.

In order to achieve well-founded conclusions, we develop a complete empirical study. The experimental framework includes a set of nineteen real-world problems from the UCI repository [9]. The measures of performance are based on the accuracy rate and Cohen's kappa metric [18]. The significance of the results is supported by the proper statistical tests as suggested in the literature [20,35,34]. We chose several well-known classifiers from different Machine Learning paradigms as base learners, namely, SVMs [73], decision trees [62], instance-based learning [1], fuzzy rule based systems [16] and decision lists [19].

Finally, we included an indepth discussion on the results, that have been acquired empirically along the experimental study. This allowed us to answer the issues previously raised and summarize the lessons learned in this paper. Additionally, we showed some new challenges on the topic in correspondence with the obtained results.

The rest of this paper is organized as follows. Section 2 presents a thorough overview of the existing binarization techniques, with special attention to OVO and OVA strategies. Section 3 presents the state-of-the-art on the aggregation strategies for the outputs of those strategies that we use in this work. The experimental framework set-up is given in Section 4, that is, the algorithms used as base classifiers, the performance measures, the statistical tests, the data sets, the parameters for the algorithms and the description of a Web page associated to the paper (http://sci2s.ugr.es/ovo-ova), which contains complementary material to the experimental study. We develop the empirical analysis in Section 5. The discussion, including the lessons learned throughout this study and future works that remain to be addressed, is presented in Section 6. Finally, in Section 7 we make our concluding remarks.

2. Reducing multi-class problems by binarization techniques

In this section, we first describe the idea behind binarization techniques to deal with multi-class problems and review the existing decomposition strategies. Then, we explain with relative detail the most common strategies that we have used in the experimental study: OVO and OVA.

2.1. Preliminaries: decomposition strategies in multi-classification

Many proposals have been developed under the label of binarization for multi-classification [52]. The underlying idea is to undertake the multi-classification using binary classifiers with a divide and conquer strategy. Binary problems are simpler to solve than the original multi-category problem; however, drawbacks exist; the outputs from each new classifier have to be combined in order to make the final decision of the predicted class. Hence, a correct management of the outputs is crucial to produce a correct prediction.

The most common decomposition strategies include OVO [47] and OVA [17,7]. The former consists in using a binary classifier to discriminate between each pair of classes, while the latter, uses a binary classifier to distinguish between a single class and the remaining ones. In both cases, the simplest combination is the application of a voting strategy where each classifier votes for the predicted class and the one with the largest number of votes is predicted (in OVA only one positive answer is expected). Allwein et al. [4] proposed a unifying approach where both decomposition techniques are encoded within a code-matrix; the final output is obtained by decoding the code word given by the outputs of the classifiers for a new input pattern with an Error Correcting Output Code (ECOC) [21]. Many proposals have been made regarding ECOC, both studying automatic designing of the code-matrix [36,61,60] and using different error correcting codes [57,54].

Many research efforts have been directed to deal with the unclassifiable region in OVO strategy when the voting strategy is

used; Decision Directed Acyclic Graph (DDAG) [59] and Nesting OVO [50,49] are clear examples. Binary Trees [28] and hierarchical structures [65] have been proposed in a similar way; however, with some exceptions [23], these strategies instead of distinguishing one class from others or one class from other, they discriminate among groups of classes in each node, producing smaller trees, but these classifiers are usually more complex. These hierarchical structures do not need an aggregation strategy because the combination is inherent in the method.

Furthermore, binary decompositions have been widely used to develop multi-class SVM showing better performance than other multi-class SVM approaches [42]. The possibility to parallelize the training and testing of the binary classifiers is also a big advantage in favour of binarization strategies apart from their good performance.

Many works have shown the suitability of decomposition techniques.

- Regarding to OVO strategy, Knerr et al. [47] showed that a digit recognition problem could be linearly separable if it was considered in pairs, Fürnkranz [30,31] showed that using an OVO strategy to extend Ripper algorithm to multi-class problems outperforms the original multi-class Ripper and Hühn and Hüllermeier presented a fuzzy rule learner (FR3) [44] with a great classification performance based on OVO strategy and a new aggregation strategy based on learning valued preference structures [45]. Also in [64] the usefulness of OVO in linear dimensionality reduction for multi-class problems is shown.
- In general, OVA decomposition has not received the same attention in the literature as OVO strategy. Nevertheless, Rifkin and Kautau [63] claimed that OVA scheme is as accurate as any other approach when the base classifiers are well-tuned.
- Furthermore, a combination of OVA and OVO strategies was presented in [37], where the classes with the two largest outputs obtained from an OVA strategy were confronted in an OVO scheme to obtain the final prediction in order to tackle with false positives of the initial estimate.

2.2. One-vs-one decomposition scheme

OVO decomposition scheme divides an m class problem into $m(m-1)/2$ binary problems. Each problem is faced by a binary classifier, which is responsible for distinguishing between a different pair of classes. The learning phase of the classifiers is done using as training data only a subset of instances from the original training data set, that contains any of the two corresponding class labels, whereas the instances with different class labels are simply ignored.

In validation phase, a pattern is presented to each one of the binary classifiers. The output of a classifier given by $r_{ij} \in [0,1]$ is the confidence of the binary classifier discriminating classes i and j in favour of the former class. The confidence of the classifier for the latter is computed by $r_{ji}=1-r_{ij}$ if the classifier does not provide it (the class with the largest confidence is the output class of a classifier). These outputs are represented by a score matrix R:

$$R=\begin{pmatrix} - & r_{12} & \cdots & r_{1m} \\ r_{21} & - & \cdots & r_{2m} \\ \vdots & & & \vdots \\ r_{m1} & r_{m2} & \cdots & - \end{pmatrix} \quad (1)$$

The final output of the system is derived from the score matrix by different aggregation models. As we state previously, the state-of-the-art on the combinations to obtain the final output will be summarized in Section 3.1. A voting strategy is the simplest case, where each classifier gives a vote for the predicted class and the class with the largest number of votes is predicted.

Even though the number of classifiers is of m^2 order, each classifier is trained only with the samples from the corresponding classes and hence the required time is usually not high. This advantage in using only these samples, which output class is one of the considered pair of classes, can also be a disadvantage. When a new pattern is submitted to all the classifiers some of then could not have seen a similar instance before, so their output would not be significant (in [32] these instances are called *non-competent* examples). Usually, OVO aggregations suppose that the base classifiers will do a correct prediction in the cases where the new pattern is one of the considered pair of classes and therefore, considering a voting strategy, the class with the largest number of votes would be the correct class. However, the assumption about the base classifiers is not always fulfilled and this fact leads to new aggregation strategies.

2.3. One-vs-all decomposition scheme

OVA decomposition divides an m class problem into m binary problems. Each problem is faced by a binary classifier, which is responsible for distinguishing one of the classes from all other classes. The learning step of the classifiers is done using the whole training data, considering the patterns from the single class as positives and all other examples as negatives.

In the validation phase, a pattern is presented to each one of the binary classifiers and then the classifier that gives a positive output indicates the output class. In many cases, the positive output is not unique and some tie-breaking techniques are required. The most common approach uses the confidence of the classifiers to decide the final output, predicting the class from the classifier with the largest confidence. Instead of having a score matrix, when dealing with the outputs of OVA classifiers (where $r_i \in [0,1]$ is the confidence for class i), a score vector is used:

$$R=(r_1,r_2,\ldots,r_i,\ldots,r_m) \quad (2)$$

In Section 3.2 we summarize the state-of-the-art on the combinations for OVA approach, even though OVA methods have not got the same attention in the literature as OVO ones have got, Rifkin and Klautau defend their good performance [63].

In spite of using the whole data set to train each classifier, which prevents the submission of unseen instances to the classifiers in testing time, it also may lead to more complex classifiers than OVO scheme with higher training times. Other issue is that usually imbalanced training data sets are produced when instances from the single class are compared with all other instances in the data set, it is well-known in the field of Machine Learning that imbalanced data sets can cause some undesirable effects in the derived classifiers [15,40,70].

3. State-of-the-art on aggregation schemes for binarization techniques

In this section we describe the state-of-the-art on aggregation strategies for binarization techniques. We divide them into two subsections: the first one is oriented to the combinations for OVO decomposition where the aggregation is made from a score matrix; the second one reviews the combinations for OVA scheme, where the outputs of the classifiers are given by a score vector.

A more extensive and detailed description of these methods can be found in the web page ⟨http://sci2s.ugr.es/ovo-ova⟩. A complementary PDF file with the original source paper descriptions

is presented in the web page named "Aggregation schemes for binarization techniques. Methods' Description".

3.1. Aggregations in one-vs-one

In this subsection, we briefly describe the aggregation methods to obtain the predicted class from a score matrix obtained from the classifiers of an OVO decomposition scheme, that we have employed in the experimental study. A short description of each method follows:

- *Voting strategy* (*VOTE*) (*also called binary voting and Max-Wins rule* [29]): Each binary classifier gives a vote for the predicted class. The votes received by each class are counted and the class with the largest number of votes is predicted:

$$Class = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} s_{ij}, \tag{3}$$

  where s_{ij} is 1 if $r_{ij} > r_{ji}$ and 0 otherwise.
- *Weighted voting strategy* (*WV*): Each binary classifier votes for both classes. The weight for the vote is given by the confidence of the classifier predicting the class. The class with the largest sum value is the final output class:

$$Class = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} r_{ij} \tag{4}$$

- *Classification by pairwise coupling* (*PC*) [39]: This method estimates the joint probability for all classes from the pairwise class probabilities of the binary classifiers. Hence, when $r_{ij} = \text{Prob}(Class_i | Class_i \text{ or } Class_j)$, the method finds the best approximation of the class posterior probabilities $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)$ according to the classifiers outputs. The class with the largest posterior probability is predicted:

$$Class = \arg \max_{i=1,\ldots,m} \hat{p}_i \tag{5}$$

  To compute the posterior probabilities the Kullback–Leibler (KL) distance between r_{ij} and μ_{ij} is minimized:

$$l(\mathbf{p}) = \sum_{1 \le j \ne i \le m} n_{ij} r_{ij} \log \frac{r_{ij}}{\mu_{ij}} = \sum_{i<j} n_{ij} \left(r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1-r_{ij}) \log \frac{1-r_{ij}}{1-\mu_{ij}} \right), \tag{6}$$

  where $\mu_{ij} = p_i/(p_i+p_j)$, $r_{ji} = 1 - r_{ij}$ and n_{ij} is the number of training data in the ith and jth classes.
- *Decision directed acyclic graph* (*DDAG*) [59]: DDAG method constructs a rooted binary acyclic graph where each node is associated to a list of classes and a binary classifier. In each level a classifier discriminates between two classes, and the class that is not predicted is removed. The last class remaining on the list is the final output class.
- *Learning valued preference for classification* (*LVPC*) [45,44]: This method considers the score matrix as a fuzzy preference relation; based on fuzzy preference modeling, the original relation is decomposed into three new relations with different meanings, the strict preference, the conflict and the ignorance. A decision rule based on voting strategy is proposed to obtain the output class from them:

$$Class = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} P_{ij} + \frac{1}{2} C_{ij} + \frac{N_i}{N_i + N_j} I_{ij}, \tag{7}$$

  where N_i is the number of examples from class i in the training data (and hence, an unbiased estimate of the class probability), C_{ij} is the degree of conflict (the degree to which both classes are supported), I_{ij} is the degree of ignorance (the degree to which none of the classes are supported) and finally, P_{ij} and P_{ji} are, respectively, the strict preference for i and j. Preference, confidence and ignorance degrees are computed as follows:

$$\begin{aligned} P_{ij} &= r_{ij} - \min\{r_{ij}, r_{ji}\} \\ P_{ji} &= r_{ji} - \min\{r_{ij}, r_{ji}\} \\ C_{ij} &= \min\{r_{ij}, r_{ji}\} \\ I_{ij} &= 1 - \max\{r_{ij}, r_{ji}\} \end{aligned} \tag{8}$$

