Pattern Recognition 48 (2015) 28–42

Contents lists available at ScienceDirect

Pattern Recognition

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

DRCW-OVO: Distance-based relative competence weighting combination for One-vs-One strategy in multi-class problems

Mikel Galar [a,*], Alberto Fernández [b], Edurne Barrenechea [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 2 October 2013
Received in revised form
30 June 2014
Accepted 18 July 2014
Available online 2 August 2014

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

ABSTRACT

One-vs-One strategy is a common and established technique in Machine Learning to deal with multi-class classification problems. It consists of dividing the original multi-class problem into easier-to-solve binary subproblems considering each possible pair of classes. Since several classifiers are learned, their combination becomes crucial in order to predict the class of new instances. Due to the division procedure a series of difficulties emerge at this stage, such as the non-competence problem. Each classifier is learned using only the instances of its corresponding pair of classes, and hence, it is not competent to classify instances belonging to the rest of the classes; nevertheless, at classification time all the outputs of the classifiers are taken into account because the competence cannot be known a priori (the classification problem would be solved). On this account, we develop a distance-based combination strategy, which weights the competence of the outputs of the base classifiers depending on the closeness of the query instance to each one of the classes. Our aim is to reduce the effect of the non-competent classifiers, enhancing the results obtained by the state-of-the-art combinations for One-vs-One strategy. We carry out a thorough experimental study, supported by the proper statistical analysis, showing that the results obtained by the proposed method outperform, both in terms of accuracy and kappa measures, the previous combinations for One-vs-One strategy.

1. Introduction

Decomposition strategies [37] are commonly applied to deal with classification problems with multiple classes [41,31]. They allow the usage of binary classifiers in multi-class problems, but they also ease the learning process following the divide-and-conquer strategy. As a consequence of facilitating the learning task, part of the difficulties are shifted to the combination stage, where the classifiers learned to solve each binary problem should be aggregated to output the final decision over the class label [35,53]. Most of these strategies can be included within Error Correcting Output Codes (ECOC) [13,4] framework. Among them, One-vs-One (OVO) [32] scheme is one of the most popular techniques, used in very well-known software tools such as WEKA [25], LIBSVM [8] or KEEL [3] to model multi-class problems for Support Vector Machines (SVMs) [50]. Its usage to deal with real-world applications is also frequent, being a simple yet effective way of overcoming multi-class problems. These multi-class problems are divided into binary subproblems (as many as possible pairs of classes), which are independently learned by different base classifiers whose outputs are then combined to classify an instance.

This final combination phase is a key factor in ensembles of classifiers [35,15]. Several combination mechanisms for OVO strategy can be found in the specialized literature [21]. The voting strategy is the most intuitive one. In this strategy each classifier gives a vote for its predicted class and that reaching the largest number of votes is predicted by the system.

In OVO, in contrast with classic ensemble methods, there are some difficulties inherent to the way in which the decomposition is carried out. Among them, the unclassifiable region when the voting strategy is used, that is, when there is a draw of votes among classes, has attracted a lot of attention from researchers [45,16,36]. Nevertheless, none of these approaches has been able to make the difference with respect to other simpler approaches such as the Weighted Voting (WV) [30] or those methods based on probability estimates [54]. A thorough empirical study on the combinations for OVO strategy can be found in [21], where the problem of non-competent classifiers in OVO strategy was pointed out as a future research line.

[*] 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), herrera@decsai.ugr.es (F. Herrera).

http://dx.doi.org/10.1016/j.patcog.2014.07.023
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

In spite of the low attention this problem has received in the literature, we aim to show that its correct management can lead to a significant enhancement of the results obtained. The term non-competence when referring to OVO classifiers comes from the fact that in testing time all the opinions of the classifiers are taken into account, even though there are classifiers that have not been trained with instances of the real class of the instance to be classified, and hence, their outputs should not be relevant to the classification. The real challenge emerges when these outputs affects the final decision of the system, hindering the labeling of the instance and leading to an incorrect classification.

It is important to stress that the competence of a classifier in OVO strategy cannot be established before classification. In such a case, the output class of the instance to be classified would be known, meaning that the classification problem would be solved before hand. Taking this issue into account, we aim to manage the non-competence using the information available prior to the classification, considering the closeness to the instances of each class in the training set. Our assumption is that the outputs considering nearer classes should be more competent, since the classifier has probably been trained with instances of that class.

In this paper, we present a Distance-based Relative Competence Weighting combination method for OVO (DRCW-OVO), which relies on how far are the nearest neighbors of each class in the problem to weight the outputs of each classifier in the final aggregation stage. This way, the larger the distance is, the lower weight the output has, and vice versa. We should emphasize that whereas neighbors-based dynamic weighting approaches have been already considered for classifier ensembles [35], they have not been adapted to OVO framework due to the way in which the ensemble is formed. Notice that in OVO, the area of competence is established a priori, and it does not take into account the input space, but the output space.

In order to test the validity of our proposal, we develop a thorough empirical study maintaining the same experimental framework used in [21] aiming to carry out a fair comparison. A set of nineteen real-world problems from the KEEL data-set repository[1] [3,2] are considered. We measure the performance of the classifiers based on both accuracy and Cohen's kappa metric [10]. The significance of the results is studied by the usage of the appropriate statistical tests as suggested in the literature [12,24]. The proposed strategy is tested using several well-known classifiers from different Machine Learning paradigms as base learners, namely, SVMs [50], decision trees [46], instance-based learning [1], fuzzy rule based systems [9] and decision lists [11]. Our approach is compared with the state-of-the-art combinations for the different base classifiers [21] and with a novel Dynamic Classifier Selection (DCS) approach [22], which also aims at solving the non-competent classifiers problem.

The rest of this paper is organized as follows. Section 2 recalls several concepts related to this work. Next, Section 3 presents our proposal to combine binary classifiers in OVO strategy, the DRCW-OVO method. The set-up of the experimental framework is presented in Section 4. The comparison of our proposal with the state-of-the-art methods is carried out in Section 5, where we also study the influence of the neighborhood size in DRCW-OVO and analyze how the weights are assigned. Finally, Section 6 concludes the paper.

2. Related works

In this section we first recall the basics of binarization, and more specifically, we describe OVO strategy and some of their combinations. Then, we recall some works related to our dynamic weighting approach.

