Pattern Recognition 46 (2013) 3412–3424

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

Dynamic classifier selection for One-vs-One strategy: Avoiding non-competent classifiers

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 3 January 2012
Received in revised form
29 November 2012
Accepted 30 April 2013
Available online 9 May 2013

Keywords:
Multi-classification
Pairwise learning
One-vs-One
Decomposition strategies
Ensembles
Classifier selection

ABSTRACT

The One-vs-One strategy is one of the most commonly used decomposition technique to overcome multi-class classification problems; this way, multi-class problems are divided into easier-to-solve binary classification problems considering pairs of classes from the original problem, which are then learned by independent base classifiers.

The way of performing the division produces the so-called non-competence. This problem occurs whenever an instance is classified, since it is submitted to all the base classifiers although the outputs of some of them are not meaningful (they were not trained using the instances from the class of the instance to be classified). This issue may lead to erroneous classifications, because in spite of their incompetence, all classifiers' decisions are usually considered in the aggregation phase.

In this paper, we propose a dynamic classifier selection strategy for One-vs-One scheme that tries to avoid the non-competent classifiers when their output is probably not of interest. We consider the neighborhood of each instance to decide whether a classifier may be competent or not. In order to verify the validity of the proposed method, we will carry out a thorough experimental study considering different base classifiers and comparing our proposal with the best performer state-of-the-art aggregation within each base classifier from the five Machine Learning paradigms selected. The findings drawn from the empirical analysis are supported by the appropriate statistical analysis.

1. Introduction

Classification belongs to the broader category of Supervized Machine Learning [20], which attempts to extract knowledge from a set of previously seen examples $(x_1, \ldots, x_n)$ of a particular problem. Depending on its application domain, the samples are characterized by a different number (i) and type (numerical or nominal) of features $(\mathbb{A} = \{a_1, \ldots, a_i\})$, which define the input space of the learning task. The aim of the knowledge discovery is to construct a system capable of generalizing the concepts learned when new unseen examples from the same problem have to be analyzed. In case of classification, a system called a *classifier* is learned to distinguish between a set of classes $\mathbb{C} = \{c_1, \ldots, c_m\}$, considering a m class problem, which is the class of the new instance whose real class is unknown (in the learning phase, the class label of each instance is known). Hence, a classifier is as a mapping function defined over the patterns $\mathbb{A}^i \rightarrow \mathbb{C}$.

Although the concept of classifier is general for m-class problems, usually two types of classification tasks are referred in the literature depending on the number of classes considered. Binary classification problems include those only discerning between pair of classes; on the other hand, multi-class problems are those considering more than two classes, and hence, more general. Classification with multiple classes is usually more difficult, since the complexity of finding the decision boundaries increases. Even so, there is a large range of application domains in which multi-classification techniques are required, for instance, the classification of fingerprints [33], handwritten digits [47], microarrays [7] or face recognition [36].

In addition to the intrinsic difficulty of multiple classes learning, some of the most commonly used classifiers in Data Mining are intrinsically designed to deal with two classes, and their extensions to multiple classes are not established yet; this is the case of the well-known Support Vector Machines (SVMs) [55] or the positive definite fuzzy classifier [11] (which extracts fuzzy rules from the former). In these cases, the usual way to address

[*] Corresponding author. Tel.: +34 948 16 60 48; fax: +34 948 16 89 24.
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 © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.018

multi-class problems is by binarization techniques [44], which divide the original problem into more easier-to-solve two-class problems that are faced by binary classifiers; these classifiers are referred to as *base learners* or *base classifiers* of the ensemble [23]. On the contrary, other learners such as decision trees [50], instance-based classifiers [1] or decision lists [14] can directly manage multiple classes; however, it has been shown that the usage of decomposition techniques when dealing with several classes is usually preferable, since their base performance can be significantly enhanced [25].

Different decomposition strategies can be found in the specialized literature [44]. Among them, the most common are called “One-vs-One” (OVO) [37] and “One-vs-All” (OVA) [12], which can be included in the Error Correcting Output Code (ECOC) [17,4] framework. In this work, we focus our attention on OVO strategy, which divides the problem into as many binary problems as all the possible combinations between pair of classes; then, a classifier is learned to distinguish each pair. Finally, a new unseen instance is submitted to all the base classifiers whose outputs are then combined in order to predict the final class. This strategy is simple but powerful, being able to outperform the baseline classifiers not using binarization [25]. Moreover, it is used in very well-known software tools such as WEKA [31], LIBSVM [10] or KEEL [3] to model the multi-class problems when using SVMs.

Once the decomposition strategy is fixed, the combination of the outputs of the base classifiers must be studied. A thorough empirical analysis of the state-of-the-art on aggregations for OVO strategy has been developed [25]. Aggregations ranging from probability estimates [60] to preference relation-based methods [21], among others [32,22] were studied. Among the problems of OVO, the unclassifiable region when the voting strategy is used has attracted a lot of attention from researchers [42]; however, these approximations have not achieved the expected enhancement of the results. Anyway, in spite of the fact that generally no significant differences were found in their application, some of them presents a more robust behavior such as the weighted voting [35] or the methods based on probability estimates [60]. From [25], some future lines were stated; among them, the problem of non-competent classifiers (or examples) was appointed as an interesting research line to improve the performance of OVO strategy, which has not been directly undertaken yet. The non-competence is inherent from the way in which the multi-class problem is divided in OVO scheme; each classifier is only trained with the instances from the two classes that it must distinguish, whereas the instances belonging to other classes are not used. That is, they are unknown for the classifier, and so they are the outputs given by itself when instances from these classes are submitted in classification phase. Therefore, this problem appears at the classification stage when a new example is presented to all the binary classifiers, which must set a score for each one of the two classes for which they have been trained. Since all outputs are then aggregated, both the competent and non-competent classifiers are taken into account in the decision process, possibly misleading the correct labeling of the example.

Obviously, we cannot know a priori which classifiers we should use, because in that case, the classification problem would be solved. In this paper, our aim is to present a novel aggregation strategy based on Dynamic Classifier Selection (DCS) [30,41], which could reduce the number of non-competent classifiers in the classification phase; this way, erroneous classifications might be avoided. We will only take into account the classifiers that are more probably competent, that is, those classifiers that we are not sure whether they are competent or not (hence, that their class could be the output class). With this aim, we will analyze the neighbors of the instance to be classified, from which we will select the classifiers for the aggregation phase that will consider a reduced subset of classifiers. This approach can also be considered as a Dynamic Ensemble Selection (DES) technique [39,19], since more than one classifiers are selected to classify the instance. In the literature, both DCS and DES techniques are mainly devoted to ensembles in which all the base classifiers can distinguish all the classes (each one being specialized in different areas of the input space) [59,16]; nevertheless, their application in OVO decomposition has not been studied yet, probably because its application is more difficult and restricted, since the area of competence of each base classifier is established a priori in OVO and it does not depend on the input space but on the output space. Therefore, the application of this idea in decomposition strategy-based ensembles is the main contribution of this paper, unlike the DCS and DES works.

In order to evaluate the validity of our proposal, we develop a thorough empirical study maintaining the same experimental framework used in [25]. It includes a set of nineteen real-world problems from the KEEL data-set repository [3,2] (http://www.keel.es/dataset.php). We measure the performance of the classifiers based on its accuracy and we study the significance of the results by the proper statistical tests as suggested in the literature [15,28]. Finally, we test the proposed DCS strategy using several well-known classifiers from different Machine Learning paradigms: SVMs [55], decision trees [50], instance-based learning [1], fuzzy rule based systems [11] and decision lists [14].

The rest of this paper is organized as follows. In Section 2 we recall several concepts related to this work, binarization strategies, aggregations for OVO, and DCS techniques. Next, Section 3 shows our proposal to avoid non-competent classifiers in OVO. The experimental framework set-up is presented in Section 4, including the algorithms used as base classifiers and their parameters, the aggregations used for comparison, the data-sets, the performance measure and the statistical tests. We carry out the comparison of our proposal with the state-of-the-art methods in Section 5. Finally, Section 6 concludes the paper.

2. Related works: decomposition strategies and dynamic classifier selection

In this section we first recall the basics of binarization, and more specifically, we describe OVO strategy and some of their aggregations. Then, we present the ideas behind DCS in ensembles, and their differences with classifier combination.

2.1. Binarization for multi-classification

Decomposition strategies for addressing multi-class problems have been widely studied in the literature, an exhaustive review can be found in [44]. The same basic idea is behind all the decomposition proposals: to handle a multiple classes problem by the usage of binary classifiers. Following the divide and conquer paradigm, the more complex multi-class problem is divided into simpler binary classification problems. However, this division produces an added factor at the expenses of simplifying the base classifiers: their outputs must be combined in order to obtain the final class. Hence, the way in which they are aggregated is crucial to produce the desired results [25].

OVO [37] and OVA [12] decompositions are known to be the most common approaches. Whereas the former consists of learning a binary classifier to discern between each pair of classes, the latter constructs a binary classifier to separate each single class from all other classes. The simplest combination strategy is to consider the voting strategy, where each classifier gives a vote for a class and that with the largest number of votes is given as output (in OVA only one classifier should give a positive vote). In [4], Allwein et al. proposed a unifying framework for both approaches where they can be encoded within a code-matrix; in classification phase, a code-word is obtained from the outputs of the classifiers, which is then compared with the code-words in the code-matrix based on an Error Correcting Output Code (ECOC) [17] to decide the final class. Within ECOC framework, many proposals have been made, where the automatic design of the code-matrix is studied [49] and where different error correcting codes are used [61]. Anyway, OVO still continue being one of the most extended decomposition scheme, established by default in several widely used software tools [3,31,10].