- *Preference relations solved by Non-Dominance Criterion* (*ND*) [24,25]: The Non-Dominance Criterion was originally defined for decision making with fuzzy preference relations [56]. In this case, as in LVPC, the score matrix is considered as a fuzzy preference relation. The relation has to be normalized. Then the degree of non-dominance is computed (the degree to which the class i is dominated by none of the remaining classes) and the class with the largest degree is predicted.
- *Binary tree of classifiers* (*BTC*): Binary Tree of SVM (BTS) [23], easily can be extented to any type of binary classifier. The idea behind this method is to reduce the number of classifiers and increase the global accuracy using some of the binary classifiers that discriminate between two classes, to distinguish other classes at the same time. The tree is constructed recursively and in a similar way to the DDAG approach, each node has associated a binary classifier and a list of classes. But in this case, the decision of the classifier can distinguish other classes as well as the pair of classes used for training. So, in each node, when the decision is done, more than one classes can be removed from the list. In order to avoid false assumptions, a probability is used when the examples from a class are near the discriminant boundary, so the class cannot be removed from the lists in the following level.
- *Nesting one-vs-one* (*NEST*) [50,49]: This method is directly developed to tackle the unclassifiable region produced in voting strategy (it is easy to see that in a three class problem, if each binary classifier votes for a different class, there is no winner, so some tie-breaking technique has to be applied). Nesting OVO uses the voting strategy, but when there exist examples within the unclassifiable region, a new OVO system is constructed using only the examples in the region in order to make them classifiable. This process is made until no examples remain in the unclassifiable region of the nested OVO.
- *Wu, Lin and Weng probability estimates by pairwise coupling approach* (*PE*) [76]: PE is similar to PC, which also estimates the posterior probabilities ($\mathbf{p}$) of each class starting from the pairwise probabilities. In this case, while the decision rule is equivalent (predicting the class with the largest probability), the optimization formulation is different. PE optimizes the following problem:

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \le j \ne i \le m} (r_{ji} p_i - r_{ij} p_j)^2 \quad \text{subject to} \quad \sum_{i=1}^{k} p_i = 1, p_i \ge 0, \forall i \tag{9}$$

3.2. Aggregations in one-vs-all

In this subsection, we briefly describe the combination methods to obtain the predicted class from a score vector from an OVA scheme output. In this case the aggregation is developed to deal with the ties when more than one classifiers give a positive answer, in other case the answer is given by the classifier giving a positive answer. A short description of each method which we have employed in the experimental study follows:

- *Maximum confidence strategy* (*MAX*): It is similar to the weighted voting strategy from OVO systems, the output class

is taken from the classifier with the largest positive answer:

$$Class = \arg \max_{i=1,\ldots,m} r_i \quad (10)$$

- *Dynamically ordered one-vs-all* (*DOO*) [41]: This method does not base its decision on the confidence of OVA classifiers. In this case, a Naïve Bayes classifier is also trained (using samples from all classes) together with all other classifiers. This new classifier establishes the order in which the OVA classifiers are executed for a given pattern. Then, the instance is submitted to each OVA classifier in that order until a positive answer is obtained, which indicates the predicted class. This is done dynamically for each example. In this manner, ties are avoided a priori by the Naïve Bayes classifier instead of relying on the degree of confidence given by the outputs of the classifiers.

4. Experimental framework

In this section, we present the set-up of the experimental framework used to develop the experiments in Section 5. We first describe the algorithms that we have selected to use as base classifiers in the study in Section 4.1. Section 4.2 describes the measures employed to evaluate the performance of the algorithms analysed in this paper. Next, we present the statistical tests applied to compare the results obtained with the different aggregations and decomposition techniques in Section 4.3. Afterwards, we provide details of the real-world problems chosen for the experimentation in Section 4.4 and the configuration parameters of the base learners and aggregation strategies in Section 4.5. Finally, Section 4.6 presents the information shown at the Web page associated with the paper.

4.1. Algorithms used for the study

In the empirical study our aim is to compare the different combination methods for the OVO and OVA schemes presented in Section 3. For this purpose, we have selected several well-known Machine Learning algorithms as base classifiers. If the original learning algorithm has the ability to manage multi-class problems without decomposition techniques, we analyse the behaviour of applying a decomposition method instead of using the original multi-class strategy.

Specifically, the selected algorithms are the following ones:

- *SVM* [73] maps the original input space into a high-dimensional feature space via a certain kernel function (avoiding the computation of the inner product of two vectors). In the new feature space, the optimal separating hyperplane with maximal margin is determined in order to minimize an upper bound of the expected risk instead of the empirical risk. We use SMO [58] training algorithm to obtain the SVM base classifiers.
- *C4.5* [62] is a decision tree generating algorithm. It induces classification rules in the form of decision trees from a set of given examples. The decision tree is constructed top-down using the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is the one used to make the decision.
- *kNN* [55] *k-nearest neighbours* finds a group of k instances in the training set, which are closest to the test pattern. The predicted class label is based on the predominance of a particular class in this neighbourhood. The distance and the number of neighbours are key elements of this algorithm.
- *Ripper* [19] *repeated incremental pruning to produce error reduction* builds a decision list of rules to predict the corresponding class for each instance. The list of rules is grown one by one and immediately pruned. Once a decision list for a given class is completely learned, an optimization stage is performed.
- *PDFC* (*Positive definite fuzzy classifier*) [16] constructs a fuzzy rule-based classification system extracting fuzzy rules from trained SVM. Since the learning process minimizes an upper bound on the expected risk instead of the empirical risk, the classifier usually has a good generalization ability.

The choice of these learning algorithms for classification was made on the basis of their good behaviour in a large number of real problems. Also, we have to point out that both SVM and PDFC do not have multi-category support in their original definition. Many approaches have been made to extend SVM to multiple classes (see [42]), but none of them have been established as a standard technique, mainly because they do not present real advantages to decomposition strategies that are used in SVM community for multi-classification.

4.2. Performance measures

In this work, we evaluate performance with multi-class data sets. However, in the literature most of the performance measures are designed only for two-class problems [69,27]. There is a big amount of well-known accuracy measures for two-class problems: classification rate (accuracy), precision, sensitivity, specificity, G-mean [11], F-score [10], AUC [43], Youden's index γ [77] and Cohen's kappa [13]. Different measures usually allow to observe different behaviours [26], this increases the strength of the empirical study in such a way that more complete conclusions can be obtained from different (not opposite, yet complementary) deductions.

Some of the accuracy measures originally designed for two-class problems have been adapted for multi-class. For example, in [48] an approximating multi-class ROC analysis is proposed, theoretically possible, but practically impossible for its computational complexity when the number of classes increases. There are two measures whose simplicity and successful application for both binary and multi-class problems have made them widely used. They are the classification rate and Cohen's kappa measures, which we explain hereafter:

- *Classification rate* also called accuracy rate, is the number of correctly classified instances (successful hits) relative to the total number of classified instances. It has been by far the most commonly used metric for assessing the performance of classifiers for years [5,75].
- *Cohen's kappa* is an alternative measure to *classification rate*, since it compensates for random hits [18,68]. In contrast with classification rate, kappa evaluates the portion of hits that can be attributed to the classifier itself (i.e., not to mere chance), relative to all the classifications that cannot be attributed to chance alone. An easy way of computing Cohen's kappa is by making use of the resulting confusion matrix (Table 1) in a classification task.

  From this matrix, Cohen's kappa is computed as follows:

$$kappa = \frac{n\sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri}T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri}T_{ci}} \quad (11)$$

  where h_{ii} is the cell count in the main diagonal (the number of true positives for each class), n is the number of examples, m is the number of class labels and T_{ri} and T_{ci} are the rows' and columns' total counts, respectively ($T_{ri} = \sum_{j=1}^{m} h_{ij}, T_{ci} = \sum_{j=1}^{m} h_{ji}$). Cohen's kappa ranges from -1 (total disagreement)

Table 1
Confusion matrix for an m-class problem.

Correct class	Predicted class				
	C_1	C_2	…	C_m	Total
C_1	h_{11}	h_{12}	…	h_{1m}	T_{r1}
C_2	h_{21}	h_{22}	…	h_{2m}	T_{r2}
⋮			⋱		⋮
C_m	h_{m1}	h_{m2}	…	h_{mm}	T_{rm}
Total	T_{c1}	T_{c2}	…	T_{cm}	T

through 0 (random classification) to 1 (perfect agreement). Being a scalar, it is less expressive than the ROC curves applied to binary-class cases. However, for multi-class problems, kappa is a very useful, yet simple, meter for measuring a classifier's classification rate while compensating for random successes. The main difference between the classification rate and Cohen's kappa is the scoring of the correct classifications. Classification rate scores all the successes over all classes, whereas Cohen's kappa scores the successes independently for each class and aggregates them. The second way of scoring is less sensitive to randomness caused by a different number of examples in each class.