2.1. One-vs-One decomposition scheme

Decomposition strategies have been widely used in the literature to address multi-class problems (see [37] for an extensive review). Most of them can be included within ECOC [13,4] framework, among which OVO is one of the most extended schemes, being established by default in several widely used software tools [3,25,8] to handle multi-class problems using SVMs. The fact that an accepted extension of SVMs to multiple classes has not been established yet has produced an increasing application of binarization techniques, which outperform other multi-class SVM approaches [29].

Apart from the works focused on SVMs [29,47], other authors have also shown the suitability and usefulness of binarization techniques with different base classifiers [19,21,49]. Moreover, empirical results in those papers have shown that the usage of OVO can enhance the results of the direct application of the baseline classifiers with inherent multi-class support. Several papers have also shown that OVO strategy is simple but powerful, being competitive with other more complex approaches, such as those based on ECOC [55,56,6] or those constructing a hierarchy among the classifiers [42,39], whose main objective is to reduce the number of classifiers considered in problems comprising a large number of classes.

In OVO, also known as Pairwise classification, the original m-class problem is divided into $m(m-1)/2$ two-class problems (one for each possible pair of classes). Then, each sub-problem is faced by a binary classifier, which ignores the instances having a different class label from the labels corresponding to the pair of classes it must distinguish. In fact, this ignorance is the source of the problem that we are trying to undertake in this paper, the so-called *non-competence*. That is, each classifier outputs a class label (out of its two classes learned) and a confidence on its prediction despite it is not able to distinguish among all classes. We must stress again that this is a key factor in our proposal, since we must be aware that this confidence might not be relevant for the decision process if the instance does not belong to the pair of classes learned by the classifier. After learning the classifiers, a new instance is classified into one of the m classes depending on all the outputs of the set of classifiers. In order to do so, it is usual to construct a score-matrix R containing these outputs, which are used to decide the final class:

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

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 combination can be used to infer the class, e.g., those presented in the following subsection or the previously mentioned voting strategy.

2.2. Combination strategies for OVO scheme

Several strategies for the decision process in OVO procedure have been proposed in the literature that intend to achieve the highest accuracy addressing different features of this inference step. In [21], we developed a thorough review and experimental

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

comparison considering the most-recent and well-known techniques. From this study, we were able to select the better suited combination strategies for different paradigms of classifiers.

In this study, it was also concluded that more complex approaches need not be better than simpler ones. For example, methods taking into account information of the classes to construct hierarchical models such as the Binary Tree of Classifiers (BTC) [16] and Nesting OVO [36], or approaches based on Directed Acyclic Graphs (DDAG) [45] performed similar to simpler approaches, even worse in many of the problems considered. For this reason, in this work we focus on the best combination models that were found in [21], which leaves these models out of the experimental comparison as they were found not to perform as well as those described hereafter.

Therefore, we consider the basis established in our former analysis, using the same experimental set-up and selecting the best combination strategies for the sake of simplicity and readability of the current manuscript. We consider this to be the fairest way to show the suitability of our proposal in all the base classifiers.

- Weighted Voting strategy (WV) [30] uses the confidence of each base classifier in each class to vote for it. The class with the largest total confidence is the final output class:

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

- Classification by Pairwise Coupling (PC) [26] aims to estimate the posterior probabilities of all the classes starting from the pairwise class probabilities. Therefore, being $r_{ij} = \text{Prob}(\mathit{Class}_i \mid \mathit{Class}_i \text{ or } \mathit{Class}_j)$, the method finds the best approximation of the class posterior probabilities $\widehat{\mathbf{p}} = (\widehat{p}_1, \ldots, \widehat{p}_m)$ according to the pairwise outputs. Finally, the class having the largest posterior probability is predicted:

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

The posterior probabilities ($\widehat{\mathbf{p}}$) are computed by minimizing the Kullback–Leibler (KL) distance between r_{ij} and μ_{ij}, where $\mu_{ij} = p_i/(p_i+p_j)$ and $r_{ji} = 1 - r_{ij}$.

- Non-Dominance Criterion (ND) [17] considers the score-matrix 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:

$$\mathit{Class} = \arg \max_{i=1,\ldots,m} \left\{ 1 - \max_{1 \le j \ne i \le m} r'_{ji} \right\} \tag{4}$$

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

- Wu, Lin and Weng Probability Estimates by Pairwise Coupling approach (PE) [54] is similar to PC, since it also estimates the posterior probabilities ($\mathbf{p}$) of each class from the pairwise probabilities. However, in this case, the formulation of the optimization problem is different, despite the same decision rule is used to classify an instance. PE optimizes the following equation:

$$\min_{\mathbf{p}} \sum_{i=1}^{m} \sum_{1 \le j \ne i \le m} (r_{ji} p_i - r_{ij} p_j)^2$$
$$\text{subject to} \quad \sum_{i=1}^{m} p_i = 1, \quad p_i \ge 0, \text{ for all } i \in \{1, \ldots, m\}. \tag{5}$$

A more extensive and detailed description of these methods with the original description of the source papers is available in [20].

In addition to these methods, we will consider a novel approach, in which we also aimed at getting rid of the non-competent classifiers by means of a Dynamic Classifier Selection (DCS) strategy [22]. In that work, the non-competent classifiers were dynamically removed in the classification phase, depending on whether both of the classes considered by the binary classifiers were present in the neighborhood of the instance to be classified. Hence, only the classifiers whose classes were on the neighborhood of the instance were used for its classification. On this account, the size of the neighborhood used was large ($3 \cdot m$, that is, three times the number of classes in the problem) compared with the usually considered one for nearest neighbors classifier [23]. The empirical results proved the validity of this new method, statistically outperforming the previously mentioned combinations in the same experimental framework as that used in [21], which is also considered in the present paper.

Remark 1. All these methods use exactly the same score-matrix values (Eq. (1)) to compute the final class, but they can obtain different results. We must emphasize the importance of this fact, since it allows us to fix the score-matrices of each base classifier, applying the combinations to the same outputs; hence, all the results shown in the experimental analysis will be due to the combinations themselves and not due to differences on the predictions of the base classifiers.

2.3. Dynamic classifier weighting methods

Dynamic Classifier Weighting (DCW) methods [18] are closely related to Dynamic Classifier Selection (DCS) [52] and Dynamic Ensemble Selection (DES) techniques [33], which are active research topics in classifier fusion. In these combination approaches, a set of classifiers from the ensemble (in DCS only one) are dynamically selected depending on their competence to classify the given instance. In this framework, the competence of a classifier refers to its ability to correctly classify the instance. As a consequence, the estimation of the competence is a key component in the design of these techniques. In the literature, several ways for its estimation have been developed, for example, by the usage of the local accuracy of each classifier [52,7] or by establishing the region of the input space in which the classifiers are experts [5,34]. The major difference between DCS and DES techniques with respect to DCW methods is that in the latter the votes of the classifiers are weighted depending on their competence instead of carrying out the removal of the classifiers that are not competent to classify the instance.