The fact that an accepted extension of SVMs to multiple classes has not been established yet has produced an increasing application of binarization techniques, besides outperforming other multi-class SVM approaches [34]. In spite of those works focused on SVMs [34,52,29], many others have shown the suitability and usefulness of binarization techniques [37,23]. Moreover, these strategies provide a framework where both the training and testing phases can be easily parallelized.

2.2. One-vs-One decomposition scheme

In OVO or Pairwise classification, the original m-class problem is divided into $m(m-1)/2$ two-class problems, that is, all the possible pairs that can be formed from the set of classes. Afterwards, each sub-problem is faced by a binary classifier responsible of distinguishing between its pair of classes. Hence, the instances having a different class label are completely ignored by the classifier, which is the source of the problem that we are trying to undertake in this paper, the formerly named non-competence.

Once the base classifiers are learned, instances can be classified into m classes depending on the outputs given by the set of classifiers. In order to do so, it is usual to first construct a score-matrix R containing these outputs, which can be thereafter used to decide the final class:

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

where $r_{ij} \in [0,1]$ is the confidence of the classifier discriminating classes i and j in favor of the former; whereas, the confidence for the latter is computed by $r_{ji} = 1 - r_{ij}$ (if it is not provided by the classifier). Also, notice that the output class (i or j) of a classifier is obtained by the largest confidence (between r_{ij} and r_{ji}). Once the score-matrix is constructed, any of the aggregations presented in the following subsection can be used to infer the final class.

2.3. Combining binary classifiers' outputs in OVO decomposition

As we have mentioned, the predicted class of the system can be obtained from the score-matrix (Eq. (1)) using different combination models. In [25], a thorough review of the possible aggregations for the OVO strategy was carried out. The voting strategy is the simplest one, where each classifier votes for one of its two classes, the votes are summed up and the class with the largest number of votes is predicted. From this point, we only recall those aggregations that were selected as the best for each one of the base classifiers in [25], since we will compare our proposal with them.

- *Weighted voting strategy* (*WV*) [35]. The confidence of each base classifier in each class is used to vote for it. The class with the largest total confidence is the final output class:

$$Class = \arg \max_{i=1,\ldots,m} \sum_{1 \le j \ne i \le m} r_{ij} \tag{2}$$

- *Classification by pairwise coupling* (*PC*) [32]. This method aims to estimate the posterior probabilities of all the classes starting from the pairwise class probabilities. Therefore, being $r_{ij} = \text{Prob}(Class_i | Class_i \text{ or } Class_j)$, the method finds the best approximation of the class posterior probabilities $\hat{\mathbf{p}} = (\hat{p}_1, \ldots, \hat{p}_m)$ according to the pairwise outputs. Finally, the class having the largest posterior probability is predicted:

$$Class = arg \max_{i=1,\ldots,m} \hat{p}_i \tag{3}$$

The posterior probabilities are computed by minimizing the Kullback–Leibler (KL) distance between r_{ij} and μ_{ij}:

$$l(\mathbf{p}) = \sum_{1 \le j \ne i \le m} n_{ij} r_{ij} \log \frac{r_{ij}}{\mu_{ij}} = \sum_{i<j} n_{ij} \left(r_{ij} \log \frac{r_{ij}}{\mu_{ij}} + (1 - r_{ij}) \log \frac{1 - r_{ij}}{1 - \mu_{ij}} \right) \tag{4}$$

where $\mu_{ij} = p_i/(p_i + p_j)$, $r_{ji} = 1 - r_{ij}$ and n_{ij} is the number of training data from the ith and jth classes.

- *Preference relations solved by non-dominance criterion* (*ND*) [21]. In the non-dominance criterion the score-matrix is considered as a fuzzy preference relation, which must be normalized. This method predicts the class with the largest degree of non-dominance, that is, the class which is less dominated by all the remaining classes (or is much dominated by no one of the remaining classes):

$$Class = \arg \max_{i=1,\ldots,m} \left\{ 1 - \sup_{j \in \mathbb{C}} r'_{ji} \right\} \tag{5}$$

where r'_{ji} corresponds to the normalized and strict score-matrix.

- *Wu, Lin and Weng probability estimates by pairwise coupling approach* (*PE*) [60]. PE is similar to PC, it also estimates the posterior probabilities ($\mathbf{p}$) of each class from the pairwise probabilities. However, in this case, the optimization formulation is different, in spite of using the same decision rule. PE optimizes the following equation:

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \le j \ne i \le m} (r_{ji} p_i - r_{ij} p_j)^2$$
$$\text{subject to} \quad \sum_{i=1}^{m} p_i = 1, p_i \ge 0 \quad \text{for all } i \in \{1, \ldots, m\}. \tag{6}$$

A more extensive and detailed description of these methods, with the original source papers' descriptions is available in [24]. Notice that all these methods use exactly the same score-matrix values to compute the final class, but they can obtain different results.

Regarding the different combinations developed in the literature to combine the base classifiers in OVO strategy, it is interesting to note that the unclassifiable region in OVO when the voting strategy is used has attracted a lot of attention [42]. However, these approaches generally do not outstand with respect to the others [25]. Moreover, neither these approaches nor the others have tried to deal with the non-competent classifiers. As previously mentioned, non-competent classifiers are those giving an answer for an instance whose class has not been considered by themselves in training phase. Technically, these classifiers do not suppose any problem whenever the base classifiers are correct, since in spite of the aggregation considered, the solution should be correct; nevertheless, this assumption is not always fulfilled, which directly leads to the development of different aggregation or combination strategies.

2.4. Dynamic classifier selection

At this point, we should note that despite the decomposition strategies are obviously ensembles, this term more commonly refers to sets of classifiers which are able to output any of the classes of the problem; for this reason, their aggregation, combination or selection is treated in a different way [53,26]. Classifier selection literature [30,41,8] is mainly devoted to this type of ensembles, with only some exceptions [33]. Hence, it is important to note that they are ensembles with different objectives, whereas decomposition techniques aims to solve a multi-class problem by binary classifiers, the classic ensembles try to improve the performance of single classifiers by inducing several classifiers and combining them to obtain a new classifier that outperforms every one of them.

Classifier selection methods perform the classifier combination of an ensemble in a different way in which the classifier aggregation does. A classifier ensemble aims to solve a classification problem by combining several (diverse) classifiers; these classifiers can be trained in such a way that the aggregation of their outputs leads to a better classification performance than their individual accuracy. However, instead of combining all the classifiers of the ensemble, classifier selection tries to select only those (or that) leading to the best classification performance. Therefore, in classifier aggregation, all the classifiers are supposed to be equally accurate in the whole feature space, but they misclassify different instances. On the contrary, classifier selection assumes that the classifiers are complementary, being experts on classifying the instances from a part of the input space; hence, when a new instance is submitted, the most competent classifiers (one or more) are selected to perform the classification.

In the literature, two different categories of classifier selection can be distinguished [40]:

- *Static*: A region of competence is defined in training time for each classifier. Then, before classifying a new example, its region is found and the corresponding classifier is used to decide its class [5,56,43].
- *Dynamic*: In this case, the classifier that classify the new instance is decided during the classification. First, the competence (accuracy) of each classifier for classifying the instance is estimated, and only the winner's output will be used to label the instance [59,38,18,58,6].

Anyway, both types can be seen as a unique DCS paradigm where, in the first case, the classification phase is accelerated by precomputing the competence of the classifiers. Moreover, DES can be shown as a generalization of DCS when a subset of classifiers is selected instead of a unique classifier [39,58,9]. These selection procedures mainly focus on defining the competence of the classifiers by its accuracy [59,54] or by the region of the input space in which they are the experts [5,40]. However, in the ensembles we are dealing with (OVO strategy) neither the individual accuracy can be used (the classifiers are more or less accurate in their pair of classes), nor regions in the input space can be defined (all classifiers are supposed to be equally accurate in the whole feature space). OVO scheme implicitly establishes the region where each classifier is competent, but the problem is that this region is defined in the output space; hence, knowing which classifier is competent, is the same as predicting the class of the instance. Therefore, we have to take into account that we can only try to avoid some non-competent classifiers, which can hinder the classification of some instances.

3. Dynamic classifier selection for One-vs-One strategy

In this section, we present our proposal to avoid non-competent classifiers in OVO strategy and we discuss the computational complexity of the proposed method. Then, we show how it works by an illustrative example.

3.1. Dynamic OVO

As we have previously stated, our aim is to avoid those non-competent classifiers which can hinder the decision of the ensemble when some of the classifiers distinguishing the correct class have failed (and even when not, see the example in Section 3.3). To do so, we consider the use of DCS algorithms; however, they do not suit our problem as they do with classic ensembles, since we cannot establish regions for each classifier in the input space or estimate their accuracies among the whole set of classes. Hence, we need to adapt these techniques to the OVO framework, since our base classifiers are only competent for the two classes used in their training phase. Obviously, we cannot restrict our search of the competent classifiers only to find the classifiers that have considered the class of the instance, which of course is a priori unknown; but, we can try to seek for a small subset of classes to whom membership is more probable. In such a way, we can consider a score-matrix where only the classifiers trained for that classes are taken into account; therefore, we could reduce the number of classifiers in the classification removing those classifiers from which we are sure enough that they do not contribute to the correct decision, or even they could harm it.