4.3. Statistical tests

Statistical analysis needs to be carried out in order to find significant differences among the results obtained by the studied methods [33]. We consider the use of non-parametric tests according to the recommendations made in [20,35,33,34], where a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers is presented. These tests are used due to the fact that the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose credibility [20].

- For pairwise comparisons, we use the Wilcoxon paired signed-rank test [74] as a non-parametric statistical procedure to perform pairwise comparisons between two algorithms.
- For multiple comparisons, we use the Iman–Davenport test [67] to detect statistical differences among a group of results and the Shaffer post-hoc test [66] in order to find out which algorithms are distinctive among an $n \times n$ comparison. The post-hoc procedure allows us to know whether a hypothesis of comparison of means could be rejected at a specified level of significance α. However, it is very interesting to compute the p-value associated with each comparison, which represents the lowest level of significance of a hypothesis that results in a rejection. In this manner, we can know whether two algorithms are significantly different and how different they are.

These tests are suggested in the studies presented in [20,35,34], where its use in the field of machine learning is highly recommended. Any interested reader can find additional information on the Website http://sci2s.ugr.es/sicidm/, together with the software for applying the statistical tests.

Considering the ratio of the number of data sets to the number of methods that we compare along this paper, we fix the significance level $\alpha = 0.1$ for all comparisons.

Furthermore, we consider the average ranking of the algorithms in order to show graphically how good a method is with respect to its partners. This ranking is obtained by assigning a position to each algorithm depending on its performance for each data set. The algorithm that achieves the best accuracy in a specific data set will have the first ranking (value 1); then, the algorithm with the second best accuracy is assigned rank 2, and so forth. This task is carried out for all data sets and finally an average ranking is computed as the mean value of all rankings.

4.4. Data sets

In the study, we selected nineteen data sets from the UCI repository [9]. Table 2 summarizes the properties of the selected data sets. It shows, for each data set, the number of examples (#Ex.), the number of attributes (#Atts.), the number of numerical (#Num.) and nominal (#Nom.) attributes and the number of classes (#Cl.). Some of the largest data sets (nursery, page-blocks, penbased, satimage shuttle and led7digit) were stratified sampled at 10% in order to reduce the computational time required for training. In the case of missing values (autos, cleveland and dermatology) we removed those instances from the data set before doing the partitions.

The selection of this data sets has been carried out according to the premise of having more than three classes and a good behaviour with all the base classifiers, that is, considering an average accuracy higher than the 50%. Our aim is to define a general classification framework where we can develop our experimental study trying to find which methods are the most robust, in such a way that the extracted conclusions are valid for general multi-classification problems. Obviously, there exist some special situations such as the scalability (the increasing of the number of classes, variables or instances in the data sets), the presence of noise or the existence of data-fractures that are out of the scope of this paper. This will allow us to make a good analysis based on data sets with a large representation of classes and without noise from data sets with low classification rate, in such a way that we obtain more meaningful results from a multi-classification point-of-view.

Accuracy rate and kappa metric estimates were obtained by means of a 5-fold cross-validation, that is, the data set was split into 5 folds, each one containing 20% of the patterns of the data set. For each fold, the algorithm was trained with the examples contained in the remaining folds and then tested with the current fold. The data partitions used in this paper can be found in KEEL-data set repository [2] and in the website associated with this paper (http://sci2s.ugr.es/ovo-ova/).

Table 2
Summary description of data sets.

Data set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Car	1728	6	6	0	4
Lymphography	148	18	3	15	4
Vehicle	846	18	18	0	4
Cleveland	297	13	5	8	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Autos	159	25	15	10	6
Dermatology	366	33	1	32	6
Flare	1389	10	0	10	6
Glass	214	9	9	0	6
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Shuttle	2175	9	9	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Led7digit	500	7	0	7	10
Penbased	1099	16	16	0	10
Yeast	1484	8	8	0	10
Vowel	990	13	13	0	11

Table 3
Parameter specification for the base learners employed in the experimentation.

Algorithm	Parameters
SVM	$C=1.0$ Tolerance parameter=0.001 Epsilon=1.0E−12 Kernel type=polynomial Polynomial degree=1 Fit logistic models=true
C4.5	Prune=true Confidence level=0.25 Minimum number of item-sets per leaf=2
1NN	$k=1$ Distance metric=heterogeneous value difference metric (HVDM)
3NN	$k=3$ Distance metric=heterogeneous value difference metric (HVDM)
Ripper	Size of growing subset=66% Repetitions of the optimization stage=2
PDFC	C=100.0 Tolerance parameter=0.001 Epsilon=1.0E−12 Kernel type=polynomial Polynomial degree=1 PDRF type=Gaussian

4.5. Parameters

The configuration parameters for the base classifiers are shown in Table 3. The selected values are common for all problems, and they were selected according to the recommendation of the corresponding authors of each algorithm, which are also the default parameters' setting included in the KEEL[1] software [3,2] that we used to develop our experiments. We considered two configurations for *k*NN algorithm, the first one with one neighbour and the second one with three neighbours, so we analysed them as two different base classifiers 1NN and 3NN. Also, note that we treat nominal attributes in SVM and PDFC as scalars to fit the data into the systems using a polynomial kernel.

Although we acknowledge that the tuning of the parameters for each method on each particular problem could lead to better results (mainly in SVM and PDFC), we preferred to maintain a baseline performance of each method as the basis for comparison. Since we are not comparing base classifiers among them, our hypothesis is that the methods that win on average on all problems would also win if a better setting was performed. Furthermore, in a framework where no method is tuned, winner methods tend to correspond to the most robust, which is also a desirable characteristic.

Some of the aggregation methods based their decision on the confidence of the predictions from the base classifiers. We obtain the confidence for each classifier as follows:

- *SVM*: Logistic model parameter is set to True in order to use the probability estimates from the SVM [58] as the confidence for the predicted class.
- *C4.5*: Confidence is obtained from the accuracy of the leaf that makes the prediction. The accuracy of a leaf is the percentage of correctly classified train examples from the total number of covered train instances.
- *kNN*: We use the following equation to estimate the confidence of *k*NN:

$$Confidence = \frac{\sum_{l=1}^{k} e_l/d_l}{\sum_{l=1}^{k} 1/d_l} \tag{12}$$

where d_l is the distance between the input pattern and the lth neighbour and $e_l=1$ if the neighbour l is from the class and 0 otherwise. Note that when $k>1$, the probability estimate depends on the distance from the neighbours of each class, hence the estimation is not restricted to a few values (when only the numbers of neighbours from each class are considered, a multi-valued result is obtained, which is not desired).
- *Ripper*: The confidence is taken from the accuracy of the rule used in the prediction, that is, the percentage of correctly classified train instances from the total number of train instances classified.
- *PDFC*: The confidence depends on the prediction of the classifier, confidence=1 is given for the predicted class.

For the models whose confidence degree is included in {0,1} such as 1NN and PDFC, note that some aggregation methods are equivalent, i.e. VOTE=WV=LVPC. In these cases, we consider VOTE as their representative.

Regarding the aggregation strategies, Binary Tree of Classifiers has a parameter to define the reasonability of reassignment. In this study we set the parameter $\delta=5\%$ which was the value recommended by the authors in [23].

Finally, in the strategies where ties are possible, the majority class is predicted, if the tie continues, then the class is selected randomly.

4.6. Web page associated to the paper

In order to provide additional material to the paper content, we have developed a Web page at (http://sci2s.ugr.es/ovo-ova) in which we have included the following information:

- A wider description of the state-of-the-art on aggregations for OVO and OVA strategies in the report named "Aggregation schemes for binarization techniques. Methods' description",
- The data sets partitions employed in the paper,
- Finally, we include some Excel files with the train and test results for all the algorithms so that any interested researcher can use them to include their own results and extend the present study.

5. Experimental study

In this section, we present the results of the experimental study. We will answer to the following questions:

1. Should we do binarization? How should we do it?
2. Which is the most appropriate aggregation for each decomposition scheme?

Thereby the study is divided into two parts, each one dedicated to a question. Since the main objective of this paper is the analysis of the different combinations, we will try to answer these questions in an upside-down manner, starting from the second one. Hence we will first analyse the aggregations for OVO and OVA strategies and then we will go through the complementary analysis comparing the best OVO and OVA aggregations against the baseline classifiers.

We develop the analysis studying the performance of the aggregation methods and their synergy with the different base

[1] 〈http://www.keel.es〉

classifiers considered in the study. To make a proper study, we develop an analysis guided by non-parametric statistical tests explained in Section 4.3. We want to verify if there exists a most suitable ensemble technique that is independent of the base classifier considered. If the appropriate aggregation is base learner dependent, then we investigate which strategies fit better in each base classifier. Finally, we will also check the goodness of ensemble methods with respect to the original classifiers (when they support multiple categories intrinsically). Therefore, we will fill out a complete study of the binarization framework in both directions, the decomposition and the combination, but centring our analysis in the different aggregations for each decomposition.

Through the experimental study, we show the average results for each method and metric in testing phase; in order to observe the complete results please refer to the web-page associated with this paper where we show both train and test results for every data set and base learner.

5.1. Which is the most appropriate aggregation for each decomposition scheme?

Our first objective is to study the behaviour of the different aggregations in OVO and OVA schemes. To do so, we divide the analysis into two parts, one for each scheme.

Before starting with the statistical analysis, we present in Table 4 the average results for each method and metric (accuracy rate and Cohen's kappa) in testing phase ($\pm$ for standard deviation). The average rank and the rank position are also included, measured for both metrics in the test scenario. The ranks are computed for each decomposition scheme independently. Finally, the best global result in each decomposition is stressed through **bold-face**.

5.1.1. Analysis of aggregations for OVO scheme

Observing Table 4, there is no aggregation method that excels from all others with the different base classifiers. Therefore, we analyse first whether there exist significant differences in using one or an other combination within each base classifier.

With respect to SVM, the results of the statistical analysis do not reject the null hypothesis that all the methods are equivalent, since the p-value returned by the Iman–Davenport test is higher than our α-value (0.1) for both performance measures, accuracy (p-value 0.58321) and kappa (p-value 0.71011). Although there are no statistical differences among the methods, regarding the average performance and the ranking of the different schemes, we may stress the good behaviour of NEST and VOTE. Moreover, observing the differences of ranking between accuracy and kappa, we may conclude that NEST approach is more robust while VOTE looses precision considering kappa metric. In addition BTC has the worst means in both metrics and also is the worst ranked, but recall that the differences among these methods are not statistically significant.