Although these terms are common in ensemble literature, in the case of decomposition techniques, and more specifically in OVO, they cannot be directly applied [21,48]. Classical ensembles are formed of base classifiers which are competent in the whole output space (classes in the problem) and hence, any of their combinations lead to an ensemble able to classify a new instance into any of the classes. In the case of OVO, each classifier is specialized in 2 out of the m classes, that is, each base classifier is focused on a part of the output space instead of the input space (as it occurs with ensembles). For these reasons, the application of DCS, DES or DCW techniques [35,18,52,33] in OVO is not straightforward. We should stress that in addition to the consideration of the non-competent classifiers, the novelty of our approach resides on the development of a DCW technique for decomposition-based strategies. Notice that in this case, neither local accuracies can be estimated (each classifier only distinguishes a pair of classes) nor the input space can be divided (all classifiers are competent in the whole input space, but not in the output space).

3. DRCW-OVO: Distance-based relative competence weighting for One-vs-One

In this section, we present our proposal to manage the non-competent classifiers in OVO strategy. First, we introduce some preliminary concepts and we present the hypothesis that has motivated our approach. Afterwards, we present its operation procedure and a simple illustrative example. Then, we discuss the computational complexity of the proposed method.

3.1. Preliminary concepts and hypothesis

As we have already mentioned, a classifier is non-competent to classify an instance whenever the real class of the instance to be classified is not one of the pair of classes which were learned by the classifier. Thereby, it is clear that in order to settle the competence of the classifiers the real class of the instance should be known, which is equivalent to solve the classification problem. Hence, the competence of the classifiers can only be estimated so that their decisions are properly handled, but it cannot be fixed without classifying the instance.

In general, non-competent classifiers in OVO need not hinder the classification, as long as the decisions of the competent classifiers are correct. For example, suppose the case of the voting strategy; if all the base classifiers considering the correct class of the instance for their training correctly classify the instance, then the output of the system will also be correct. Otherwise, if just only one of the competent classifiers fail, the final decision would also depend on the votes of the non-competent ones, which could lead to a misclassification of the instance.

Therefore, our hypothesis is that the number of examples whose classification could be corrected alleviating the negative effect of the non-competent classifiers during the final inference is significant. We develop our approach under the premise that non-competent classifiers might harm the decision process, even though this problem has not received much attention in the specialized literature.

Since the competence of each classifier cannot be known a priori, we aim to weight the outputs of the classifiers depending on their competence in each class. In order to define this competence, we consider the usage of the distance between the instance and each one of the classes. For this purpose, we use the distance to the k nearest neighbors of the corresponding class (we will show in the experimental study that the method is robust with respect to different values of k). Then, we consider higher weights for the outputs of the classes that are closer to the instance to be classified, since we suppose that the outputs of the classifiers corresponding to classes which are closer to the instance will probably be more competent than those corresponding to classes which are farther.

Remark 2. As we have previously mentioned, this weighting procedure is needed since neither DCS, DES nor DCW techniques suit our problem as they do with classic ensembles. In OVO, we cannot establish different regions for the classifiers in the input space or estimate their local accuracies among the whole set of classes.

3.2. DRCW-OVO combination

Since we are proposing a combination method, we assume that the base classifiers have been trained. Hence, the score-matrix of the instance to be classified is given by R (Eq. (1)).

Once the score-matrix has been obtained, the operating procedure of DRWC-OVO is as follows:

1. Compute the k nearest neighbors of each class for the given instance and store the average distances of the k neighbors of each class in a vector $\mathbf{d} = (d_1, \dots, d_m)$.
2. A new score-matrix R^w is created where the output r_{ij} of a classifier distinguishing classes i, j are weighted as follows:
$$r_{ij}^w = r_{ij} \cdot w_{ij}, \tag{6}$$
where w_{ij} is the relative competence of the classifier on the corresponding output computed as
$$w_{ij} = \frac{d_j^2}{d_i^2 + d_j^2}, \tag{7}$$
being d_i the distance of the instance to the nearest neighbor of class i.
3. Use weighted voting strategy on the modified score-matrix R^w to obtain the final class.

Notice that the modification of the outputs makes that the corresponding score-matrix R^w is no longer normalized, and hence all the previous combinations cannot be directly used. In our case we use WV method since its robustness has been both theoretically [30] and experimentally [21] proved. Considering this combination, steps 2) and 3) can be merged obtaining the output class as follows:

$$Class = \arg \max_{i=1,\dots,m} \sum_{1 \le j \ne i \le m} r_{ij} \cdot w_{ij} \tag{8}$$

We acknowledge that there are previous approaches to distance-weighted nearest neighbor [14], but the followed objectives are completely different as well as the way in which the weights are computed and applied, i.e., we use the distance to weight the outputs of the classifiers instead of using them in k NN to vote for each class. Notice also that the distance with respect to the k nearest neighbors of each class are used, that is, $k \cdot m$ neighbors are used and hence, taking $k=1$ is not the same as using 1NN classifier, because a neighbor for each class is obtained. We will show in the experimental study that this fact makes the algorithm robust to the selection of the value of k.

In this work, the nearest neighbors are computed using the Heterogeneous Value Difference Metric (HVDM) [51], which can properly handle nominal values on the contrary to the well-known Euclidean distance. HVDM between two input instances $\mathbf{x}$, $\mathbf{y}$ is computed as follows:

$$d_{\text{HVDM}}(\mathbf{x}, \mathbf{y}) = \sqrt{\sum_{a=1}^{t} d_a^2(x_a, y_a)} \tag{9}$$

where t is the number of attributes. The function $d_a(x, y)$ computes the distance between two values x and y for attribute a and is defined as

$$d_a(x,y) = \begin{cases} 1 & \text{if } x \text{ or } y \text{ is unknown;} \\ & \text{otherwise}\dots \\ normalized\_vdm_a(x,y) & \text{if } a \text{ is nominal} \\ normalized\_diff_a(x,y) & \text{if } a \text{ is linear.} \end{cases} \tag{10}$$

Hence, $d_a(x, y)$ uses different functions to compute the distance between the values of the attributes depending on the type of a (whether it is nominal or numerical). The functions considered in this paper are the following ones.

$$normalized\_vdm_a(x,y) = \sqrt{\sum_{i=1}^{m} \left| \frac{N_{a,x,i}}{N_{a,x}} - \frac{N_{a,y,i}}{N_{a,y}} \right|^2} \tag{11}$$

where $N_{a,x}$ is the number of instances in the training set with value x for attribute a, $N_{a,x,i}$ is the number of instances in the training set with value x for attribute a and whose output class is i.

$$normalized\_diff_a(x,y) = \frac{|x-y|}{4 \cdot \sigma_a} \tag{12}$$

being σ_a the standard deviation of the values of the attribute a in the training set. For the sake of brevity, we refer the reader to [51] for more details on this distance and the reason why these functions are considered.