In the same way as in other DCS methods [59], we consider to use the neighbors of the instance to be classified in order to decide whether a classifier may be competent or not. However, in spite of using the neighbors to estimate the local accuracy, we will use them to select the classes that will take part in the reduced score-matrix. The dynamic procedure is as follows:

(1) We compute the k nearest neighbors of the instance to be classified (k is a parameter of the algorithm, which actually is established to 3 times the number of classes).
(2) We select those classes in the neighborhood (in case that within the k nearest neighbors there is a unique class, we increase the number of neighbors until a maximum of $2 \cdot k$, in which case we follow with the standard OVO so we do not use kNN to label the instance).
(3) We will only use the classifiers that were trained with a pair of classes from the selected subset to form the new score-matrix and continue with one of the aggregations for OVO (Section 2.3).

The simplicity of this method is an important advantage, it only uses kNN dynamically to make a pre-selection of classes among a large number of neighbors. For this reason, it is difficult to misclassify an instance due to the elimination of the correct class from the score-matrix. Nevertheless, it could occur that an instance is misclassified because its class has been deleted in the dynamic procedure, but in such a case, the original OVO would probably not predict it correctly (it may be an outlier or a rare example), since none of its neighbors is of the same class (notice that our neighborhood size is larger than the usual size used for classification purposes [27]).

We propose to use $k = 3 \cdot m$ in order to consider a large enough neighborhood, where there would much probably be instances belonging to more than one class. However, at the same time, this value should not be too large, since some of the classes should fall out of the neighborhood (we aim to show that removing non-competent classifiers helps the final decision). The selected value, intuitively, provides a good trade-off to achieve both objectives.

Moreover, if the initial value is low, we get closer to *k*NN classifier, which is not desired, whereas if greater values are used, we get farther from the dynamic strategy. The case of $2 \cdot k$ as a limit for the search procedure, which is hardly ever reached, is established aiming to not extend this search excessively in such rare cases.

After making the class pre-selection dynamically, it is interesting to note that any of the existing OVO aggregations [25] can be used to decide over the new score-matrix. In our case, we propose to use the WV strategy since its robustness has been both theoretically [35] and experimentally (showing a robust behavior with different base classifiers) [25] proved.

3.2. On the computational complexity of Dynamic OVO

We acknowledge that the proposed method might be more computationally expensive than standard aggregation techniques, but this would also highly depend on the way of its implementation. After carrying out the DCS, any aggregation must be used at the final phase, but it is performed over the reduced score-matrix.

In our case, we need to find the neighbors of a given instance, that is, to apply *k*NN classifier, whose complexity is of $\mathcal{O}(n \cdot i)$, where n is the number of examples in the training set and i is the number of attributes. In case of dealing with large data-sets, an instance selection procedure [27] could be carried out in order to reduce the reference set and hence, to reduce the testing time.

In any case, this additional cost can be nearly avoided if Dynamic OVO is implemented in parallel with the classifiers classification phase. Moreover, if it is not carried out in parallel, it can be executed before the classifiers, hence reducing the time needed to classify an instance by all the classifiers (since the size of the ensemble is reduced).

3.3. An illustrative example

In order to show the how the proposal works, hereafter we show an illustrative example were it would solve the problem of non-competence.

Suppose that we have to classify an instance x, whose real class is known to be c_1 and whose corresponding OVO score-matrix obtained by submitting it to the binary classifiers is the following:

$$R(x) = \left(\begin{array}{cccccc} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} \\ \mathbf{c_1} & - & 0.55 & 0.6 & 0.75 & 0.7 \\ \mathbf{c_2} & 0.45 & - & 0.4 & 1 & 0.8 \\ \mathbf{c_3} & 0.4 & 0.6 & - & 0.5 & 0.4 \\ \mathbf{c_4} & 0.25 & 0.0 & 0.5 & - & 0.1 \\ \mathbf{c_5} & 0.3 & 0.2 & 0.6 & 0.9 & - \end{array} \right) \tag{7}$$

Starting from $R(x)$, if we consider the usage of the WV strategy, we would obtain class c_2 as output class (Expression (8)). However, none of the classifiers considering c_1 have failed; but the rest of the classifiers distinguishing c_2, in spite of its non-competence (which is the source of the failure), have strongly voted for it, and for this reason it is predicted.

$$R(x) = \left(\begin{array}{cccccc|c} & \mathbf{c_1} & \mathbf{c_2} & \mathbf{c_3} & \mathbf{c_4} & \mathbf{c_5} & \text{WV} \\ \mathbf{c_1} & - & 0.55 & 0.6 & 0.75 & 0.7 & 2.6 \\ \mathbf{c_2} & 0.45 & - & 0.4 & 1 & 0.8 & \mathbf{2.65} \\ \mathbf{c_3} & 0.4 & 0.6 & - & 0.5 & 0.4 & 1.9 \\ \mathbf{c_4} & 0.25 & 0.0 & 0.5 & - & 0.1 & 0.85 \\ \mathbf{c_5} & 0.3 & 0.2 & 0.6 & 0.9 & - & 2.1 \end{array} \right) \tag{8}$$

Now, having the same score-matrix, if we apply our algorithm, we should first compute the k nearest neighbors of x (where $k = 3 \cdot 5$ classes $= 15$ neighbors). Suppose that the subset of classes in this neighborhood is $\{c_1, c_4, c_5\}$. Then, we remove from the score-matrix those classifiers which do not consider pairs of classes from this subset (Expression (9)), that is any classifier trained with classes $\{c_2, c_3\}$. Finally, we apply the WV (any other aggregation could be used) procedure to the new score-matrix. This time, c_1 is predicted, which is actually the real class of the instance.

$$R_{dyn}(x) = \left(\begin{array}{cccccc|c} & \mathbf{c_1} & \cancel{\mathbf{c_2}} & \cancel{\mathbf{c_3}} & \mathbf{c_4} & \mathbf{c_5} & \text{WV} \\ \mathbf{c_1} & - & \cancel{0.55} & \cancel{0.6} & 0.75 & 0.7 & \mathbf{1.45} \\ \cancel{\mathbf{c_2}} & \cancel{0.45} & - & \cancel{0.4} & \cancel{1} & \cancel{0.8} & - \\ \cancel{\mathbf{c_3}} & \cancel{0.4} & \cancel{0.6} & - & \cancel{0.5} & \cancel{0.4} & - \\ \mathbf{c_4} & 0.25 & \cancel{0.0} & \cancel{0.5} & - & 0.1 & 0.35 \\ \mathbf{c_5} & 0.3 & \cancel{0.2} & \cancel{0.6} & 0.9 & - & 1.2 \end{array} \right) \tag{9}$$

4. Experimental framework

In this section, we present the experimental framework's set-up used to develop the empirical comparison in Section 5. First, we describe the base classifiers that we have considered and we show the parameter configuration used in Section 4.1. Then, in Section 4.2 we recall which were the best aggregations for each base classifier [25] that will be the base for the comparisons. Afterwards, we provide details of the real-world problems chosen for the experimentation in Section 4.3. Finally, in Section 4.4 we present the performance measure used and the statistical tests applied to make the comparison of the results between the different aggregations.

4.1. Base learners and parameter configuration

Our aim in the empirical study is to compare our DCS proposal with the state-of-the-art aggregations for OVO strategy. For this purpose, we have selected several well-known Machine Learning algorithms as base learners, so we are able to compare our approach within different paradigms. We should mention that the whole experimental set-up is the same as the one in [25], where the state-of-the-art on aggregations for the OVO strategy were compared. We want to maintain the same framework in order to show a fair comparison.

Therefore, the algorithms used in the comparison are the following ones:

- *SVM* (*support vector machine*) [55] maps the original input space into a high-dimensional feature space; it uses a certain kernel function to avoid the computation of the inner product between vectors. Once the instances are in the new feature space, the optimal separating hyperplane is found, that is, the one with maximal margin such that the upper bound of the expected risk is minimized. We use SMO [48] algorithm to train the SVM base classifiers.
- *C4.5* [50] decision tree induces classification rules in the form of decision trees. The decision tree is constructed by using the given examples in a top-down manner, where the normalized information gain (difference in entropy) is used to select the attribute that better splits the data.
- *kNN* (*k-nearest neighbors*) [1] is an instance-based classifier that finds the k instances in the training set that are the closest to the test pattern. Then, the instance is labeled based on the predominance of a particular class in this neighborhood. Both the distance and the number of neighbors are key elements of this algorithm.
- *Ripper* (*repeated incremental pruning to produce error reduction*) [14] builds a decision list of rules to predict the corresponding class for each instance. Each list of rules is grown one by one and immediately pruned. When a decision list for a given class

is completed, it goes through an optimization phase in the next stage.
- *PDFC* (*positive definite fuzzy classifier*) [11] extracts fuzzy rules from a 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.

These learning algorithms were selected for the current (and the previous) study due to their good behavior in a large number of real problems. Moreover, in case of SVM and PFDC there is not a multi-category approach established yet, despite there exist several extensions [34], they do not present real advantages to decomposition strategies that are used in the SVM community for multi-classification. Most of the aggregation methods for OVO classification make their predictions based on a given confidence of the base classifiers' outputs. In this paper, 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 [48] as the confidence for the predicted class.
- *C4.5*: The confidence is obtained from the accuracy of the leaf making the prediction. The accuracy of a leaf is the percentage of correctly classified train examples divided by the total number of covered train instances.
- *kNN*: We use the following equation to estimate the confidence of *k*NN:

$$Confidence = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}} \tag{10}$$

  where d_l is the distance between the input pattern and the lth neighbor and $e_l = 1$ if the neighbor l is from the class and 0 otherwise. Note that when $k > 1$, the probability estimate depends on the distance from the neighbors, hence the estimation is not restricted to a few values (when only the numbers of neighbors from each class are considered, a multi-valued result is obtained, which is not a desired characteristic).
- *Ripper*: Similarly to C4.5, the confidence is taken from the accuracy of the rule used in the prediction, that is, the percentage of correctly classified train instances divided by the total number of train instances that it covers.
- *PDFC*: The confidence only depends on the prediction of the classifier, confidence equal to 1 is given for the predicted class.