Regarding C4.5 decision tree as base classifier, the Iman–Davenport test applied to the results of all methods shows significant differences among the algorithms using both kappa (p-value 0.03341) and accuracy (p-value 0.07101) performance measures. We analyse the differences found with both metrics by applying Shaffer's post-hoc tests summarized in Table 5. In this table a "+" symbol implies that the first algorithm is statistically better than the confronting one, whereas "−" implies the contrary; "=" means that the two algorithms compared have no significant differences. In brackets the adjusted p-value associated with each comparison is shown. Note that the last hypothesis denoted as "Others" summarizes the rest of possible pairwise comparisons between methods where no statistical differences were found (p-value = 1.00).

The table shows that the greatest differences are with respect to NEST and DDAG methods, the most robust aggregations being WV, PC and LVPC. It is interesting to note that the NEST method, with which SVM has a great behaviour, it shows the worst performance in this case. Possibly it is due to the overfitting problems of C4.5 and the nature of NEST, where new trees are constructed with a low number of examples in the unclassifiable region. Hence, these examples are overlearned in the nested classifiers.

As we state in Section 4.5, when we consider 1NN as base classifier, the confidence of the base classifier is always 1 for the predicted class, and therefore the results using VOTE, WV and LVPC are completely equivalent. For this reason, we only consider VOTE as representative for these strategies. Statistical study with Iman–Davenport test rejects the null hypothesis of equivalence between the algorithms with accuracy and Cohen's kappa since the returned p-value is lower than the significance level (accuracy 0.05370 and kappa 0.00136). Hence, there exist significant differences among the methods and we execute the Shaffer post-hoc tests for both measures; the results are shown in Table 6.

In this case, the method having the best behaviour is PE, nearly followed by PC. Both methods compute the probability estimates prior to deciding the output class, which seems to be an appropriate approximation when the confidences given by the classifier are in {0,1} (looking also at the results from PDFC, where PC is the best ranked method). Once again, NEST method does not perform as well as when SVM is used as base classifier, and it is possibly because the nearest neighbour rule is not a proper rule when only the examples in the unclassifiable region are taken into account. Moreover, ND is the worst aggregation due to the confidence given by 1NN and the procedure used to compute the output. There are more ties than usual and hence a random guess does not produce good results.

Considering 3NN, we can analyse the differences of using a priori more suitable confidence degrees. Regarding the statistical test, the Iman–Davenport test does not reject the equivalence between the algorithms using either accuracy measure (p-value 0.39094) or kappa measure (p-value 0.24752). In this case, despite no significant differences being found, ND behaviour stands out, mainly taking into account that its performance with other base classifiers was not so stressed. Recalling the results with 1NN we should note the positive synergy between OVO strategies and base classifiers providing a confidence measure different from the total confidence.

Concerning Ripper rule learning algorithm, we execute the Iman–Davenport test for both performance measures, obtaining a p-value of 0.00036 and 0.00366 with accuracy and kappa, respectively. This means that the null hypothesis of equivalence between algorithms is rejected, so we proceed with the Shaffer test. Table 7 show the results from both tests.

We conclude that mainly WV and also LVPC perform better than other approaches. Even other methods such as DDAG and BTC, which are not appropriate for Ripper, are outperformed with statistical significance.

Finally PDFC, which is similar to SVM, is designed to tackle binary problems and thereby decomposition strategies are used to face up multi-class problems. Besides, analogously to 1NN, there is no way to obtain an appropriate confidence estimate varying in the unit interval, so total confidence is given to the predicted class. In these conditions, VOTE, WV and LVPC are equivalent. Hence, we consider VOTE as representative of the group and we only consider comparison between the different ensembles formed by the different aggregations.

In spite of the stand out behaviour of VOTE, PC and PE with both measures (which is in concordance with the results obtained

Table 4
Average accuracy and kappa results in test for each base classifier and binarization technique.

Method	Aggregation	SVM		C4.5		1NN		3NN		Ripper		PDFC	
		Acc_{tst}	Avg. rank	Acc_{tst}	Avg. rank	Acc_{tst}	Avg. rank	Acc_{tst}	Avg. rank	Acc_{tst}	Avg. rank	Acc_{tst}	Avg. rank
Base	–	–	–	80.51 ± 3.85	–	81.24 ± 2.98	–	81.54 ± 2.65	–	76.52 ± 4.00	–	–	–
OVO	VOTE	81.14 ± 3.22	**4.37 (1)**	81.57 ± 3.29	4.63 (4)	82.06 ± 3.38	3.82 (3)	83.00 ± 2.92	5.05 (6)	80.57 ± 3.17	3.89 (3)	**84.33 ± 3.10**	3.37 (2)
	WV	81.05 ± 2.92	5.08 (6)	**81.59 ± 3.28**	3.97 (2)	–	–	**83.11 ± 2.87**	4.47 (3)	80.54 ± 3.03	3.87 (2)	–	–
	DDAG	81.01 ± 3.28	5.39 (8)	81.02 ± 3.56	6.21 (9)	81.86 ± 3.31	4.32 (5)	82.73 ± 2.83	5.87 (8)	77.62 ± 3.61	7.08 (9)	84.05 ± 3.00	3.71 (3)
	PC	81.08 ± 2.89	5.29 (7)	81.49 ± 3.32	4.34 (3)	82.26 ± 3.33	3.21 (2)	83.00 ± 2.96	5.11 (7)	80.33 ± 3.30	4.87 (5)	84.12 ± 3.05	**3.29 (1)**
	LVPC	81.14 ± 3.11	4.50 (3)	81.57 ± 3.28	**3.87 (1)**	–	–	83.07 ± 2.79	4.61 (4)	**80.58 ± 3.16**	**3.68 (1)**	–	–
	ND	81.01 ± 3.15	4.92 (5)	81.12 ± 3.24	5.58 (6)	81.48 ± 3.51	4.97 (7)	83.07 ± 2.93	**4.29 (1)**	79.38 ± 3.27	5.29 (7)	84.05 ± 2.96	4.68 (6)
	BTC	80.82 ± 3.24	6.18 (9)	81.22 ± 2.87	5.61 (7)	82.21 ± 3.12	3.89 (4)	82.99 ± 2.98	5.00 (5)	79.19 ± 3.07	6.39 (8)	84.24 ± 3.01	4.29 (5)
	NEST	**81.14 ± 3.32**	4.47 (2)	81.20 ± 3.47	5.74(8)	81.68 ± 3.47	4.68 (6)	82.67 ± 2.94	6.16 (9)	80.01 ± 3.50	5.08 (6)	83.88 ± 3.02	4.89 (7)
	PE	81.03 ± 3.35	4.79 (4)	81.42 ± 3.22	5.05 (5)	**82.30 ± 3.11**	**3.11 (1)**	83.11 ± 2.94	4.45 (2)	80.07 ± 3.08	4.84 (4)	84.06 ± 3.04	3.76 (4)
OVA	MAX	78.66 ± 3.00	1.53 (2)	78.01 ± 4.19	1.84 (2)	81.18 ± 4.51	1.63 (2)	82.75 ± 4.29	1.58 (2)	78.30 ± 4.94	1.71 (2)	**83.59 ± 3.12**	**1.39 (1)**
	DOO	**78.75 ± 3.15**	**1.47 (1)**	**78.78 ± 4.36**	**1.16 (1)**	**81.77 ± 4.45**	**1.37 (1)**	**82.76 ± 4.38**	**1.42 (1)**	**79.12 ± 4.67**	**1.29 (1)**	83.01 ± 3.10	1.61 (2)

Method	Aggregation	SVM		C4.5		1NN		3NN		Ripper		PDFC	
		$Kappa_{tst}$	Avg. rank	$Kappa_{tst}$	Avg. rank	$Kappa_{tst}$	Avg. rank	$Kappa_{tst}$	Avg. rank	$Kappa_{tst}$	Avg. rank	$Kappa_{tst}$	Avg. rank
Base	–	–	–	0.7203 ± 0.0554	–	0.7369 ± 0.0475	–	0.7335 ± 0.0452	–	0.6799 ± 0.0554	–	–	–
OVO	VOTE	0.7233 ± 0.0548	4.82 (2)	0.7331 ± 0.0490	5.16 (5)	0.7419 ± 0.0535	3.84 (3)	0.7507 ± 0.0500	5.03 (6)	**0.7250 ± 0.0475**	4.26 (3)	**0.7677 ± 0.0538**	3.63 (2)
	WV	0.7229 ± 0.0506	5.05 (6)	**0.7348 ± 0.0485**	**3.76 (1)**	–	–	0.7519 ± 0.0487	4.71 (3)	0.7249 ± 0.0455	**3.68 (1)**	–	–
	DDAG	0.7230 ± 0.0555	5.11 (7)	0.7304 ± 0.0535	5.92 (8)	0.7402 ± 0.0522	3.89 (4)	0.7479 ± 0.0487	5.87 (8)	0.6957 ± 0.0489	6.42 (8)	0.7659 ± 0.0518	3.97 (5)
	PC	0.7234 ± 0.0520	5.18 (8)	0.7341 ± 0.0493	4.13 (3)	0.7449 ± 0.0525	3.00 (2)	0.7505 ± 0.0505	4.89 (5)	0.7227 ± 0.0483	4.61 (5)	0.7670 ± 0.0529	**3.11 (1)**
	LVPC	0.7211 ± 0.0531	5.03 (5)	0.7341 ± 0.0488	4.03 (2)	–	–	0.7496 ± 0.0475	5.18 (7)	0.7246 ± 0.0469	4.00 (2)	–	–
	ND	0.7225 ± 0.0533	4.82 (2)	0.7286 ± 0.0489	5.53 (7)	0.7340 ± 0.0556	5.37 (7)	0.7524 ± 0.0500	**4.03 (1)**	0.7098 ± 0.0479	5.92 (7)	0.7625 ± 0.0524	5.32 (7)
	BTC	0.7204 ± 0.0551	6.05 (9)	0.7297 ± 0.0428	5.42 (6)	0.7438 ± 0.0498	4.29 (5)	0.7519 ± 0.0514	4.87 (4)	0.7087 ± 0.0476	6.58 (9)	0.7668 ± 0.0527	3.79 (3)
	NEST	**0.7243 ± 0.0559**	**4.03 (1)**	0.7291 ± 0.0514	6.34 (9)	0.7366 ± 0.0547	4.79 (6)	0.7461 ± 0.0505	6.24 (9)	0.7195 ± 0.0496	4.97 (6)	0.7641 ± 0.0514	4.37 (6)
	PE	0.7228 ± 0.0537	4.92 (4)	0.7330 ± 0.0480	4.71 (4)	**0.7453 ± 0.0497**	**2.82 (1)**	**0.7526 ± 0.0499**	4.18 (2)	0.7193 ± 0.0457	4.55 (4)	0.7653 ± 0.0524	3.82 (4)
OVA	MAX	0.6868 ± 0.0553	1.55 (2)	0.6826 ± 0.0629	1.89 (2)	0.7298 ± 0.0705	1.63 (2)	**0.7481 ± 0.0695**	1.58 (2)	0.6896 ± 0.0743	1.79 (2)	**0.7556 ± 0.0589**	**1.37 (1)**
	DOO	**0.6868 ± 0.0565**	**1.45 (1)**	**0.6938 ± 0.0649**	**1.11 (1)**	**0.7368 ± 0.0701**	**1.37 (1)**	0.7473 ± 0.0710	**1.42 (1)**	**0.7004 ± 0.0716**	**1.21 (1)**	0.7478 ± 0.0587	1.63 (2)