3.3. Understanding DRCW-OVO

In this section, our aim is to explain the behavior of DRCW-OVO, which is complemented with the last section of the experimental study where we show how the weights are assigned to the classifiers. We should emphasize that our method weights the relative competence of the outputs of the classifiers, and consequently in the case of a non-competent classifier both outputs should be weighted as similarly as possible (that is, assigning a 0.5 to both of them); otherwise, in the case of a competent one, the weight for the output of the real class should be weighted more than the other. In this way, the outputs of the competent classifiers for the correct class are empowered, whereas non-competent ones are not. Hence, we assume that the closeness of each instance to the corresponding class can be used as a measure of competence. As a result, if both classes in a classifier are equally far from the instance, the competence will be the same, whereas in the case that the instance is closer to one of the classes, its output will get a higher weight.

Hereafter, we present an illustrative example of the application of DRCW-OVO so that its working procedure can be better understood. We acknowledge that this is only an example where the method would work as expected, correcting the misclassification of the instance due to the non-competent classifiers. Certainly, the real behavior and appropriateness of the method will be analyzed in the experimental study, which will show us whether our hypothesis is correct or not for a significant number of classifications.

Suppose that an instance $\mathbf{x}$, whose real class is known to be c_1 is going to be classified. After submitting it to all the base classifiers, the following score-matrix R has been obtained:

$$R(\mathbf{x}) = \left(\begin{array}{cccccc} & \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 \\ \mathbf{c}_1 & - & 0.55 & 0.45 & 0.80 & 0.90 \\ \mathbf{c}_2 & 0.45 & - & 0.55 & 1.00 & 0.80 \\ \mathbf{c}_3 & 0.55 & 0.45 & - & 0.45 & 0.40 \\ \mathbf{c}_4 & 0.20 & 0.00 & 0.55 & - & 0.10 \\ \mathbf{c}_5 & 0.10 & 0.20 & 0.60 & 0.90 & - \end{array}\right) \tag{13}$$

Applying the voting strategy (V) there would be a draw of votes between classes c_1 and c_2, whereas using the WV strategy class c_2 would be predicted due to the high confidences given by the classifiers considering c_2 and c_4, c_5, respectively (Eq. (14)). But it is important to point out that, actually, these votes come from non-competent classifiers, which are the source of the failure in the case of WV.

$$R(\mathbf{x}) = \left(\begin{array}{cccccc|c|c} & \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 & \text{V} & \text{WV} \\ \mathbf{c}_1 & - & 0.55 & 0.45 & 0.80 & 0.90 & 3 & 2.70 \\ \mathbf{c}_2 & 0.45 & - & 0.55 & 1.00 & 0.80 & 3 & \mathbf{2.80} \\ \mathbf{c}_3 & 0.55 & 0.45 & - & 0.45 & 0.40 & 1 & 1.85 \\ \mathbf{c}_4 & 0.20 & 0.00 & 0.55 & - & 0.10 & 1 & 0.85 \\ \mathbf{c}_5 & 0.10 & 0.20 & 0.60 & 0.90 & - & 2 & 1.80 \end{array}\right) \tag{14}$$

Otherwise, we consider the same score-matrix but DRCW-OVO is applied. First, the distances to the k nearest neighbors of each class ($\mathbf{d}$) are computed (notice that for the illustrative purpose of the example the value of k is not important). Suppose that in this case, $\mathbf{d} = (0.8, 0.9, 0.6, 1.2, 1.6)$. Based on these distances, we can compute a weight-matrix W to represent all the w_{ij} for $i,j = 1\ldots m$:

$$W(\mathbf{x}) = \left(\begin{array}{cccccc} & \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 \\ \mathbf{c}_1 & - & 0.56 & 0.36 & 0.69 & 0.80 \\ \mathbf{c}_2 & 0.44 & - & 0.31 & 0.64 & 0.76 \\ \mathbf{c}_3 & 0.64 & 0.69 & - & 0.80 & 0.88 \\ \mathbf{c}_4 & 0.31 & 0.36 & 0.20 & - & 0.64 \\ \mathbf{c}_5 & 0.20 & 0.24 & 0.12 & 0.36 & - \end{array}\right) \tag{15}$$

Applying this weight-matrix W to the score-matrix R, we obtain the modified score-matrix R^w, in which the WV is applied to obtain the predicted class using DRCW-OVO (Eq. (16)). As it can be observed, after applying the proposed methodology the classification has been corrected and now c_1, which is the real class of the instance, is predicted.

$$R^w(x) = \left(\begin{array}{cccccc|c} & \mathbf{c}_1 & \mathbf{c}_2 & \mathbf{c}_3 & \mathbf{c}_4 & \mathbf{c}_5 & \text{WV} \\ \mathbf{c}_1 & - & 0.31 & 0.16 & 0.55 & 0.72 & \mathbf{1.74} \\ \mathbf{c}_2 & 0.20 & - & 0.17 & 0.64 & 0.61 & 1.66 \\ \mathbf{c}_3 & 0.35 & 0.31 & - & 0.36 & 0.35 & 1.37 \\ \mathbf{c}_4 & 0.06 & 0.00 & 0.11 & - & 0.06 & 0.24 \\ \mathbf{c}_5 & 0.02 & 0.05 & 0.07 & 0.32 & - & 0.47 \end{array}\right) \tag{16}$$

3.4. On the computational complexity of DRCW-OVO

Our proposal requires to find the k nearest neighbors of each instance. Hence, it might be computationally more expensive than standard combination techniques, but this would also highly depend on the way of its implementation, since this process can be easily parallelized (both the computation of the distances and the testing phase with the classifiers forming the ensemble).

The search for the nearest neighbors has a complexity of $\mathcal{O}(n \cdot t)$, where n is the number of examples in the training set and t is the number of attributes. In case of dealing with large data-sets, an instance selection procedure [23] could be carried out in order to reduce the reference set and therefore, to reduce the testing time.