In some of the aggregation strategies ties might occur, as usual, in those cases, the majority class is predicted, if the tie continues, the class is selected randomly.

The configuration parameters used to train the base classifiers are shown in Table 1. The selected values are common for all problems, and they were selected according to the recommendation of the corresponding authors of each algorithm, which is also the default parameters' setting included in KEEL[1] software [3,2] that we used to develop our experiments. We considered two configurations for SVMs, where the parameter C and the kernel function are changed, so we can study the behavior of our strategy with different configurations, which should address for the robustness of the proposal (in the sense that despite how fine-tuned are the base classifiers, its behavior is maintained with respect to the others). 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.

Finally, with respect to the DCS, we use the Euclidean distance to find the neighbors of the instance (except when the data-set contains nominal values where we use the Heterogeneous Value Difference Metric, HVDM).

Table 1
Parameter specification for the base learners employed in the experimentation.

Algorithm	*Parameters*
SVM$_{Poly}$	C=1.0 Tolerance parameter=0.001 Epsilon=1.0E−12 Kernel type=polynomial Polynomial degree=1 Fit logistic models=true
SVM$_{Puk}$	C=100.0 Tolerance parameter=0.001 Epsilon=1.0E−12 Kernel type=Puk PukKernel $\omega = 1.0$ PukKernel $\sigma = 1.0$ Fit logistic models=true
C4.5	Prune=true Confidence level=0.25 Minimum number of item-sets per leaf=2
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

4.2. Aggregations considered

In this paper, we will consider as representative aggregations (presented in Section 2.3) for each base classifier the same as those selected in [25] (recall that the same experimental framework is being used). There is only an exception regarding SVMs. In [25], the best performer aggregation was Nesting OVO [42], but without significant differences with the rest of the aggregations. Moreover, even though this strategy constructs several OVO ensembles recursively, it does not improve other simpler approaches such as the probability estimates method by Wu et al. [60], which is more extended. For this reason, being the second method equivalent yet simpler, we consider it to be the representative. Furthermore, in this manner, we are able to perform the comparison using exactly the same score-matrices in all aggregations, so the unique differences between the results on this analysis are due to the aggregations themselves. Therefore, the aggregations for each classifier are the following:

- *SVM*: PE (Wu et al. probability estimates by pairwise coupling).
- *C4.5*: WV (weighted voting strategy).
- *kNN*: ND (non-dominance criterion).

[1] http://www.keel.es.

- *Ripper*: WV (weighted voting strategy).
- *PDFC*: PC (probability estimates by pairwise coupling).

Regarding Dynamic OVO, as we have previously stated, we use the WV as aggregation for the last phase due to its robustness; however, we will also show the results using the best performer aggregation in each base classifier.

4.3. Data-sets

In the study, we have used the same nineteen data-sets from KEEL data-set repository[2] [2] that were considered in [25]. Table 2 summarizes their properties. 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.) are shown. Some of the largest data-sets (nursery, page-blocks, penbased, satimage and shuttle) 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. Notice that the stratified data-sets were used in order to maintain the same experimental framework as that in [25], so that any interested reader could compare the results presented in this paper with those in the associated web-page http://sci2s.ugr.es/ovo-ova. In addition, we have considered the non-reduced versions of the stratified data-sets in Section 5.3, in order to test the proposed methodology in large data-sets. The information of the data-sets is completed with the number of instances per class in each data-set (Table 3). As it can be observed, they comprise a number of situations, from totally balanced data-sets to highly imbalanced ones, besides the different number of classes.

The selection of these data-sets was carried out according to the premise of having more than 3 classes and a good behavior 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 verify the validity of our proposal and to study its robustness, in such a way that the extracted conclusions are valid for general multi-classification problems. In this manner, we will be able 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 more meaningful results are obtained from a multi-classification point-of-view.

The performance estimates were obtained by means of a 5-fold stratified cross-validation (SCV), 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 to our previous work [25] (http://sci2s.ugr.es/ovo-ova/). From our point view, 5-fold SCV is more appropriate than a 10-fold SCV in the current framework, since using smaller partitions, there would be more partitions that will not contain any instance from some of the classes (there are data-sets with few instances belonging to some classes). In addition to the commonly used SCV, we will also test our algorithm with a recently published partitioning procedure: the Distribution Optimally Balanced SCV (DOB-SCV) [46]. This procedure aims to correct the data-set shift (when the training data and the test data do not follow the same distribution [51,45]) that might be produced when dividing the data [46].

Table 2
Summary description of data-sets.

Data-set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Car	1728	6	0	6	4
Lymphography	148	18	3	15	4
Vehicle	846	18	18	0	4
Cleveland	297	13	13	0	5
Nursery	1296	8	0	8	5
Page-blocks	548	10	10	0	5
Shuttle	2175	9	9	0	5
Autos	159	25	15	10	6
Dermatology	358	34	1	33	6
Flare	1066	11	0	11	6
Glass	214	9	9	0	7
Satimage	643	36	36	0	7
Segment	2310	19	19	0	7
Zoo	101	16	0	16	7
Ecoli	336	7	7	0	8
Led7digit	500	7	0	7	10
Penbased	1100	16	16	0	10
Yeast	1484	8	8	0	10
Vowel	990	13	13	0	11

4.4. Performance measure and statistical tests

As a performance measure, we have considered the classification rate, also called accuracy rate. It is computed as 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 to assess the performance of classifiers for years. For the sake of brevity, we have not included the kappa metric [13], since in this framework it provides equivalent results to the more extended accuracy rate.

In order to properly compare classifiers' performances, statistical analysis needs to be carried out to find whether significant differences among the results obtained exist or not. We consider the use of non-parametric tests, according to the recommendations made in [15,28] where a set of simple, safe and robust non-parametric tests for statistical comparisons of classifiers is presented. These tests are needed because the initial conditions that guarantee the reliability of the parametric tests may not be satisfied, causing the statistical analysis to lose its credibility [15]. Any interested reader can find additional information on the thematic website http://sci2s.ugr.es/sicidm/, together with the software for applying the statistical tests.

Finally, we will compare our approach with the best aggregation in each base classifier; hence, we need to perform pairwise comparisons. For this reason, we use the Wilcoxon paired signed-rank test [57] as a non-parametric statistical procedure to perform comparisons between two algorithms.

5. Experimental study

In this section, we will study the usefulness of our proposal. To do so, we compare our dynamic approach with the best performers aggregations from the state-of-the-art. We have divided this section into four parts. First, we will show the comparison of the dynamic approach using the WV as aggregation function (Section 5.1). Then, we will perform a similar analysis using for each base classifier the aggregation that was the best performer also in the dynamic approach (Section 5.2). In such a manner, we aim to verify the goodness of our approach and to stress the positive synergy existing between Dynamic OVO and the WV. Thereafter, we will analyze the behavior of the Dynamic approach with large data-sets (Section 5.3) and we will end the experimental study considering DOB-SVC data-partitioning

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

Table 3
Number of instances per class in each data-set.

Data-set	**#Ex.**	**#Cl.**	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}
Car	1728	4	1210	384	65	69							
Lymphography	148	4	2	81	61	4							
Vehicle	846	4	199	217	218	212							
Cleveland	297	5	160	54	35	35	13						
Nursery	1296	5	1	32	405	426	432						
Pageblocks	548	5	492	33	8	12	3						
Shuttle	2175	5	1706	2	6	338	123						
Autos	159	6	3	20	48	46	29	13					
Dermatology	358	6	111	60	71	48	48	20					
Flare	1066	6	331	239	211	147	95	43					
Glass	214	7	70	76	17	0	13	9	29				
Satimage	643	7	154	70	136	62	71	0	150				
Segment	2310	7	330	330	330	330	330	330	330				
Zoo	101	7	41	20	5	13	4	8	10				
Ecoli	336	8	143	77	2	2	35	20	5	52			
Led7digit	500	10	45	37	51	57	52	52	47	57	53	49	
Penbased	1100	10	115	114	114	106	114	106	105	115	105	106	
Yeast	1484	10	244	429	463	44	51	163	35	30	20	5	
Vowel	990	11	90	90	90	90	90	90	90	90	90	90	90

Table 4
Average accuracy results in test of the representative aggregations and the dynamic strategy (with WV) for each base classifier.