Table 5
Shaffer test for OVO aggregations with C4.5 as base classifier.

i	Hypothesis	p-Value
(a) *Accuracy*		
1	DDAG vs LVPC	=(0.30205)
2	WV vs DDAG	=(0.33095)
3	LVPC vs NEST	=(0.99344)
4	DDAG vs PC	=(0.99344)
5–36	Others	=(1.00)
(b) *Kappa*		
1	WV vs NEST	=(0.13327)
2	LVPC vs NEST	=(0.25625)
3	PC vs NEST	=(0.35983)
4	WV vs DDAG	=(0.42437)
5	DDAG vs LVPC	=(0.92314)
6–36	Others	=(1.00)

Table 6
Shaffer test for OVO aggregations with 1NN as base classifier.

i	Hypothesis	p-Value
(a) *Accuracy*		
1	ND vs PE	=(0.16127)
2	PC vs ND	=(0.17822)
3	NEST vs PE	=(0.36406)
4	PC vs NEST	=(0.53247)
5–21	Others	=(1.00)
(b) *Kappa*		
1	ND vs PE	−(0.00568)
2	PC vs ND	+(0.01090)
3	NEST vs PE	−(0.07293)
4	PC vs NEST	=(0.16011)
5	VOTE vs ND	=(0.44140)
6	BTC vs PE	=(0.53247)
7	DDAG vs ND	=(0.53247)
8	PC vs BTC	=(0.72377)
9–21	Others	=(1.00)

Table 7
Shaffer tests for OVO aggregations with Ripper as base classifier.

i	Hypothesis	p-Value
(a) *Accuracy*		
1	DDAG vs LVPC	−(0.00479)
2	WV vs DDAG	+(0.00846)
3	VOTE vs DDAG	+(0.00948)
4	LVPC vs BTC	+(0.06395)
5	WV vs BTC	=(0.12503)
6	VOTE vs BTC	=(0.13715)
7	DDAG vs PE	=(0.33095)
8	DDAG vs PC	=(0.35983)
9	DDAG vs NEST	=(0.68293)
10	DDAG vs ND	=(0.96824)
11–36	Others	=(1.00)
(b) *Kappa*		
1	WV vs BTC	+(0.04040)
2	WV vs DDAG	+(0.05792)
3	LVPC vs BTC	=(0.10365)
4	DDAG vs LVPC	=(0.18015)
5	VOTE vs BTS	=(0.25625)
6	WV vs ND	=(0.33095)
7	VOTE vs DDAG	=(0.42437)
8	BTC vs PE	=(0.63211)
9	PC vs BTC	=(0.73724)
10	LVPC vs ND	=(0.73724)
11	DDAG vs PE	=(0.78056)
12	DDAG vs PC	=(0.90184)
13–36	Others	=(1.00)

Table 8
Shaffer test for OVO aggregations with PDFC as base classifier (kappa).

i	Hypothesis	p-Value
1	PC vs ND	+(0.03383)
2	VOTE vs ND	=(0.24391)
3	ND vs BTC	=(0.44140)
4	ND vs PE	=(0.48511)
5	DDAG vs ND	=(0.83259)
6–21	Others	=(1.00)

with 1NN using the total confidence), the Iman–Davenport test does not reject the null hypothesis of equivalence for accuracy (p-value 0.13163), but it rejects for kappa (p-value $0.06318 < 0.1$). Therefore, there are significant differences between these methods; we execute the Shaffer post-hoc test for kappa measure and the results are shown in Table 8. Again VOTE (which also represents WV and LVPC), PC and PE have better performance on average, which is in agreement with the previous experiments with other classifiers.

Summarizing the results obtained from this analysis, we have shown that the most robust OVO ensemble models are formed with WV, PC, PE and LVPC approaches. In general, and mainly when the confidence estimates are different from the total confidence, there exist statistical differences in using one or another aggregation, hence it should be selected carefully. In any case, clearly the choice of the best aggregation scheme is base classifier dependant.

We also have to point out some special cases such as the positive synergy between ND and 3NN, and PC, PE with both 1NN and PDFC. This is due to the fact that 1NN and PDFC instead of giving a confidence degree for each class, only output the predicted class; besides more than being appropriate aggregations, probably they do not perform so bad as the ones that need accurate confidence estimates.

5.1.2. Analysis of aggregations for OVA scheme

Considering OVA decomposition, we have only two methods to be compared (note that to the best of our knowledge, no more approaches have been made). In this case, looking at Table 4, the most dominant aggregation is DOO, but we should study if their results are significantly better, and also why it performs better with all but PDFC base classifiers. To do so, in this case we only use Wilcoxon signed-rank test to detect differences since we are comparing two methods.

Table 9 presents the results for the Wilcoxon signed-rank tests. The results are more significant than in the previous section. DOO approach outperforms significantly MAX in three cases (C4.5, 1NN and Ripper) while the contrary occurs in one case (PDFC); in the last case (3NN) DOO behaviour is better (having more rank), but the differences are not significant.

These results have a direct conclusion; DOO approach performs better when the base classifiers accuracy is not better than the Naïve Bayes ones. Hence, in those cases, it can help selecting the most appropriate classifier to use dynamically. However, when the base classifier has enough accuracy, the previous use of the Naïve Bayes classifier can distort the decision, reducing the performance of using the MAX strategy.

5.2. Should we do binarization? How should we do it?

Within the framework of binarization techniques we have analysed which are the best or the most appropriate proposals for OVO and OVA decomposition schemes, observing good results in each case. However, an important question remains to be answered: should we do binarization? Thereby, we are going to

analyse the results obtained from the stressed OVO and OVA methods in contrast with the baseline classifiers to show their suitability to tackle multi-class problems, also when the base classifiers manage multiple categories inherently (C4.5, k NN and Ripper). Previous studies [30,31,63] have been done around this question, showing the goodness of binarization techniques, although we consider that it is important to remark these claims with an exhaustive experimental study with the appropriate statistical analysis. Anyway, we also study the differences of using one or another decomposition in both cases, when there does not exist a way to handle multi-class problems (SVM and PDFC) and when the base classifiers can face them intrinsically.

To do so, we select for each base classifier, the best OVO and OVA methods from the analysis of the previous sections, these will be the representatives of each approach for the comparison.

Table 9
Wilcoxon tests to compare MAX and DOO combinations in OVA scheme with different base classifiers. R^+ corresponds to the sum of the ranks for MAX scheme and R^- for DOO.

Base classifier	Measure	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p-Value
SVM	Accuracy	82	108	Not rejected	0.53213
	Kappa	86	104	Not rejected	0.75637
C4.5	Accuracy	14	176	Rejected for DOO	0.00179
	Kappa	11.5	178.5	Rejected for DOO	0.00118
1NN	Accuracy	55	135	Rejected for DOO	0.09097
	Kappa	55	135	Rejected for DOO	0.09097
3NN	Accuracy	75	115	Not rejected	0.73532
	Kappa	76	114	Not rejected	0.86577
Ripper	Accuracy	44.5	145.5	Rejected for DOO	0.04286
	Kappa	42	148	Rejected for DOO	0.03294
PDFC	Accuracy	130.5	59.5	Rejected for MAX	0.0464
	Kappa	138	52	Rejected for MAX	0.02799

Table 10
Representative ensembles for each base classifiers in the OVO vs OVA comparison.

	SVM	C4.5	1NN	3NN	Ripper	PDFC
OVO	$NEST_{ovo}$	WV_{ovo}	PE_{ovo}	ND_{ovo}	WV_{ovo}	PC_{ovo}
OVA	DOO_{ova}	DOO_{ova}	DOO_{ova}	DOO_{ova}	DOO_{ova}	MAX_{ova}

The criterion for choosing each method is based on the best performing method according to the differences supported by the statistical analysis. If no significant differences are found, then the best mean result is taken. Selected methods are shown in Table 10.

In Table 11 we show the average results for accuracy and Cohen's kappa for all the selected ensembles together with the original base classifier's results. The table summarizes the results shown in the previous sections. In this case, the average ranks are computed within each base classifier. For brevity, the results for every single data are available in the web-page associated with this paper.

An easy way to interpret these results is observing the average rankings. They are shown in Fig. 1(a) and (b) for accuracy and kappa, respectively, computed within each base classifier. The results for the test partitions are also depicted in Fig. 2(a) and (b) using box plot as representation scheme. Box plots proved to be a most valuable tool in data reporting, since they allow the graphical representation of the performance of the algorithms, indicating important features such as the median, extreme values and spread of values about the median in the form of quartiles.

From these figures, we have to point out that in most cases OVO strategy based ensembles perform better than OVA ones and the base classifiers. The box plot shows that their results are more robust in the sense that the boxes are more compact, and hence in spite of not being always the technique with the highest result, they behave more steadily in a larger amount of data sets. On the other hand OVA strategies result on accuracy are more questionable, since it often reduces their performance more than the others when kappa measure is considered. In general, OVO seems to be the most appropriate technique, but this claim should be proved by the proper statistical analysis that followed.

In the same manner as in the previous study, we will make use of Wilcoxon signed-rank test to detect differences between OVO and OVA strategies when the base classifiers do not support multi-class problems (SVM and PDFC). When we compare both strategies together with the results of the underlying classifier, we will execute the Iman–Davenport test to detect differences among the algorithms, and in case that they are found, we will proceed with Shaffer's post-hoc test.