For the sake of comparison we should also provide the computational complexities of the other methods. In the case of the WV and ND, the complexity is constant $\mathcal{O}(1)$, since the values are directly aggregated from the score-matrix. PC and PE use an iterative method whose complexity is of $\mathcal{O}(m)$. These methods need less time to decide for the class since they do not consider more information than the score-matrix. However, the most accurate strategy from the state-of-the-art, the DCS model has also a complexity of $\mathcal{O}(n \cdot t)$ as the proposed one.

4. Experimental framework

In this section, we introduce the set-up of the experimental framework used to develop the empirical comparison in Section 5. We must emphasize that the whole experimental set-up is the same as that in [21], where the state-of-the-art on combinations for the OVO strategy were compared. From our point of view, this fact allows us to perform a fair comparison and to maintain the conclusions drawn from that study (e.g., reducing the length of the comparative study considering the best performers combinations).

First, we describe the base classifiers considered and their configuration in Subsection 4.1. Then, in Subsection 4.2, we recall

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 ω = 1.0, PukKernel σ = 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 = 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

which were the best combinations for each base classifier [21] that will be the baseline for the comparisons as explained in Section 2.2 together with the configuration for the DCS approach [22]. Afterwards, we provide details of the data-sets in Subsection 4.3 and we present the performance measures and the statistical tests used to make the comparison in Subsection 4.4.

4.1. Base learners and parameter configuration

Our aim in the empirical study is to compare our DRCW-OVO with the state-of-the-art combinations. For this purpose, we have selected several well-known Machine Learning algorithms as base learners. The algorithms used are the following ones:

- **SVM** [50] *Support Vector Machine* maps the original input space into a high-dimensional feature space using 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, i.e., that reaching the maximal margin such that the upper bound of the expected risk is minimized. We use SMO [43] algorithm to train the SVM base classifiers.
- **C4.5** [46] decision tree induces classification rules in the form of decision trees. The construction of the tree is carried out in a top-down manner. The normalized information gain (difference in entropy) is used to select the attribute that better splits the data in each node.
- ***k*NN** [1] *k-Nearest Neighbors* finds the *k* instances in the training set that are the closest to the test pattern. Then, the instance is labeled based on the number of examples from each class in this neighborhood. Both the distance and the number of neighbors are key factors of this algorithm.
- **Ripper** [11] *Repeated Incremental Pruning to Produce Error Reduction* induces a decision list of rules to label new instances. Each list of rules is grown one by one and immediately pruned. After completing a decision list for a given class, an optimization phase is carried out in the next stage.
- **PDFC** [9] *Positive Definite Fuzzy Classifier* extracts fuzzy rules from a SVM. Since the learning process minimizes an upper bound on the expected risk instead of the empirical risk, the classifier usually obtains a good generalization ability.

These learning algorithms were selected for the current (and the previous [21]) study due to their good behavior in a large number of real-world problems. Moreover, in case of SVM and PFDC there is not a multi-category approach established yet. Even though there exist several extensions [29] to deal with multiple classes, they have not shown real advantages to decomposition strategies that are commonly used in the SVM community for multi-class classification. Most of the combination methods for OVO classification make their predictions based on the confidence of the outputs of the base classifiers. We obtain the confidence for each classifier as follows:

- **SVM –** Probability estimates from the SVM [44].
- **C4.5 –** Accuracy of the leaf making the prediction (correctly classified train examples divided by the total number of covered train instances).
- ***k*NN –** Distance-based confidence estimation.

  $$Confidence = \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}} \tag{17}$$

  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.
- **Ripper –** Accuracy of the rule used in the prediction (computed as in C4.5 considering rules instead of leafs).
- **PDFC –** The prediction of the classifier, that is, confidence equal to 1 is given for the predicted class.

In some of the combination strategies ties might occur. As usual, in those cases the majority class is predicted. If the tie continues, the class is selected randomly.

The parameters used to train the base classifiers are shown in Table 1. These values are common for all problems, and they were selected according to the recommendation of the corresponding authors, which is also the default setting of the parameters included in KEEL[2] software [3,2] used to develop our experiments. Two configurations for SVMs are considered, 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). We treat nominal attributes in SVM and PDFC as scalars to fit the data into the systems using a polynomial kernel.

Even though 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 on 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. Moreover, when methods are not

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

tuned, winner methods tend to correspond to the most robust ones, which is also desirable.

4.2. Combinations considered

As we have already mentioned, we consider as representative combinations the same as those selected in [21] (recall that the same experimental framework is being used). There is a unique exception regarding SVMs. In [21], the best performer was Nesting OVO [36], but without significant differences with the rest of the methods. Although this strategy constructs several OVO ensembles recursively, it does not outperform other simpler approaches such as the probability estimates method by Wu et al. [54], which is much more extended. For this reason, we consider it to be the representative, and also in this manner, we are able to perform the comparison using exactly the same score-matrices in all combinations, so that the differences between the results are only due to the combinations themselves, which is a desirable characteristic to carry out their evaluation. We should stress that since the experimental set-up is maintained, those combinations that were found to be the best for each base classifier in [21] are also the best in the current one (without taking into account the new proposal); hence, we can reduce the experimental study to these aggregations instead of including all of them, which would make harder to follow the experimentation. The representative combinations for each classifier are the following:

- **SVM** – PE (Wu et al. Probability Estimates by Pairwise Coupling).
- **C4.5** – WV (Weighted Voting strategy).
- *k***NN** – ND (Non-Dominance criterion).
- **Ripper** – WV (Weighted Voting strategy).
- **PDFC** – PC (Probability Estimates by Pairwise Coupling).

In addition to these aggregations, we have also considered the approach based on DCS [22], which was able to outperform most of them in the same experimental framework as the one we are considering. In order to use this strategy, we have considered the same parameter value for k as that in the original paper, that is, $k = 3 \cdot m$ is considered as the neighborhood of the instance from which the probably competent classifiers are selected.

4.3. Data-sets

We have used the same nineteen data-sets from KEEL data-set repository[3] [2] that were considered in [21]. Tables 2 and 3 summarize their properties. In the former table 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. In the latter one, the number of instances from each class in each data-set is presented. 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 that more meaningful results are obtained from a multi-class classification point-of-view.

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

The performance estimates were obtained by means of a 5-fold stratified cross-validation (SCV). 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 test sets that will not contain any instance from some of the classes. More specifically, the data partitions were obtained by the Distribution Optimally Balanced SCV (DOB-SCV) [40,38], which aims to correct the data-set shift (when the training data and the test data do not follow the same distribution) that might be produced when dividing the data.