Data-set	3NN		C4.5		Ripper		SVM_{Poly}		SVM_{Puk}		PDFC	
	ND	Dyn^{WV}	WV	Dyn^{WV}	WV	Dyn^{WV}	PE	Dyn^{WV}	PE	Dyn^{WV}	PC	Dyn^{WV}
Autos	76.75	**77.38**	**81.17**	79.92	78.65	**79.27**	74.80	**75.42**	**68.53**	67.88	76.71	**78.61**
Car	92.82	**93.06**	93.00	**93.29**	91.32	**92.36**	**92.71**	92.65	63.60	**84.72**	99.88	**99.94**
Cleveland	56.58	**56.58**	51.53	**52.53**	48.45	**49.81**	**58.25**	57.59	45.09	**45.42**	52.83	**53.85**
Dermatology	89.96	**93.58**	96.37	**98.31**	93.02	**93.30**	94.13	**94.69**	**96.09**	**96.09**	84.36	**92.18**
Ecoli	**80.96**	80.66	79.47	**80.07**	**75.89**	75.60	77.69	**77.99**	75.31	**75.90**	**82.75**	**82.75**
Flare	**72.14**	71.95	74.20	**74.29**	**72.61**	72.61	74.67	**75.33**	69.42	**72.89**	**72.98**	**72.98**
Glass	**73.38**	**73.38**	70.53	**71.47**	**74.30**	**74.30**	61.26	**62.66**	70.60	**71.09**	68.25	**70.11**
Led7digit	72.60	**72.80**	72.20	**72.60**	70.00	**71.60**	73.00	**74.00**	70.20	**71.40**	71.80	**72.60**
Lymphography	**83.72**	**83.72**	73.63	**76.30**	76.28	**77.63**	**81.68**	**81.68**	**80.34**	**80.34**	**78.94**	**78.94**
Nursery	**92.52**	**92.52**	**89.04**	88.89	**89.43**	89.28	**91.90**	**91.90**	81.33	**89.12**	**96.84**	**96.84**
Pageblocks	94.70	**94.88**	95.61	**95.79**	**95.25**	95.25	**94.70**	94.52	94.16	**94.34**	**95.43**	**95.43**
Penbased	96.18	**96.27**	90.64	**90.73**	89.91	**90.82**	95.27	**95.64**	97.82	**97.91**	**98.27**	**98.27**
Satimage	**86.47**	86.32	81.65	**82.74**	79.78	**80.40**	**84.14**	83.67	84.92	**85.08**	**87.41**	87.10
Segment	96.71	**96.97**	**97.06**	96.97	95.84	**96.41**	92.55	**92.60**	**97.10**	97.06	**96.62**	96.54
Shuttle	99.54	**99.63**	**99.72**	99.68	99.49	**99.68**	96.37	**97.52**	**99.72**	99.68	97.47	**98.21**
Vehicle	**71.40**	**71.40**	71.39	**72.81**	71.04	**71.75**	72.46	**73.29**	80.49	**80.61**	83.57	**83.69**
Vowel	**95.86**	**95.86**	80.00	**81.21**	**78.99**	78.48	69.90	**70.30**	**99.39**	**99.39**	**97.98**	**97.98**
Yeast	**54.72**	54.52	**59.91**	59.57	56.00	**56.07**	**59.10**	58.96	**56.54**	56.27	**59.10**	59.03
Zoo	92.10	**94.10**	**93.10**	**93.10**	94.10	**95.10**	95.05	**96.05**	84.19	**85.19**	**97.05**	**97.05**
Mean	83.11	**83.45**	81.59	**82.12**	80.54	**81.04**	81.03	**81.39**	79.73	**81.60**	84.12	**84.85**

technique in order to show the robustness of the dynamic approach despite of the data-partitioning method used (Section 5.4).

5.1. Dynamic OVO vs. state-of-the-art aggregations

Hereafter, we compare our DCS technique to avoid non-competent examples with the previously mentioned aggregations. We want to investigate whether the avoidance of the non-competent classifiers is translated into an enhancement of the results, or otherwise, if trying to avoid these classifiers we are also removing relevant classes from the decision process that cannot be predicted, even though they are the classes that should be predicted.

Table 4 shows the test accuracy results of the different methods. For each base classifier, we show the results corresponding to its best performer aggregation and the results of the Dynamic OVO procedure, which in this case uses the WV strategy in order to make the final decision (Dyn refers to the Dynamic OVO where its superscript denotes the aggregation used). The best result within each base classifier and data-set is stressed in bold-face.

From Table 4 we can observe that our proposal works well in all the base classifiers studied. The mean accuracy among all data-sets is always better, and the number of data-sets in which Dynamic OVO outstands is remarkable. However, we cannot obtain any meaningful conclusions without basing our statements on the proper statistical tests. Hence, we carry out a pairwise comparison for each base classifier using the Wilcoxon signed-rank test, which results are shown in Table 5.

The results of the statistical tests are clear. Dynamic OVO is significantly better than the representative of the corresponding base classifier in five out of six cases (in all except 3NN), with very low p-values. In case of 3NN, although the test is not rejected, the ranks are in favor of the dynamic strategy and its p-value is low. Therefore, these results put out the superiority of the dynamic procedure with respect to the classic approaches. The avoidance of the non-competent classifiers have led us to obtain significantly

Table 5
Wilcoxon tests to compare the representative aggregations and the dynamic OVO with different base classifiers. R^+ corresponds to the sum of the ranks for Dynamic OVO and R^- for the representative aggregation.

Classifier	Comparison	R^+	R^-	Hypothesis	p-Value
3NN	Dyn^{WV} vs. ND	133.5	56.5	Not rejected	0.140146
C4.5	Dyn^{WV} vs. WV	153.5	36.5	**Rejected for Dyn^{WV} at 5%**	0.017621
Ripper	Dyn^{WV} vs. WV	162.5	27.5	**Rejected for Dyn^{WV} at 5%**	0.005684
SVM_{Poly}	Dyn^{WV} vs. PE	149.5	40.5	**Rejected for Dyn^{WV} at 5%**	0.024520
SVM_{Puk}	Dyn^{WV} vs. PE	154.0	36.0	**Rejected for Dyn^{WV} at 5%**	0.015086
PDFC	Dyn^{WV} vs. PC	138.0	52.0	**Rejected for Dyn^{WV} at 5%**	0.040860

Table 6
Average accuracy results in test of the representative aggregations and the dynamic strategy (with the a priori best aggregation) for each base classifier.

Data-set	3NN		C4.5		Ripper		SVM_{Poly}		SVM_{Puk}		PDFC	
	ND	Dyn^{ND}	WV	Dyn^{WV}	WV	Dyn^{WV}	PE	Dyn^{PE}	PE	Dyn^{PE}	PC	Dyn^{PC}
Autos	76.75	**80.50**	**81.17**	79.92	78.65	**79.27**	**74.80**	**74.80**	**68.53**	67.88	76.71	**77.34**
Car	92.82	**93.23**	93.00	**93.29**	91.32	**92.36**	**92.71**	90.86	63.60	**84.72**	**99.88**	**99.88**
Cleveland	56.58	**57.59**	51.53	**52.53**	48.45	**49.81**	**58.25**	**58.25**	**45.09**	**45.09**	52.83	**53.50**
Dermatology	89.96	**93.86**	96.37	**98.31**	93.02	**93.30**	94.13	**94.69**	**96.09**	**96.09**	84.36	**92.18**
Ecoli	**80.96**	79.77	79.47	**80.07**	**75.89**	75.60	77.69	**78.28**	**75.31**	75.01	**82.75**	**82.75**
Flare	72.14	**72.32**	74.20	**74.29**	**72.61**	72.61	**74.67**	74.58	69.42	**73.17**	**72.98**	**72.98**
Glass	**73.38**	72.44	70.53	**71.47**	**74.30**	**74.30**	61.26	**61.27**	70.60	**71.54**	68.25	**68.72**
Led7digit	**72.60**	72.20	72.20	**72.60**	70.00	**71.60**	73.00	**73.40**	70.20	**71.40**	71.80	**72.60**
Lymphography	**83.72**	**83.72**	73.63	**76.30**	76.28	**77.63**	81.68	**82.34**	**80.34**	**80.34**	**78.94**	**78.94**
Nursery	**92.52**	**92.52**	**89.04**	88.89	**89.43**	89.28	**91.90**	91.74	81.33	**89.12**	**96.84**	**96.84**
Pageblocks	**94.70**	94.70	95.61	**95.79**	**95.25**	95.25	**94.70**	94.52	94.16	**94.52**	**95.43**	**95.43**
Penbased	96.18	**96.55**	90.64	**90.73**	89.91	**90.82**	95.27	**95.64**	97.82	**97.91**	**98.27**	**98.27**
Satimage	**86.47**	86.16	81.65	**82.74**	79.78	**80.40**	**84.14**	83.98	84.92	**85.39**	**87.41**	**87.41**
Segment	96.71	**96.75**	**97.06**	96.97	95.84	**96.41**	**92.55**	92.51	**97.10**	96.06	**96.62**	96.54
Shuttle	99.54	**99.63**	**99.72**	99.68	99.49	**99.68**	96.37	**97.43**	**99.72**	99.68	97.47	**98.21**
Vehicle	**71.40**	**71.40**	71.39	**72.81**	71.04	**71.75**	72.46	**72.58**	**80.49**	80.26	83.57	**83.69**
Vowel	95.86	**96.06**	80.00	**81.21**	**78.99**	78.48	69.90	**70.20**	**99.39**	**99.39**	**97.98**	**97.98**
Yeast	**54.72**	54.18	**59.91**	59.57	56.00	**56.07**	59.10	**59.23**	**56.54**	55.86	59.10	**59.23**
Zoo	92.10	**93.10**	**93.10**	**93.10**	94.10	**95.10**	95.05	**96.05**	84.19	**85.19**	**97.05**	**97.05**
Mean	83.11	**83.51**	81.59	**82.12**	80.54	**81.04**	81.03	**81.18**	79.73	**81.51**	84.12	**84.71**

better results among a variety of data-sets and base classifiers, being not only appropriate for a unique base learner, but having a robust behavior among all the classifiers considered. Furthermore, we should emphasize that not only the accuracy is improved, but the number of classifiers (the size of the ensemble) is also reduced.