In the first place, regarding the classifiers that need to use binarization techniques, Table 12 shows a comparison between OVO and OVA approaches. In these cases, OVO outperforms OVA with significant differences. In Section 6 we will discuss these

Table 11
Average accuracy and kappa results of the best OVO and OVA ensembles. If exist, the original algorithm results are shown.

Base classifier	Aggregation	Accuracy		Kappa	
		Test	Avg. rank	Test	Avg. rank
SVM	$NEST_{ovo}$	**81.14 ± 3.32**	**1.37 (1)**	**0.7243 ± 0.0559**	**1.32 (1)**
	DOO_{ova}	78.75 ± 3.15	1.63 (2)	0.6868 ± 0.0565	1.68 (2)
C4.5	C45	80.51 ± 3.85	2.05 (2)	0.7203 ± 0.0554	2.14 (2)
	WV_{ovo}	**81.59 ± 3.28**	**1.42 (1)**	**0.7348 ± 0.0485**	**1.21 (1)**
	DOO_{ova}	78.78 ± 4.36	2.53 (3)	0.6938 ± 0.0649	2.64 (3)
1NN	1NN	81.24 ± 2.98	**1.84 (1)**	0.7369 ± 0.0475	**1.82 (1)**
	PE_{ovo}	**82.30 ± 3.11**	2.05 (2)	**0.7453 ± 0.0497**	2.05 (2)
	DOO_{ova}	81.77 ± 4.45	2.11 (3)	0.7368 ± 0.0701	2.13 (3)
3NN	3NN	81.54 ± 2.65	2.24 (3)	0.7335 ± 0.0452	2.42 (3)
	ND_{ovo}	**83.07 ± 2.93**	**1.87 (1)**	**0.7524 ± 0.0500**	1.84 (2)
	DOO_{ova}	82.76 ± 4.38	1.89 (2)	0.7473 ± 0.0710	**1.74 (1)**
Ripper	Ripper	76.52 ± 4.00	2.61 (3)	0.6799 ± 0.0554	2.42 (3)
	WV_{ovo}	**80.54 ± 3.03**	**1.66 (1)**	**0.7249 ± 0.0455**	**1.58 (1)**
	DOO_{ova}	79.12 ± 4.67	1.74 (2)	0.7004 ± 0.0716	2.00 (2)
PDFC	PC_{ovo}	**84.12 ± 3.05**	**1.26 (1)**	**0.7670 ± 0.0529**	**1.26 (1)**
	MAX_{ova}	83.59 ± 3.12	1.74 (2)	0.7556 ± 0.0589	1.74 (2)

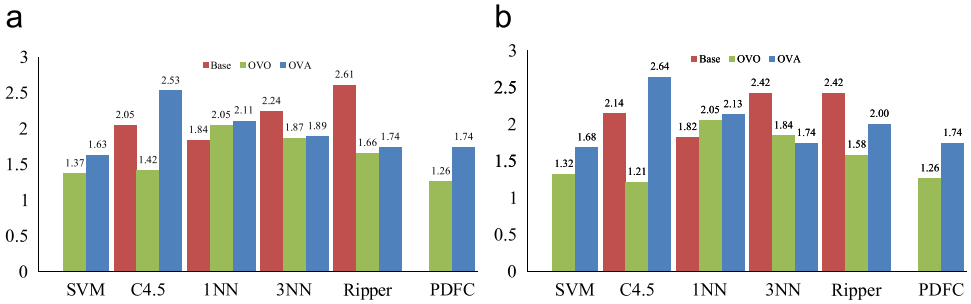

Fig. 1. Rankings of the OVA and OVO representatives for each base classifier and also base classifier's ranking: (a) ranking in accuracy and (b) ranking in Kappa.

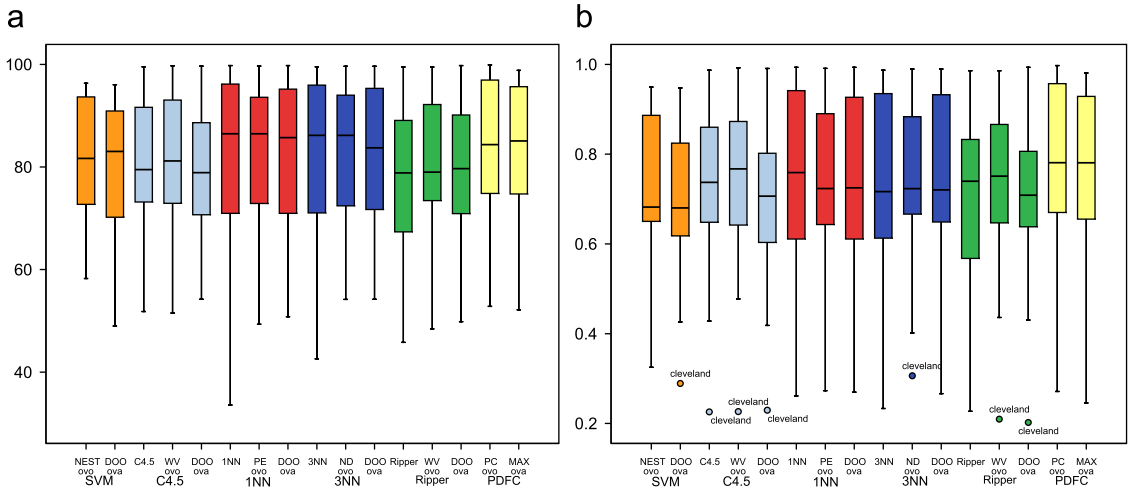

Fig. 2. Box plot representations for the results of the representative OVO and OVA methods with base classifier's results: (a) box plot for accuracy and (b) box plot kappa.

Table 12
Wilcoxon tests to compare OVO and OVA combinations with different base classifiers. R^+ corresponds to the sum of the ranks for OVO scheme and R^- for OVA.

Base classifier	Comparison	Measure	R^+	R^-	Hypothesis ($\alpha = 0.1$)	p-Value
SVM	$NEST_{ovo}$ vs DOO_{ova}	Accuracy	153	37	Rejected for $NEST_{ovo}$	0.01959
		Kappa	156	34	Rejected for $NEST_{ovo}$	0.0141
PDFC	PC_{ovo} vs MAX_{ova}	Accuracy	146	44	Rejected for PC_{ovo}	0.04014
		Kappa	147	43	Rejected for PC_{ovo}	0.03639

Table 13
p-Values returned by Iman–Davenport test. * indicates that the null hypothesis of equivalence is rejected with $\alpha = 0.1$.

	C4.5	1NN	3NN	Ripper
Accuracy	0.00134*	0.70296	0.45982	0.00296*
Kappa	0.00026*	0.61089	0.07585*	0.02982*

Table 14
Shaffer tests for 3NN based methods (kappa).

i	Hypothesis	p-Value
1	3NN vs DOO_{ova}	=(0.10487)
2	3NN vs ND_{ovo}	=(0.10487)
3	ND_{ovo} vs DOO_{ova}	=(0.74560)

results. At a first glance, in OVA a larger number of instances from different classes are considered; in most of the cases they produce more complex binary problems than the ones produced in OVO strategy, this fact together with the imbalanced data sets [40,70] created could be the hitch of OVA.

With respect to the rest of the base classifiers, we first execute the Iman–Davenport test to detect differences among the groups of three methods. We present the returned p-values for accuracy and kappa measures in Table 13.

Concerning the comparison between kNN based classifiers the hypothesis of equivalence is rejected only when 3NN with kappa measure is considered. Table 14 shows the results of the Shaffer post-hoc test considering kappa measures.

Despite no significant differences being found, OVO and OVA strategies nearly outperform the original classifier. When kNN is used as base classifier of the ensembles, the result is not as beneficial as before. This is possibly due to the prediction rule of kNN, although this does not mean that the results are worse when binarization is used. In mean accuracy and kappa, binarization techniques improve the original nearest neighbours, while in terms of ranks the differences are not statistically significant (except for 3NN with kappa), but they exist.

While considering kNN the hypothesis of equivalence is not always rejected; for C4.5 and Ripper significant differences are found with both metrics. Hence, we execute Shaffer's post-hoc test; in Tables 15 and 16 we present the results of the tests for

Table 15
Shaffer tests for C4.5 based methods.

i	Hypothesis	p-Value
(a) *Accuracy*		
1	WV_{ovo} vs DOO_{ova}	+(0.00197)
2	C4.5 vs WV_{ovo}	−(0.05158)
3	C4.5 vs DOO_{ova}	=(0.14429)
(b) *Kappa*		
1	WV_{ovo} vs DOO_{ova}	+(0.00057)
2	C4.5 vs WV_{ovo}	−(0.03496)
3	C4.5 vs DOO_{ova}	=(0.10476)

Table 16
Shaffer tests for Ripper based methods.

i	Hypothesis	p-Value
(a) *Accuracy*		
1	Ripper vs WV_{ovo}	−(0.01050)
2	Ripper vs DOO_{ova}	−(0.01050)
3	WV_{ovo} vs DOO_{ova}	=(0.80775)
(b) *Kappa*		
1	Ripper vs WV_{ovo}	−(0.02833)
2	Ripper vs DOO_{ova}	=(0.19437)
3	WV_{ovo} vs DOO_{ova}	=(0.19437)

C4.5 and Ripper, respectively (each table shows the tests for accuracy and kappa).

Regarding C4.5, using both measures the WV together with OVO stands out as the best one. It outperforms the OVA approach, with statistical differences and also the original C4.5 algorithm when kappa measure is considered (in accuracy the p-value is also low, which means a better overall behaviour).

Analogously, Ripper algorithm is significantly improved when an OVO decomposition with the WV aggregation is considered. In spite of no significant differences being found between OVO and OVA approaches, the former outperforms Ripper with significant differences with both measures, while the latter does not yield to do it with kappa measure. Hence, OVO approaches once again present a significant improvement with respect to the baseline classifiers and OVA approaches. Note that these results are in concordance with previous studies [30,31].

Concerning Ripper, we have to point out that the original algorithm manage multi-class problems in a similar way to OVA. It trains the rule-sets in a hierarchical manner, starting from the class with the lowest number of instances, and removing the classified instances in each new level. The study shows that it is not the best approach to tackle this type of problems with Ripper, probably because it does not make use of confidence scores in order to predict the correct class, but it predicts the first one with a positive answer. Also, it trusts first on the classifier that has been trained with the most imbalanced data set, which is more prone to errors.