4.4. Performance measures and statistical tests

Different measures usually allow to observe different behaviors, which increases the strength of the empirical study in such way that more complete conclusions can be yielded from different (not opposite, yet complementary) deductions. There are two measures whose simplicity and successful application for both binary and multi-class problems have made them widely used. The accuracy rate and Cohen's kappa [10] measure. In the case of multi-class problems, only considering accuracy might not show the real behavior of the methods from a multi-class classification perspective. For this reason, we include kappa measure in the analysis, which 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. Kappa measure is computed as follows:

$$\kappa = \frac{n\sum_{i=1}^{m} h_{ii} - \sum_{i=1}^{m} T_{ri}T_{ci}}{n^2 - \sum_{i=1}^{m} T_{ri}T_{ci}}, \qquad (18)$$

where h_{ij} is the number of examples in the ith row jth column of the confusion matrix obtained from the predictions of the classifier and T_{ri}. 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) through 0 (random classification) to 1 (perfect agreement).

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. Accuracy rate scores all the successes over all

[3] 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

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.

In order to properly compare the performance of the classifiers, statistical analysis needs to be carried out. We consider the usage of non-parametric tests, according to the recommendations made in [12] and [24] (for more information see http://sci2s.ugr.es/sicidm/).

In the experimental study, we will carry out multiple comparisons, since we will first study the performance of the method with different values of k and then we will compare our method against the best combinations and the dynamic approach. On this account, we use Friedman aligned-ranks test [27] as a non-parametric statistical procedure to perform comparison among a set of algorithms. Then, if this test detects significant differences among them, we check if the control algorithm (the best one) is significantly better than the others (that is, $1 \times n$ comparison) using Holm post-hoc test [28]. Moreover, we consider the average aligned-ranks of each algorithm (used in the Friedman aligned-ranks test) in order to compare the behavior of each algorithm with respect to the others. These rankings are obtained computing the difference between the performance obtained by the algorithm and the mean performance of all algorithms in the corresponding data-set. These differences are ranked from 1 to $k \cdot n$ (being k the number of data-sets and n the number of methods), assigning the corresponding rank to the method from which the difference has been computed. Hence, the lower the rank is, the better the method is. At last, the average ranking of each algorithm in all data-sets can be computed to show their global performance.

Finally, we recall that we are comparing different combinations, and hence, we carry out the comparisons in each base classifiers independently, whereas the cross-comparison between base classifiers is out of the scope of this paper.

5. Experimental study

In this section, we will study the usefulness of our proposal. To do so, we first analyze the behavior of our approach with different values of k so as to check its robustness with respect to this value and to show its influence in the results obtained. Afterwards, we compare our DRCW-OVO with the best performer combinations from the state-of-the-art (including the DCS approach [22]). Our aim is to investigate whether the relative competence weighting of the classifiers is translated into an enhancement of the results, checking if the proposed weighting procedure is appropriate. Moreover, these comparisons will also show us whether the management of the non-competent classifiers can lead to enhance the results obtained, and hence, it would put out the importance of this problem in OVO strategy if such behavior is observed. Furthermore, we will conclude this section by analyzing the weights assigned to the outputs of the classifiers aiming at explaining why the method is working properly (and following the notions presented in Section 3.3).

According to these three objectives, we have therefore divided this Section into three Subsections:

- We study the different values of k in Subsection 5.1.
- We compare the proposed approach against the state-of-the-art combinations in Subsection 5.2.
- We aim to explain the relative weighting mechanism of DRCW-OVO in Subsection 5.3.

5.1. *Analyzing the influence of the value of k in DRCW-OVO*

In this Section we want to investigate how the different values of k may affect the behavior of the proposed method. Recall that we measure the competence of each output of a base classifier depending on how far the instance is from the class of the corresponding output. In order to do so, we have considered several values for $k = 1, 3, 5$ and 10 neighbors. Notice that as we have mentioned, this value is not exactly the same as the number of neighbors used in k NN classification, since $k \cdot m$ neighbors are used in this case to compute the weights (k instances from each class). In addition, we have also considered another configuration denoted as $k = -1$, in which all the instances of each class are taken as neighbors for the computation of the weights, that is, the distance from the instance to the class is computed as the average distance to all the examples of that class. This configuration may avoid noise and can be seen as using the distance to the class centroid without carrying out a clustering model (which should be studied as a future research line). However, taking such a general distance could also produce a loss of discrimination power.

The results obtained for the five configurations considered in each base classifier are shown in Tables 4 and 5, with accuracy and kappa performance measures, respectively. The best result within each base classifier and data-set is stressed in bold-face. One can observe that the results with $k=1,3,5$ are stable, whereas with 10 neighbors the results are slightly inferior and with $k=-1$ the lowest results for all the base classifiers are obtained. However, we cannot obtain meaningful conclusions without carrying out the proper statistical analysis. Hence, we have performed several Friedman aligned-ranks tests in order to compare the different configurations of k in each base classifier, whose results are shown in Tables 6 and 7 (accuracy and kappa, respectively). In each table, a test is executed for each base classifier, that is, each test compares the different configurations ($k=1,3,5,10,-1$) in the same column (base classifier). In these tables, the aligned-ranks obtained by each configuration are shown (the lower the better). Near the ranks obtained by each method the p-value obtained by the Holm post-hoc test is shown in brackets, which compares a control method (the best one, i.e., the one with the lowest rank in a column) against the rest. A '+' close to the p-value means that statistical differences are found in the comparison with $\alpha=0.1$ (90% confidence) and a '*' with $\alpha=0.05$ (95% confidence).

Looking at Tables 6 and 7, it can be observed that only statistical differences are found with respect to the case where all the instances of a class are taken as neighbors. Hence, we can conclude that the method is robust with respect to the value of k. Furthermore, high (close to 1) p-values are obtained when $k=1,3,5$ are considered (and also with k=10 in most of the cases). Therefore, this value does not highly affect the behavior of DRCW-OVO as long as it is not set too high, since in such an extreme case, the discriminant power is lost.

Notice that increasing the neighborhood the distances from the instance to the different classes become more similar; as a result, the new mechanism lose its importance because the relative weighting has less influence in the final decision. Overall, similar results are obtained regardless of this parameter (neglecting $k=-1$). On account of these experimental results, for the remainder of this section we set the value of k=5, because it is the one achieving the lowest ranks the most times (considering the tests with accuracy and kappa).