5.2. Dynamic OVO (with best aggregations) vs. state-of-the-art aggregations

Afterwards, we aim to complement the previous study by making a similar comparison. We will show the performance of Dynamic OVO in combination with the representative aggregation in each base classifier instead of always using the WV strategy in conjunction with the DCS. As mentioned earlier, we want to show that any of the classic aggregations can be used after the DCS; nonetheless, in this case we must emphasize that the aggregation that was the best performer using the whole score-matrix does not also need to be the best in the reduced score-matrix.

In Table 6, the results for each base learner are shown. As in the previous table of results, Dynamic OVO (referred as Dyn) is accompanied by the superscript indicating the aggregation used, which is the best performer aggregation in each base classifier. As well as in the previous table, the best result for each base classifier and data-set is stressed.

Observing Table 6, the dynamic procedure continues maintaining an advantage in comparison with the classic aggregations. The mean accuracy is nearly the same, in case of 3NN there is a small improvement, more evident is the enhancement of PDFC. On the other hand, both SVMs' configurations have slightly decrease its accuracy with respect to Table 4. Notice that C4.5 and Ripper maintain its results because their best performer aggregation is the WV. Anyway, we must carry out the corresponding statistical tests in order to achieve well-founded conclusions. The results of the tests are shown in Table 7.

From Table 6, we can observe that the improvement is not so remarkable. In case of PDFC, the p-value has decreased, showing a better performance; obviously, for C4.5 and Ripper the dynamic strategy continues outperforming the WV. With respect to SVMs, the p-values have increased, and hence, using the same aggregation Dynamic OVO is not able to statistically outperform the PE aggregation. However, the p-values are low, showing an important advantage of the dynamic strategy. Finally, regarding 3NN, statistical differences are not found either, but the ranks show a better behavior of Dynamic OVO.

From these tests we can conclude that, the best performer aggregation for the DCS does not need to be the same as the best performer in the non-dynamic case. For this reason, we propose to use the Dynamic OVO in conjunction with the WV strategy, whose positive synergy has been demonstrated in the previous subsection. Anyway, despite the aggregation used in the second phase, we must stressed that the dynamic selection actually leads to the avoidance of non-competent classifiers, which are hindering the classification performance; this fact has been demonstrated by the enhancement of the results. Obviously, there are also cases in which we can lose some classes which are interesting (there are data-sets where the accuracy slightly decreases), but they are less

Table 7
Wilcoxon tests to compare the representative aggregations and the dynamic OVO with different base classifiers. R^+ corresponds to the sum of the ranks for Dynamic OVO and R^- for the representative aggregation.

Classifier	Comparison	R^+	R^-	Hypothesis	p-Value
3NN	DynND vs. ND	119.0	71.0	Not rejected	0.351979
C4.5	Dyn^{WV} vs. WV	153.5	36.5	**Rejected for Dyn^{WV} at 5%**	0.017621
Ripper	Dyn^{WV} vs. WV	162.5	27.5	**Rejected for Dyn^{WV} at 5%**	0.005684
SVM$_{Poly}$	Dyn^{PE} vs. PE	133.5	56.5	Not Rejected	0.112779
SVM$_{Puk}$	Dyn^{PE} vs. PE	127.0	63.0	Not Rejected	0.139756
PDFC	Dyn^{PC} vs. PC	151.5	38.5	**Rejected for Dyn^{PC} at 5%**	0.010862

Table 8
Average accuracy results in test of the representative aggregations and the dynamic strategy (with WV) for each base classifier considering large data-sets (nursery, pageblocks, penbased, satimage and shuttle).

Data-set	3NN		C4.5		Ripper		SVM$_{Poly}$		SVM$_{Puk}$		PDFC	
	ND	Dyn^{WV}	WV	Dyn^{WV}	WV	Dyn^{WV}	PE	Dyn^{WV}	PE	Dyn^{WV}	PC	Dyn^{WV}
Autos	76.75	**77.38**	**81.17**	79.92	78.65	**79.27**	74.80	**75.42**	**68.53**	67.88	76.71	**78.61**
Car	92.82	**93.06**	93.00	**93.29**	91.32	**92.36**	**92.71**	92.65	63.60	**84.72**	99.88	**99.94**
Cleveland	56.58	**56.58**	51.53	**52.53**	48.45	**49.81**	**58.25**	57.59	45.09	**45.42**	52.83	**53.85**
Dermatology	89.96	**93.58**	96.37	**98.31**	93.02	**93.30**	94.13	**94.69**	**96.09**	**96.09**	84.36	**92.18**
Ecoli	**80.96**	80.66	79.47	**80.07**	**75.89**	75.60	77.69	**77.99**	75.31	**75.90**	**82.75**	**82.75**
Flare	**72.14**	71.95	74.20	**74.29**	**72.61**	72.61	74.67	**75.33**	69.42	**72.89**	**72.98**	**72.98**
Glass	**73.38**	**73.38**	70.53	**71.47**	**74.30**	**74.30**	61.26	**62.66**	70.60	**71.09**	68.25	**70.11**
Led7digit	72.60	**72.80**	72.20	**72.60**	70.00	**71.60**	73.00	**74.00**	70.20	**71.40**	71.80	**72.60**
Lymphography	**83.72**	**83.72**	73.63	**76.30**	76.28	**77.63**	**81.68**	**81.68**	**80.34**	**80.34**	**78.94**	**78.94**
Nursery	98.61	**98.69**	96.71	**96.74**	97.79	**97.93**	93.06	**93.55**	99.22	**99.68**	**99.98**	**99.98**
Pageblocks	**96.71**	96.66	97.08	**97.11**	96.56	**96.75**	94.54	**94.81**	**96.80**	96.78	**95.91**	95.87
Penbased	99.40	**99.41**	96.95	**97.66**	96.90	**97.51**	98.00	**98.52**	99.59	**99.59**	**99.50**	99.49
Satimage	91.19	**91.20**	87.15	**87.71**	86.95	**87.01**	85.00	**86.05**	**91.55**	91.52	89.17	**89.34**
Segment	96.71	**96.97**	**97.06**	96.97	95.84	**96.41**	92.55	**92.60**	**97.10**	97.06	**96.62**	96.54
Shuttle	**99.92**	99.92	**99.98**	99.97	**99.96**	99.95	97.18	**99.49**	99.85	**99.87**	99.52	**99.68**
Vehicle	**71.40**	**71.40**	71.39	**72.81**	71.04	**71.75**	72.46	**73.29**	80.49	**80.61**	83.57	**83.69**
Vowel	**95.86**	**95.86**	80.00	**81.21**	**78.99**	78.48	69.90	**70.30**	**99.39**	**99.39**	**97.98**	**97.98**
Yeast	**54.72**	54.52	**59.91**	59.57	56.00	**56.07**	**59.10**	58.96	**56.54**	56.27	**59.10**	59.03
Zoo	92.10	**94.10**	**93.10**	**93.10**	94.10	**95.10**	95.05	**96.05**	84.19	**85.19**	**97.05**	**97.05**
Total	83.98	**84.31**	82.71	**83.24**	81.82	**82.29**	81.32	**81.87**	81.26	**82.72**	84.57	**85.30**

Table 9
Wilcoxon tests to compare the representative aggregations and the dynamic OVO with different base classifiers considering large data-sets. R^+ corresponds to the sum of the ranks for Dynamic OVO and R^- for the representative aggregation.

Classifier	Comparison	R^+	R^-	Hypothesis	p-Value
3NN	Dyn^{WV} vs. ND	131.0	59.0	Not rejected	0.139756
C4.5	Dyn^{WV} vs. WV	158.5	31.5	**Rejected for Dyn^{WV} at 5%**	0.010844
Ripper	Dyn^{WV} vs. WV	165.5	24.5	**Rejected for Dyn^{WV} at 5%**	0.004337
SVM$_{Poly}$	Dyn^{WV} vs. PE	169.5	20.5	**Rejected for Dyn^{WV} at 5%**	0.002849
SVM$_{Puk}$	Dyn^{WV} vs. PE	142.0	48.0	**Rejected for Dyn^{WV} at 5%**	0.049422
PDFC	Dyn^{WV} vs. PC	143.5	46.5	**Rejected for Dyn^{WV} at 5%**	0.019223

than the cases in which the procedure helps to remove incompetent classifiers. Moreover, these cases could be misclassified due to a failure in the competent classifiers, which was corrected by the rest of the competent classifiers, but due to their elimination, they can no longer correct the decision; besides, this cases might be corrected by increasing the value of k.

5.3. Analyzing the behavior of Dynamic OVO with large data-sets

In this subsection, we carry out the same comparison as that of Section 5.1, but in this case considering the non-reduced version of the large data-sets. This way, we will also show the usefulness of this strategy in the context of large data-sets. Table 8 shows the test accuracy results following the same format used in the previous tables of results.

In this scenario, the results obtained by the dynamic strategy seems to remain unchanged. However, we need to contrast this fact with the proper statistical tests in order to extract meaningful conclusions. The results of the corresponding Wilcoxon tests are presented in Table 9. From this table, and comparing it to Table 5 (reduced version of the data-sets), we can conclude that the obtained results are equivalent. Hence, the superiority of the dynamic approach also stands out when large data-sets are considered, that is, its behavior does not worsen with increasing number of instances.

Table 10
Average accuracy results in test of the representative aggregations and the dynamic strategy (with WV) for each base classifier considering 5-fold SCV carried out with DOB-SCV partitioning.