Summarizing, from the analysis of these results, we may conclude that OVO methods are more suitable and robust than OVA ones, this conclusion is supported by the experiments that have shown OVO methods outperforming the original base classifier and OVA ones, while none of the OVA have yielded to do the contrary.

6. Discussion: lessons learned and future work

This paper has provided an exhaustive empirical analysis of the main ensemble methods for binary classifiers in multi-class problems, specifically the methods based on OVO and OVA strategies. We structured the analysis in two sections, studying the different ways in which the outputs of the underlying binary classifiers can be combined and then, filling up the analysis investigating the use of binarization techniques when the multi-class problem can also be faced up by a unique classifier.

From this study we emphasize seven important lessons learned:

a. The use of binarization techniques to deal with multi-class problems is beneficial when the problem can also be directly handled by a single base classifier. This claim is supported by the analysis performed in Section 5.2 and it is in agreement with previous studies [30,31,42,63]. When significant differences are not always found, in general the classifiers are more robust, offering a better behaviour.
b. When OVO decomposition is considered, mainly WV, LVPC, PC and PE methods stand out as the most robust aggregation strategies. However, we found different behaviours among the studied methods depending on the base classifier considered. Hence, the choice of the best aggregation is dependent on the base classifier.
   Moreover, VOTE that is the simplest combination (the weakest a priori), performs quite well with all base classifiers, but when kappa is considered, it shows its weaknesses. The analysis has also shown that it can be improved by using the appropriate confidence estimates.
c. Within OVA approach, DOO outperforms in all except one case (PDFC) the simpler MAX method. These results indicate that even though the aggregation strategies for OVA score vector have received much less attention in the literature than those for OVO score matrix, they can still be improved by changing the decision rule and not relying completely on the confidence given by the base classifiers.
d. The OVO unclassifiable region when VOTE aggregation is used is over-exploited. Many research efforts have been made in this area without significant differences (NEST, BTC, DDAG); therefore, the attention of new aggregations for OVO should be made in other directions.
e. Regardless of OVA strategy being seemingly weaker in comparison with OVO, the imbalanced data that is produced when instances from one class are confronted with all other examples could be the hitch.
   Furthermore, considering OVO vs OVA comparison, OVO methods in general have shown a better behaviour, especially according to the average performance obtained. It is worth noting that the difference between OVO and OVA approaches is more significant when kappa performance measure is considered, indicating the robustness of OVO strategies in contrast with OVA schemes. This issue is not totally against the findings made by Rifkin and Klautau in [63], since they are using fine-tuned SVM, and only in that case OVA approach was competitive (but without yielding to outperform OVO approach). Even though we could tune each base learner to adapt it to each data set, by using the same configuration for all the benchmark problems considered in the experimental study, we can observe the robustness of each method without depending on the refinement level of the base classifiers.
f. We have shown that confidence estimates different from total confidence yield better performance of the binarization strategies (a first study was developed in [31]), both in OVO and OVA approaches. Hence, an accurate confidence estimation produces useful ensembles, whereas a too much strict or an imprecise confidence does not allow the exploitation of all the underlying power within decomposition schemes.
g. Finally, considering the scalability of the methods, in the sense of the number of classes, OVO seems to have better performance

when the number of classes increases (see detailed results in the web-page associated to this paper). Considering the number of instances in a data set, theoretically it is also more suitable, basically because the learning of each classifier only involves examples from two classes, which produces simpler problems easier to learn (with a lower number of instances) while OVA system base classifiers have to deal with the whole data, which in this case highly increases the training times of the classifiers.

Throughout this paper, we have identified that binarization techniques with an appropriate combination strategy are simple but useful to improve classifiers performance, but still many future works remain to be addressed:

a. *Non-competent* examples (as stated in [46]): In OVO strategy, all classifiers are not trained with all the instances in the data set, but in testing phase, the new instance is submitted to all classifiers. The classifiers that have not been trained with the instance from the class of the new example will make a prediction that probably affects negatively the final results since these classifiers are not competent. Hence, detecting which classifiers are giving a response without really knowing anything about the example that has been introduced should improve the behaviour of OVO scheme.
b. *Techniques for imbalanced data sets*: Regarding OVA strategy, the training of each base classifier is usually affected by imbalanced data, which is a really hard problem in Machine Learning [15]. To undertake it, techniques from the community of imbalanced data sets should be applied to balance the instances of the class which is going to be discriminated, in order to improve the generalization ability of each base classifier.
c. *Scalability*: One of the challenges of Data Mining is the design of learners that extract information from large data sets. The theoretically better suitability of OVO scheme should be proved and also, its adaptation to large data sets in contrast with the original base classifiers adaptation remains to be studied. The scalability with respect to the number of classes should be also considered, since the learning of the decision boundaries and their combinations can be directly affected by this issue.
d. *OVO strategy as a decision making problem*: Many of the aggregations studied to combine the score matrix from OVO classifiers try to deal with the unclassifiable region when VOTE is used. Usually slightly improvements can be made within it, despite its importance; it has been widely studied, and new approaches should be more centred on other considerations such as LVPC and ND considering the problem as a decision making problem where the classifiers also could be inaccurate or erroneous in some cases.
e. *New combinations for OVA approach*: To the best of our knowledge this strategy has received less attention than OVO scheme in the literature. In general, the output from each classifier is used only to tie-break when more than one positive answers are obtained. Considering the score vector, more techniques can be developed by taking into account the uncertainty of the outputs and hence, combining all of them instead of only deciding with the positive ones, since probably more information is hidden among the outputs of the other classifiers.
f. *Data complexity measures*: The prediction capabilities of classifiers are strongly dependent on the problem's characteristics. These measures were proposed by Ho and Basu [12] and have been used in recent studies for extracting the domains of competence of an algorithm [53,14]. The data complexity analysis consists in measuring different aspects of the problem that are considered as complex to the classification task. Their application, particularly the application of the ones that are dependent on the overlapping of classes, could make if possible to characterize the behaviour of the ensemble techniques with respect to different base classifiers in particular problems. Hence, they would allow us to obtain a priori knowledge about the most suitable way to deal with each problem.

7. Concluding remarks

We made a thorough analysis of several ensemble methods applied on multi-classification problems in a general classification framework. All of them are based on two well-known strategies, OVO and OVA, whose suitability have been tested in several real-world problems data sets.

From this work we conclude that actually OVO methods and specifically the ensembles using WV, LVPC, PC and PE combinations are the ones with the best average behaviour, but the best aggregation within a problem depends on the base classifier that is considered. Besides, the best aggregation reasonably depends on the problem, but our aim has been to analyse which ones are the most robust strategies accounting for all the problems considered as a whole. In this manner, we have also studied which methods are more adaptable without having fine-tuned base classifiers or parameters for each problem.

SVM and PDFC work better when OVO decomposition is used, as well as C4.5 and Ripper where OVO ensembles outperforms significantly the original classifier. Using *k*NN as base classifier also improves the original classifier both in OVO and OVA strategies, but not with significant differences.

The results from the use of different base classifiers with different confidence estimates have shown that this point is a key factor. The best binarization techniques base their decision on these confidence estimates; therefore, to exploit all their capabilities the way in which the confidence is estimated should be chosen carefully.

Therefore, we have tried to answer to both questions that we have put forward. We have shown the suitability of binarization techniques with respect to the baseline classifiers and also the base classifier dependence of the aggregation strategy. We have obtained these conclusions by means of an exhaustive empirical analysis and finally we have discussed them in depth together with some future trends in the field of multi-classification with OVA and OVO decomposition strategies. We must conclude that in both of them there are many research lines to deal with yet.

Acknowledgements

This work has been supported by the Spanish Ministry of Education and Science under Projects TIN2010-15055 and TIN2008-06681-C06-01.

References

[1] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Machine Learning 6 (1991) 37–66.
[2] J. Alcalá-Fdez, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, F. Herrera, KEEL Data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, Journal of Multiple-Valued Logic and Soft Computing 17 (2–3) (2011) 255–287.
[3] J. Alcalá-Fdez, L. Sánchez, S. García, M.J. del Jesus, S. Ventura, J.M. Garrell, J. Otero, C. Romero, J. Bacardit, V.M. Rivas, J. Fernández, F. Herrera, KEEL: a software tool to assess evolutionary algorithms for data mining problems, Soft Computing 13 (3) (2009) 307–318.
[4] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