5.2. DRCW-OVO vs. state-of-the-art combinations in OVO strategy

As we have explained, we analyze the results of the state-of-the-art combinations (including the best performer combinations in [21] and DCS method in [22]) and DRCW-OVO both in terms of accuracy and kappa. Table 8 shows the test accuracy results of the different combinations (the best performer, DCS and DRCW-OVO with k=5, shown as DRCW) for each base classifier. The best result within each base classifier and data-set is stressed in bold-face. Similarly, Table 9 presents the same results in terms of kappa measure.

From Tables 8 and 9, we can observe that our proposal performs much better than the state-of-the-art combinations in all the base classifiers studied and in both performance measures. Both the average performance values and the number of data-sets in which DRCW-OVO are remarkable. Nevertheless, in this case we can neither extract any meaningful conclusion without supporting our statements on the appropriate statistical analysis.

Hence, we have followed the same methodology as that in the previous section; we have also used Friedman aligned-rank test to compare the three methods in each base classifier. The results of these tests are shown in Tables 10 (accuracy) and 11 (kappa). Recall that a test is carried out in each column and the aligned-ranks for each method are presented together with the p-value obtained with the Holm post-hoc test comparing the best one (the one with the lowest ranks) against the rest. A '+' close to the p-value means that statistical differences are found in the

Table 4
Average accuracy results in test for the different values of k in DRCW-OVO (with $k=-1$ all the instances of the class are used to compute the distance).

Data-set	C45					SVM$_{Poly}$					SVM$_{Puk}$					3NN					PDFC					Ripper				
	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1
Autos	**85.90**	81.55	80.96	74.40	75.01	**83.91**	80.15	79.48	74.36	73.07	**79.60**	73.39	71.45	70.84	71.45	**79.40**	77.02	75.14	72.44	71.37	81.28	**81.33**	80.74	77.42	76.24	**89.38**	85.15	84.58	81.94	81.26
Car	96.41	**97.16**	96.99	96.88	90.33	96.76	**97.22**	97.16	97.05	90.62	**82.47**	82.06	81.65	80.50	64.64	96.41	96.82	96.93	**97.11**	89.76	98.49	99.31	99.42	**99.71**	96.99	96.47	**96.59**	96.35	96.18	91.03
Cleveland	54.22	54.88	55.23	**55.24**	52.57	58.31	58.66	58.66	**59.00**	58.96	47.87	48.21	48.88	**48.89**	47.87	56.95	57.28	56.61	56.93	**57.28**	54.93	56.27	56.61	**57.63**	55.95	55.57	55.20	**56.90**	56.58	54.53
Dermatology	97.49	**98.06**	**98.06**	**98.06**	97.78	**96.11**	95.83	95.55	95.27	94.99	**97.76**	97.48	97.48	97.48	97.20	96.35	96.62	96.90	96.90	**96.91**	**92.47**	91.90	91.90	91.90	91.63	95.28	**95.55**	95.27	95.00	95.54
Ecoli	84.34	84.41	**85.58**	84.70	83.18	81.94	**82.53**	82.25	81.97	78.96	78.34	80.72	81.64	**81.95**	80.76	80.73	82.53	**84.30**	83.73	82.54	83.10	84.37	**84.68**	84.38	83.79	81.12	81.44	82.34	**83.20**	80.53
Flare	74.21	74.53	**75.27**	74.83	75.13	75.39	**76.01**	75.86	75.96	75.43	71.91	72.14	72.04	72.36	**73.82**	70.87	72.30	72.43	72.73	**75.21**	73.28	73.69	73.69	74.09	**75.13**	75.01	**75.90**	75.60	75.53	75.00
Glass	**77.46**	75.71	74.81	74.84	71.52	72.36	**72.36**	71.04	68.81	65.93	76.56	**77.55**	76.19	74.76	74.25	**74.78**	73.42	74.33	73.92	72.42	**73.32**	70.57	70.12	69.62	67.81	75.76	**76.32**	75.40	71.75	69.75
Led7digit	64.79	64.84	65.33	67.24	**69.32**	67.88	66.47	66.47	**70.22**	69.99	61.45	61.43	62.54	63.64	**67.77**	67.42	67.63	68.26	**70.41**	69.30	64.40	65.16	65.42	63.90	**66.98**	64.56	64.68	64.19	**66.97**	65.84
Lymphography	75.81	**76.44**	**76.44**	75.81	75.88	83.68	83.10	83.10	83.77	**84.43**	**83.16**	82.50	82.50	82.50	82.50	**81.46**	79.52	79.52	78.83	78.90	**83.19**	**83.19**	**83.19**	**83.19**	**83.19**	**77.73**	77.11	77.04	76.33	77.02
Nursery	**92.67**	91.59	90.90	90.20	90.90	94.37	**94.53**	94.53	94.06	90.90	**92.37**	90.90	90.83	89.67	84.26	**94.06**	93.68	93.68	93.75	93.98	97.61	97.76	97.84	97.92	**98.07**	**93.44**	93.06	92.44	91.97	90.59
Pageblocks	95.82	**96.37**	95.82	95.63	95.44	**95.64**	95.63	95.27	95.09	94.72	94.93	95.11	95.11	**95.45**	94.92	94.37	95.27	95.09	95.27	**95.64**	**95.28**	95.09	95.09	95.09	95.09	95.82	95.82	96.00	96.00	**96.36**
Penbased	**96.28**	**96.28**	95.64	94.73	89.10	**97.83**	97.37	97.01	96.74	89.19	97.82	**98.00**	98.00	97.82	89.74	**97.09**	**97.09**	96.91	95.91	89.01	98.10	**98.28**	98.10	97.92	89.74	**96.55**	96.46	96.01	95.56	89.10
Satimage	85.56	85.10	85.41	**86.04**	85.10	**88.20**	87.74	86.34	85.42	83.70	86.78	86.79	87.56	**88.03**	87.10	**89.43**	87.88	88.34	88.66	87.58	**87.41**	87.26	87.25	87.26	86.02	85.70	86.01	**86.01**	85.87	83.54
Segment	**98.14**	98.10	97.97	97.75	96.71	**96.36**	96.02	95.58	95.02	92.34	**97.45**	97.36	97.40	97.36	97.14	**97.19**	96.97	96.84	96.93	96.02	**97.53**	97.36	97.27	97.23	96.88	97.88	**97.92**	97.84	97.66	96.80
Shuttle	**99.77**	99.68	99.72	99.72	99.68	**99.59**	99.54	99.50	99.31	95.40	**99.68**	**99.68**	99.63	99.63	98.62	99.50	99.45	99.40	99.40	**99.50**	**99.17**	98.94	98.76	98.49	96.46	**99.63**	99.54	99.54	99.54	98.89
Vehicle	**74.11**	73.52	73.88	73.87	73.05	73.88	74.00	74.48	**74.59**	73.52	81.80	81.80	**82.04**	81.92	81.92	**73.18**	72.35	72.23	72.47	72.11	84.41	**84.53**	84.41	84.41	84.29	**71.87**	71.76	71.29	70.81	70.34
Vowel	**96.36**	95.45	94.75	92.63	85.15	**97.78**	96.36	95.05	90.30	73.74	99.60	99.39	99.29	99.39	**99.70**	**98.89**	97.68	97.27	96.97	97.17	**99.29**	98.89	98.59	98.28	98.28	**97.17**	95.66	94.44	92.42	82.12
Yeast	59.24	60.58	60.46	**61.13**	60.19	59.17	60.45	60.92	**62.07**	59.85	58.56	61.26	62.14	**62.54**	62.01	56.61	58.16	58.30	59.11	**59.85**	60.05	60.86	**60.92**	60.32	60.59	60.05	61.33	**61.81**	61.60	60.13
Zoo	92.17	92.17	93.22	**94.05**	**94.05**	95.72	95.72	**96.77**	**96.77**	**96.77**	**91.76**	90.80	90.80	89.85	90.80	**95.69**	94.64	94.64	94.64	**95.69**	96.77	95.72	**97.82**	**97.82**	**97.82**	95.05	94.00	**96.10**	**96.10**	**96.10**
Average	**84.25**	84.02	84.02	83.57	82.11	**84.99**	84.72	84.48	83.99	81.18	**83.15**	82.98	83.01	82.87	81.39	**84.23**	84.02	84.06	84.01	83.17	85.27	85.29	**85.36**	85.08	84.26	**84.42**	84.18	84.17	83.74	81.81