Data-set	3NN		C4.5		Ripper		SVM$_{Poly}$		SVM$_{Puk}$		PDFC	
	ND	Dyn^{WV}	WV	Dyn^{WV}	WV	Dyn^{WV}	PE	Dyn^{WV}	PE	Dyn^{WV}	PC	Dyn^{WV}
Autos	**78.88**	76.96	**76.24**	74.96	**85.09**	84.42	73.75	**73.81**	69.02	**70.27**	78.82	**79.40**
Car	**93.57**	93.40	**94.68**	94.50	92.59	**93.52**	**93.58**	**93.58**	64.99	**84.84**	99.77	**99.88**
Cleveland	**58.31**	57.96	52.55	**53.55**	52.18	**54.54**	58.97	**59.31**	47.53	**47.87**	53.92	**55.93**
Dermatology	92.14	**95.49**	95.24	**98.32**	93.32	**94.43**	94.71	**94.99**	**97.20**	**97.20**	84.66	**93.85**
Ecoli	81.66	**82.52**	81.06	**81.94**	78.47	**78.74**	79.37	**79.63**	**77.11**	**77.11**	**84.07**	83.78
Flare	71.21	**71.59**	**75.34**	73.62	**75.24**	74.83	75.43	**75.46**	69.28	**73.39**	73.64	**73.92**
Glass	73.35	**74.27**	**72.03**	71.63	**68.56**	68.12	62.14	**63.14**	73.72	**74.15**	68.72	**70.12**
Led7digit	66.68	**67.88**	64.51	**65.35**	**63.98**	63.86	67.90	**68.09**	61.33	**61.57**	62.17	**62.60**
Lymphography	68.19	**79.55**	74.50	**76.44**	**75.68**	**75.68**	**82.48**	**82.48**	**81.87**	**81.87**	**83.19**	**83.19**
Nursery	**93.29**	**93.29**	89.66	**89.81**	90.66	**90.81**	**92.13**	**92.13**	80.33	**89.05**	**97.92**	**97.92**
Pageblocks	**95.63**	95.46	**95.64**	95.46	**95.45**	95.11	**94.90**	94.53	94.58	**94.76**	**95.09**	94.91
Penbased	**97.00**	96.64	91.10	**91.11**	**91.38**	91.11	95.92	**96.01**	97.55	**97.64**	**98.19**	98.10
Satimage	87.58	**87.73**	82.15	**82.92**	**82.61**	82.14	**84.48**	84.16	84.77	**85.70**	86.79	**86.95**
Segment	96.58	**96.80**	96.28	**96.71**	96.58	**96.88**	92.68	**92.90**	97.23	**97.36**	97.32	**97.36**
Shuttle	**99.50**	99.40	99.59	**99.68**	99.40	**99.68**	96.55	**97.61**	99.59	**99.63**	97.43	**98.03**
Vehicle	72.11	**72.23**	72.33	**72.81**	69.27	**70.20**	73.53	**74.00**	**81.92**	**81.92**	**84.53**	84.40
Vowel	**97.78**	97.37	83.43	**83.64**	**80.20**	79.39	71.41	**71.82**	**99.70**	**99.70**	**98.28**	98.08
Yeast	**56.68**	56.54	59.57	**59.84**	**58.30**	58.10	**60.52**	59.98	59.31	**59.51**	**60.25**	59.98
Zoo	89.90	**91.86**	**92.17**	**92.17**	**94.05**	**94.05**	**95.72**	**95.72**	78.06	**84.13**	**96.77**	**96.77**
Total	82.63	**83.52**	81.48	**81.81**	81.21	**81.35**	81.38	**81.55**	79.74	**81.98**	84.29	**85.01**

Table 11
Wilcoxon tests to compare the representative aggregations and the dynamic OVO with different base classifiers considering 5-fold SCV with DOB-SCV partitioning. R^+ corresponds to the sum of the ranks for Dynamic OVO and R^- for the representative aggregation.

Classifier	Comparison	R^+	R^-	Hypothesis	p-Value
3NN	Dyn$_{WV}$ vs. ND	123.5	66.5	Not rejected	0.231059
C4.5	Dyn$_{WV}$ vs. WV	136.5	53.5	Not rejected	0.102440
Ripper	Dyn$_{WV}$ vs. WV	98.5	91.5	Not rejected	0.831310
SVM$_{Poly}$	Dyn$_{WV}$ vs. PE	142.0	48.0	**Rejected for Dyn$_{WV}$ at 10%**	0.099540
SVM$_{Puk}$	Dyn$_{WV}$ vs. PE	182.5	7.5	**Rejected for Dyn$_{WV}$ at 5%**	0.000982
PDFC	Dyn$_{WV}$ vs. PC	132.0	58.0	Not rejected	0.108941

5.4. Studying the performance of Dynamic OVO with DOB-SCV data-partitioning method

In order to complete the experimental study, we have performed another comparison considering a different data-partitioning method, DOB-SCV (with 5 folds), which aims to avoid the data-set shift that might be produced by the standard stratified cross-validation. In this manner, we want to show the robustness of the dynamic approach. The results obtained for each base classifier and data-set (we use the reduced version of the data-sets) are shown in Table 10, following the same output format as that of the previous results.

Observing Table 10, the differences are apparently maintained in most of the cases, except for Ripper. Moreover, in some cases they have increased (e.g., SVM$_{Puk}$). Anyway, we should look at the corresponding Wilcoxon test to show whether statistical differences exist or not (Table 11). The null hypothesis of equivalence has been rejected in two out of six cases (SVM$_{Puk}$ and SVM$_{Poly}$). Moreover, we must stress that in the case of C4.5 and PDFC the p-values are low, besides in the other two cases, the ranks are in favor of the dynamic approach. These facts, together with the previously presented results, show that Dynamic OVO, despite the experimental framework considered, allows one to enhance the results obtained by the state-of-the-art aggregations eliminating non-competent classifiers from the decision process. Hence, it can also be concluded that these classifiers can hinder the decisions taken by the aggregations.

6. Concluding remarks

In this paper, we have presented a DCS procedure to overcome the non-competent classifiers problem in OVO strategy. In order to do so, we have proposed to use the neighborhood of each instance in such a way that only those classes which are in the neighborhood are considered in the new score-matrix. Hence, we remove those classifiers that we are sure enough that must not be considered. Finally, any aggregation can be used to decide over the new score-matrix.

The novel procedure proposed has shown its usefulness despite its simplicity. We have shown the positive synergy existing between Dynamic OVO and the WV strategy, which has been able to statistically outperform the best state-of-the-art aggregations in five out of six base classifiers (in all except 3NN, for which a low p-value was obtained in the comparison). In addition, we have analyzed the behavior of Dynamic OVO with large data-sets and considering a different data-partition method. Both experiments have shown that the performance improvement obtained is maintained in spite of the experimental framework considered.

Furthermore, we must stress that all the differences found in this paper are due to the aggregations studied, and not due to differences in the base classifiers. All the aggregations base their decision on the same score-matrix (their start point is the same); for this reason, these results are meaningful since in spite of using the same initial score-matrix Dynamic OVO is able to achieve significant differences with respect to the state-of-the-art aggregations, which were not found in the majority of the comparisons carried out in [25].

In the future, several works remain to be addressed. Among them, the scalability of OVO and more specifically, of our approach must be studied; on the other hand, different ways of DCS procedures should be studied. It seems difficult to obtain differences only considering the same score-matrix and nothing else, however, observing the enhancement of the results that we have obtained, DCS seems to be a promising area to be studied within OVO and OVA, and more generally, within ECOC scenario. Clustering-based competence should also be studied in this framework. Such an approach could reduce the sensitivity to noise and the complexity in testing time by translating it to the training process, but it must be analyzed whether the provided results are maintained.

Conflict of interest statement

None declared.

Acknowledgments

This work has been supported by the Spanish Ministry of Education and Science under Projects TIN2010-15055 and TIN2011-28488 and the Andalusian Research Plan P10-TIC-6858.

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 (2010) 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) (2008) 307–318.

[4] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multi-class to binary: a unifying approach for margin classifiers, Journal of Machine Learning Research 1 (2000) 113–141.

[5] R. Avnimelech, N. Intrator, Boosted mixture of experts: an ensemble learning scheme, Neural Computation 11 (2) (1999) 483–497.

[6] L. Batista, E. Granger, R. Sabourin, Dynamic selection of generative-discriminative ensembles for off-line signature verification, Pattern Recognition 45 (4), 1326–1340, http://dx.doi.org/10.1016/j.patcog.2011.10.011.

[7] V. Bolón-Canedo, N.S.-M. no, A. Alonso-Betanzos, An ensemble of filters and classifiers for microarray data classification, Pattern Recognition 45 (1) (2012) 531–539.

[8] A.M.P. Canuto, M.C.C. Abreu, L. de Melo Oliveira, J.C. Xavier Jr., A.d.M. Santos, Investigating the influence of the choice of the ensemble members in accuracy and diversity of selection-based and fusion-based methods for ensembles, Pattern Recognition Letters 28 (4) (2007) 472–486.

[9] P.R. Cavalin, R. Sabourin, C.Y. Suen, LoGID: an adaptive framework combining local and global incremental learning for dynamic selection of ensembles of HMMs, Pattern Recognition 45 (9) (2012) 3544–3556.

[10] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (2011). pp. 27:1–27:27. Available at ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩..

[11] Y. Chen, J.Z. Wang, Support vector learning for fuzzy rule-based classification systems, IEEE Transactions Fuzzy Systems 11 (6) (2003) 716–728.

[12] 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.

[13] J. Cohen, A coefficient of agreement for nominal scales, Educational and Psychological Measurement 20 (1) (1960) 37–46.

[14] W.W. Cohen, Fast effective rule induction, in: ICML'95: Proceedings of the Twelfth International Conference on Machine Learning, 1995.