[5] E. Alpaydin, Introduction to Machine Learning, The MIT Press, 2004.
[6] A. Anand, P.N. Suganthan, Multiclass cancer classification by support vector machines with class-wise optimized genes and probability estimates, Journal of Theoretical Biology 259 (3) (2009) 533–540.
[7] R. Anand, K. Mehrotra, C.K. Mohan, S. Ranka, Efficient classification for multiclass problems using modular neural networks, IEEE Transactions on Neural Networks 6 (1) (1995) 117–124.
[8] O. Aran, L. Akarun, A multi-class classification strategy for fisher scores: application to signer independent sign language recognition, Pattern Recognition 43 (5) (2010) 1776–1788.
[9] A. Asuncion, D.J. Newman, UCI machine learning repository (2007), URL: 〈http://www.ics.uci.edu/∼mlearn/MLRepository.html〉.
[10] R.A. Baeza-Yates, B. Ribeiro-Neto, Modern Information Retrieval, Addison-Wesley Longman Publishing Co. Inc., Boston, MA, USA, 1999.
[11] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recognition 36 (3) (2003) 849–851.
[12] M. Basu, T.K. Ho, Data Complexity in Pattern Recognition, Springer, 2006.
[13] A. Ben-David, A lot of randomness is hiding in accuracy, Engineering Applications of Artificial Intelligence 20 (7) (2007) 875–885.
[14] E. Bernado-Mansilla, T.K. Ho, Domain of competence of XCS classifier system in complexity measurement space, IEEE Transactions on Evolutionary Computation 9 (1) (2005) 82–104.
[15] N.V. Chawla, N. Japkowicz, A. Kolcz (Eds.), Special Issue on Learning from Imbalanced Datasets, vol. 6, no. 11, 2004.
[16] Y. Chen, J.Z. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Transactions on Fuzzy Systems 11 (6) (2003) 716–728.
[17] P. Clark, R. Boswell, Rule induction with CN2: some recent improvements, in: EWSL'91: Proceedings of the European Working Session on Machine Learning, Springer-Verlag, London, UK, 1991, pp. 151–163.
[18] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46.
[19] W.W. Cohen, Fast effective rule induction, in: ICML'95: Proceedings of the 12th International Conference on Machine Learning, 1995, pp. 1–10.
[20] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.
[21] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.
[22] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, second ed., John Wiley, 2001.
[23] B. Fei, J. Liu, Binary tree of SVM: a new fast multiclass training and classification algorithm, IEEE Transactions on Neural Networks 17 (3) (2006) 696–704.
[24] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera, Enhancing fuzzy rule based systems in multi-classification using pairwise coupling with preference relations, in: EUROFUSE'09: Workshop on Preference Modelling and Decision Analysis, 2009, pp. 39–46.
[25] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera, Solving multi-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations, Fuzzy Sets and Systems 161 (23) (2010) 3064–3080.
[26] A. Fernández, S. García, J. Luengo, E. Bernadó-Mansilla, F. Herrera, Genetics-based machine learning for rule induction: state of the art, taxonomy and comparative study, IEEE Transactions on Evolutionary Computation 14 (6) (2010) 913–941.
[27] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recognition Letters 30 (1) (2009) 27–38.
[28] E. Frank, S. Kramer, Ensembles of nested dichotomies for multi-class problems, in: ICML '04: Proceedings of the 21st International Conference on Machine Learning, ACM, New York, NY, USA, 2004, pp. 305–312.
[29] J.H. Friedman, Another approach to polychotomous classification, Technical Report, Department of Statistics, Stanford University, 1996. URL: 〈http://www-stat.stanford.edu/∼jhf/ftp/poly.ps.Z〉.
[30] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (2002) 721–747.
[31] J. Fürnkranz, Round robin ensembles, Intelligent Data Analysis 7 (5) (2003) 385–403.
[32] J. Fürnkranz, E. Hüllermeier, S. Vanderlooy, Binary decomposition methods for multipartite ranking, in: W.L. Buntine, M. Grobelnik, D. Mladenic, J. Shawe-Taylor (Eds.), Machine Learning and Knowledge Discovery in Databases. Lecture Notes in Computer Science, vol. 5781(1), Springer, 2006, pp. 359–374.
[33] S. García, A. Fernández, J. Luengo, F. Herrera, A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability, Soft Computing 13 (10) (2009) 959–977.
[34] 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, Information Sciences 180 (2010) 2044–2064.
[35] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, Journal of Machine Learning Research 9 (2008) 2677–2694.
[36] N. Garcia-Pedrajas, C. Fyfe, Evolving output codes for multiclass problems, IEEE Transactions on Evolutionary Computation 12 (1) (2008) 93–106.
[37] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1001–1006.
[38] I. Guler, E.D. Ubeyli, Multiclass support vector machines for EEG-signals classification, IEEE Transactions on Information Technology in Biomedicine 11 (2) (2007) 117–126.
[39] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Annals of Statistics 26 (2) (1998) 451–471.
[40] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering 21 (9) (2009) 1263–1284.
[41] J.H. Hong, J.K. Min, U.K. Cho, S.B. Cho, Fingerprint classification using one-vs-all support vector machines dynamically ordered with Naïve Bayes classifiers, Pattern Recognition 41 (2) (2008) 662–671.
[42] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.
[43] J. Huang, C.X. Ling, Using AUC and accuracy in evaluating learning algorithms, IEEE Transactions on Knowledge and Data Engineering 17 (3) (2005) 299–310.
[44] J.C. Huhn, E. Hüllermeier, FR3: a fuzzy rule learner for inducing reliable classifiers, IEEE Transactions on Fuzzy Systems 17 (1) (2009) 138–149.
[45] E. Hüllermeier, K. Brinker, Learning valued preference structures for solving classification problems, Fuzzy Sets and Systems 159 (18) (2008) 2337–2352.
[46] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recognition 43 (1) (2010) 128–142.
[47] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in: F. Fogelman Soulié, J. Hérault (Eds.), Neurocomputing: Algorithms, Architectures and Applications. NATO ASI Series, vol. F68, Springer-Verlag, 1990, pp. 41–50.
[48] T.C.W. Landgrebe, R.P.W. Duin, Efficient multiclass ROC approximation by decomposition via confusion matrix perturbation analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (5) (2008) 810–822.
[49] B. Liu, Z. Hao, E.C.C. Tsang, Nesting one-against-one algorithm based on SVMs for pattern classification, IEEE Transactions on Neural Networks 19 (12) (2008) 2044–2052.
[50] B. Liu, Z. Hao, X. Yang, Nesting algorithm for multi-classification problems, Soft Computing 11 (4) (2007) 383–389.
[51] K.H. Liu, C.G. Xu, A genetic programming-based approach to the classification of multiclass microarray datasets, Bioinformatics 25 (3) (2009) 331–337.
[52] A.C. Lorena, A.C. Carvalho, J.M. Gama, A review on the combination of binary classifiers in multiclass problems, Artificial Intelligence Review 30 (1–4) (2008) 19–37.
[53] J. Luengo, F. Herrera, Domains of competence of fuzzy rule based classification systems with data complexity measures: a case of study using a fuzzy hybrid genetic based machine learning method, Fuzzy Sets and Systems 161 (1) (2010) 3–19.
[54] F. Masulli, G. Valentini, Effectiveness of error correcting output coding methods in ensemble and monolithic learning machines, Pattern Analysis and Applications 6 (4) (2003) 285–300.
[55] G.J. McLachlan, Discriminant Analysis and Statistical Pattern Recognition, John Wiley and Sons, 2004.
[56] S.A. Orlovsky, Decision-making with a fuzzy preference relation, Fuzzy Sets and Systems 1 (3) (1978) 155–167.
[57] A. Passerini, M. Pontil, P. Frasconi, New results on error correcting output codes of kernel machines, IEEE Transactions on Neural Networks 15 (1) (2004) 45–54.
[58] J.C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, MIT Press, Cambridge, MA, USA, 1999.
[59] 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.
[60] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognition 41 (2) (2008) 713–725.
[61] O. Pujol, P. Radeva, J. Vitria, Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1007–1012.
[62] J.R. Quinlan, C4.5: Programs for Machine Learning, first ed., Morgan Kaufmann Publishers, San Mateo, California, 1993.
[63] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (2004) 101–141.
[64] L. Rueda, B.J. Oommen, C. Henríquez, Multi-class pairwise linear dimensionality reduction using heteroscedastic schemes, Pattern Recognition 43 (7) (2010) 2456–2465.
[65] F. Schwenker, Hierarchical support vector machines for multi-class pattern recognition, in: KES'2000: Proceedings of the Fourth International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, vol. 2, 2000, pp. 561–565.
[66] J.P. Shaffer, Modified sequentially rejective multiple test procedures, Journal of the American Statistical Association 81 (395) (1986) 826–831.
[67] D. Sheskin, Handbook of Parametric and Nonparametric Statistical Procedures, second ed., Chapman & Hall/CRC, 2006.
[68] S.F. Smith, Flexible learning of problem solving heuristics through adaptive search, in: IJCAI'83: Proceedings of the Eighth International Joint Conference

on Artificial Intelligence, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1983, pp. 422–425.
[69] M. Sokolova, N. Japkowicz, S. Szpakowicz, Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation, in: A. Sattar, B.H. Kang (Eds.), Australian Conference on Artificial Intelligence. Lecture Notes in Computer Science, vol. 4304, Springer, 2006, pp. 1015–1021.
[70] Y. Sun, A.C. Wong, M.S. Kamel, Classification of imbalanced data: a review, International Journal of Pattern Recognition and Artificial Intelligence 23 (4) (2009) 687–719.
[71] L. Tao, Z. Chengliang, O. Mitsunori, A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression, Bioinformatics 20 (15) (2004) 2429–2437.
[72] A. Torralba, K.P. Murphy, W.T. Freeman, Sharing visual features for multiclass and multiview object detection, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (5) (2007) 854–869.
[73] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[74] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.
[75] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations, second ed., Morgan Kaufmann, San Francisco, 2005.
[76] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, Journal of Machine Learning Research 5 (2004) 975–1005.
[77] W. Youden, Index for rating diagnostic tests, Cancer 3 (1) (1950) 32–35.

Mikel Galar received his M.Sc. degree in Computer Sciences from the Public University of Navarra, Pamplona, Spain, in 2009. Currently he holds a research position at the Department of Automatics and Computation. His research interests are data-mining, classification, multi-classification, ensemble learning, evolutionary algorithms and fuzzy systems.

Alberto Fernández received his M.Sc. degree in Computer Sciences in 2005 and the Ph.D. degree in Computer Science in 2010, both from the University of Granada, Spain. He is currently a Supply Assistant Professor in the Department of Computer Science, University of Jaén, Jaén, Spain. His research interests include data mining, classification in imbalanced domains, fuzzy rule learning, evolutionary algorithms and multi-classification problems.

Edurne Barrenechea is an Assistant Lecturer at the Department of Automatics and Computation, Public University of Navarra. She received an M.Sc. in Computer Science at the Pais Vasco University in 1990. She worked in a private company (Bombas Itur) as analyst programmer from 1990 to 2001, and then she joined the Public University of Navarra as Associate Lecturer. She obtained the Ph.D. in Computer Science in 2005 on the topic interval-valued fuzzy sets applied to image processing. Her research interests are fuzzy techniques for image processing, fuzzy sets theory, interval type-2 fuzzy sets theory and applications, decision making, and industrial applications of soft computing techniques. She is member of the board of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Humberto Bustince is a Full Professor at the Department of Automatics and Computation, Public University of Navarra, Spain. He holds a Ph.D. degree in Mathematics from Public University of Navarra from 1994. His research interests are fuzzy logic theory, extensions of Fuzzy sets (Type-2 fuzzy sets, Interval-valued fuzzy sets, Atanassov's intuitionistic fuzzy sets), Fuzzy measures, Aggregation functions and fuzzy techniques for Image processing. He is author of over 50 published original articles and involved in teaching Artificial Intelligence for students of Computer Sciences.

Francisco Herrera received the M.Sc. degree in Mathematics in 1988 and the Ph.D. degree 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 published more than 150 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001). As edited activities, he has co-edited five international books and co-edited 20 special issues in international journals on different Soft Computing topics. He acts as associated editor of the journals IEEE Transactions on Fuzzy Systems, Information Sciences, Mathware and Soft Computing, Advances in Fuzzy Systems, Advances in Computational Sciences and Technology, and International Journal of Applied Metaheuristics Computing. He currently serves as area editor of the Journal Soft Computing (area of genetic algorithms and genetic fuzzy systems), 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 and Memetic Computation. His current research interests include computing with words and decision making, data mining, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.