[15] J. Demšar, Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research 7 (2006) 1–30.

[16] L. Didaci, G. Giacinto, F. Roli, G. Marcialis, A study on the performances of dynamic classifier selection based on local accuracy estimation, Pattern Recognition 38 (11) (2005) 2188–2191.

[17] T.G. Dietterich, G. Bakiri, Solving multi-class learning problems via error-correcting output codes, Journal of Artificial Intelligence Research 2 (1995) 263–286.

[18] E. Dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognition 41 (10) (2008) 2993–3009.

[19] E.M. Dos Santos, R. Sabourin, P. Maupin, A dynamic overproduce-and-choose strategy for the selection of classifier ensembles, Pattern Recognition 41 (10) (2008) 2993–3009.

[20] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, 2nd edition, John Wiley, 2001.

[21] 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.

[22] 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⟩.

[23] J. Fürnkranz, Round robin classification, Journal of Machine Learning Research 2 (2002) 721–747.

[24] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Aggregation Schemes for Binarization Techniques Methods' Description, Technical Report, Research Group on Soft Computing and Intelligent Information Systems, 2011. URL ⟨http://sci2s.ugr.es/ovo-ova/AggregationMethodsDescription.pdf⟩.

[25] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, An overview of ensemble methods for binary classifiers in multi-class problems: experimental study on one-vs-one and one-vs-all schemes, Pattern Recognition 44 (8) (2011) 1761–1776.

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

[27] S. García, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (3) (2012) 417–435.

[28] 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.

[29] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multi-class pattern recognition by the combination of two strategies, IEEE Transactions on Pattern Analysis and Machine Intelligence 28 (6) (2006) 1001–1006.

[30] V. Gunes, M. Ménard, P. Loonis, S. Petit-Renaud, Combination cooperation and selection of classifiers: a state of the art, International Journal of Pattern Recognition and Artificial Intelligence 17 (8) (2003) 1303–1324.

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

[32] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Annals of Statistics 26 (2) (1998) 451–471.

[33] 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.

[34] C.W. Hsu, C.J. Lin, A comparison of methods for multi-class support vector machines, IEEE Transactions on Neural Networks 13 (2) (2002) 415–425.

[35] 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.

[36] N.M. Khan, R. Ksantini, I.S. Ahmad, B. Boufama, A novel svm+nda model for classification with an application to face recognition, Pattern Recognition 45 (1) (2012) 66–79.

[37] 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. F68Springer-Verlag, 1990, pp. 41–50.

[38] A. Ko, R. Sabourin, A. Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (5) (2008) 1735–1748.

[39] A.H.R. Ko, R. Sabourin, A.S. Britto Jr., From dynamic classifier selection to dynamic ensemble selection, Pattern Recognition 41 (5) (2008) 1718–1731.

[40] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: an experiment, IEEE Transactions on Systems, Man, and Cybernetics. Part B. Cybernetics 32 (2) (2002) 146–156.

[41] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, Wiley-Interscience, 2004.

[42] 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.

[43] R. Liu, B. Yuan, Multiple classifiers combination by clustering and selection, Information Fusion 2 (3) (2001) 163–168.

[44] A.C. Lorena, A.C. Carvalho, J.M. Gama, A review on the combination of binary classifiers in multi-class problems, Artificial Intelligence Review 30 (1–4) (2008) 19–37.

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

[46] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on *k*-fold cross-validation, IEEE Transactions on Neural Networks and Learning Systems 23 (8) (2012) 1304–1312.

[47] X.-X. Niu, C.Y. Suen, A novel hybrid CNN-SVM classifier for recognizing handwritten digits, Pattern Recognition 45 (4) (2012) 1318–1325.
[48] J.C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, MIT Press, Cambridge, MA, USA, 1999.
[49] O. Pujol, S. Escalera, P. Radeva, An incremental node embedding technique for error correcting output codes, Pattern Recognition 41 (2) (2008) 713–725.
[50] J.R. Quinlan, C4.5: Programs for Machine Learning, 1st edition, Morgan Kaufmann Publishers, San Mateo-California, 1993.
[51] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[52] R. Rifkin, A. Klautau, In defense of one-vs-all classification, Journal of Machine Learning Research 5 (2004) 101–141.
[53] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (2010) 1–39.
[54] H. Shin, S. Sohn, Selected tree classifier combination based on both accuracy and error diversity, Pattern Recognition 38 (2) (2005) 191–197.
[55] V. Vapnik, Statistical Learning Theory, John Wiley, New York, 1998.
[56] A. Verikas, A. Lipnickas, K. Malmqvist, M. Bacauskiene, A. Gelzinis, Soft combination of neural classifiers: a comparative study, Pattern Recognition Letters 20 (4) (1999) 429–444.
[57] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bulletin 1 (6) (1945) 80–83.
[58] T. Woloszynski, M. Kurzynski, A probabilistic model of classifier competence for dynamic ensemble selection, Pattern Recognition 44 (10–11) (2011) 2656–2668.
[59] K. Woods, W. Philip Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (4) (1997) 405–410.
[60] 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.
[61] J.D. Zhou, X.D. Wang, H.J. Zhou, J.M. Zhang, N. Jia, Decoding design based on posterior probabilities in ternary error-correcting output codes, Pattern Recognition 45 (4) (2012) 1802–1818.

Mikel Galar received the M.Sc. and Ph.D. degrees in Computer Science in 2009 and 2012, both from the Public University of Navarra, Pamplona, Spain. He is currently a teaching assistant in the Department of Automatics and Computation at the Public University of Navarra. His research interests are data-mining, classification, multi-classification, ensemble learning, evolutionary algorithms and fuzzy systems.

Alberto Fernández received the M.Sc. and Ph.D. degrees in computer science in 2005 and 2010, both from the University of Granada, Granada, Spain.

He is currently an 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, resolution of multi-classification problems with ensembles and decomposition techniques, and Business Intelligence in Cloud Computing.

He have been also awarded with the “Lofti A. Zadeh Prize” of the International Fuzzy Systems Association (IFSA) for the “Best paper on 2009-2010” for the work Hierarchical fuzzy rule based classification system with genetic rule selection for imbalanced data-sets.

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 an Associate Lecturer. She obtained the Ph.D. in Computer Science in 2005 on the topic interval-valued fuzzy sets applied to image processing. Her publications comprise more than 20 papers in international journals and about 15 book chapters. Her research interests are fuzzy techniques for image processing, fuzzy sets theory, interval type-2 fuzzy sets theory and applications, decision making, and medical and industrial applications of soft computing techniques. She is a member of the board of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Humberto Bustince received his B.Sc. degree on Physics from the Salamanca University, Spain, in 1983 and his Ph.D. degree in Mathematics from the Public University of Navarra, Pamplona, Spain, in 1994. He has been a teacher at the Public University of Navarra since 1991, and he is currently a Full Professor with the Department of Automatics and Computation. He served as subdirector of the Technical School for Industrial Engineering and Telecommunications from 01/01/2003 to 30/10/2008 and he was involved in the implantation of Computer Science courses at the Public University of Navarra. He is currently involved in teaching artificial intelligence for students of computer sciences. Dr. Bustince has authored more than 70 journal papers (Web of Knowledge), and more than 73 contributions to international conferences. He has also been co-author of four books on fuzzy theory and extensions of fuzzy sets.

He is fellow of the IEEE Computational Intelligence Systems society and Member of the board of the European Society for Fuzzy Logic and Applications (EUSFLAT). He currently acts as Editor in chief of the Mathware & Soft Computing Magazine and of Notes on Intuitionistic Fuzzy Sets. He is also guest editor of the Fuzzy Sets and Systems journal and member of the editorial board of the IEEE Transactions on Fuzzy Systems, the Journal of Intelligence & Fuzzy Systems, the International Journal of Computational Intelligence Systems and the Axioms Journal.

His current research interests include interval-valued fuzzy sets, Atanassov's intuitionistic fuzzy sets, aggregation functions, implication operators, inclusion measures, image processing, decision making and approximate reasoning.

Francisco Herrera He currently acts as Editor in Chief of the international journal “Progress in Artificial Intelligence (Springer). He acts as an area editor of the International Journal of Computational Intelligence Systems and associated ereceived his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has been the supervisor of 28 Ph.D. students. He has published more than 240 papers in international journals. He is coauthor of the book “*Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases*" (World Scientific, 2001). ditor of the journals: IEEE Transactions on Fuzzy Systems, Information Sciences, Knowledge and Information Systems, Advances in Fuzzy Systems, and International Journal of Applied Metaheuristics Computing; and he serves as a member of several journal editorial boards, among others: Fuzzy Sets and Systems, Applied Intelligence, Information Fusion, Evolutionary Intelligence, International Journal of Hybrid Intelligent Systems, Memetic Computation, and Swarm and Evolutionary Computation.

He received the following honors and awards: ECCAI Fellow 2009, IFSA 2013 Fellow, 2010 Spanish National Award on Computer Science ARITMEL to the "Spanish Engineer on Computer Science", International Cajastur "Mamdani" Prize for Soft Computing (Fourth Edition, 2010), IEEE Transactions on Fuzzy System Outstanding 2008 Paper Award (bestowed in 2011), and 2011 Lotfi A. Zadeh Prize Best paper Award of the International Fuzzy Systems Association.

His current research interests include computing with words and decision making, bibliometrics, data mining, big data, data preparation, instance selection, fuzzy rule based systems, genetic fuzzy systems, knowledge extraction based on evolutionary algorithms, memetic algorithms and genetic algorithms.