Table 5
Average kappa results in test for the different values of k in DRCW-OVO (with $k=-1$ all the instances of the class are used to compute the distance).

Data-set	C45					SVM$_{Poly}$					SVM$_{Puk}$				
	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1
Autos	**0.8141**	0.7570	0.7495	0.6606	0.6700	**0.7898**	0.7398	0.7307	0.6629	0.6496	**0.7336**	0.6529	0.6266	0.6187	0.6272
Car	0.9224	**0.9392**	0.9355	0.9332	0.8067	0.9293	**0.9399**	0.9388	0.9363	0.8124	**0.6783**	0.6718	0.6656	0.6472	0.4294
Cleveland	0.2609	0.2653	**0.2673**	0.2672	0.2451	0.3179	0.3215	0.3227	0.3276	**0.3348**	0.2674	0.2702	**0.2752**	0.2721	0.2664
Dermatology	0.9685	**0.9756**	**0.9756**	**0.9756**	0.9721	**0.9514**	0.9480	0.9444	0.9409	0.9374	**0.9719**	0.9683	0.9683	0.9683	0.9649
Ecoli	0.7830	0.7844	**0.7997**	0.7864	0.7701	0.7475	**0.7561**	0.7516	0.7467	0.7081	0.7067	0.7368	0.7492	**0.7536**	0.7400
Flare	0.6629	0.6663	**0.6759**	0.6700	0.6745	0.6789	**0.6866**	0.6844	0.6857	0.6798	0.6373	0.6402	0.6387	0.6431	**0.6628**
Glass	**0.6892**	0.6633	0.6495	0.6467	0.6117	**0.6183**	0.6171	0.5986	0.5621	0.5301	0.6823	**0.6942**	0.6755	0.6548	0.6510
Led7digit	0.6062	0.6071	0.6124	0.6338	**0.6575**	0.6411	0.6254	0.6254	**0.6674**	0.6650	0.5698	0.5694	0.5817	0.5942	**0.6405**
Lymphography	0.5368	**0.5462**	**0.5462**	0.5368	0.5413	0.6824	0.6732	0.6732	0.6869	**0.6996**	**0.6721**	0.6581	0.6581	0.6581	0.6581
Nursery	**0.8919**	0.8758	0.8653	0.8549	0.8668	0.9175	**0.9198**	0.9198	0.9130	0.8677	**0.8894**	0.8688	0.8678	0.8515	0.7811
Pageblocks	0.7770	**0.8036**	0.7718	0.7552	0.7465	**0.7517**	0.7459	0.7242	0.6988	0.6823	0.7483	**0.7483**	0.7400	0.7365	0.7211
Penbased	**0.9586**	0.9586	0.9515	0.9415	0.8789	**0.9758**	0.9708	0.9668	0.9638	0.8799	0.9758	**0.9778**	0.9778	0.9758	0.8859
Satimage	0.8216	0.8160	0.8198	**0.8274**	0.8162	**0.8537**	0.8481	0.8308	0.8193	0.7985	0.8383	0.8381	0.8474	**0.8532**	0.8420
Segment	**0.9783**	0.9778	0.9763	0.9737	0.9616	**0.9576**	0.9535	0.9485	0.9419	0.9106	**0.9702**	0.9692	0.9697	0.9692	0.9667
Shuttle	**0.9936**	0.9910	0.9923	0.9923	0.9910	**0.9884**	0.9871	0.9859	0.9809	0.8683	**0.9911**	0.9911	0.9898	0.9898	0.9618
Vehicle	**0.6548**	0.6468	0.6517	0.6516	0.6410	0.6518	0.6535	0.6597	**0.6612**	0.6470	0.7572	0.7572	**0.7604**	0.7588	0.7588
Vowel	**0.9600**	0.9500	0.9422	0.9189	0.8367	**0.9756**	0.9600	0.9456	0.8933	0.7111	0.9956	0.9933	0.9922	0.9933	**0.9967**
Yeast	0.4693	0.4863	0.4848	**0.4935**	0.4838	0.4674	0.4830	0.4889	**0.5038**	0.4794	0.4671	0.4990	0.5089	**0.5137**	0.5119
Zoo	0.8966	0.8966	0.9106	**0.9212**	**0.9212**	0.9419	0.9419	**0.9561**	**0.9561**	**0.9561**	**0.8869**	0.8753	0.8753	0.8622	0.8753
Average	**0.7708**	0.7688	0.7672	0.7600	0.7417	**0.7809**	0.7774	0.7735	0.7657	0.7272	**0.7600**	0.7569	0.7562	0.7534	0.7338
Data-set	3NN					PDFC					Ripper				
	1	3	5	10	−1	1	3	5	10	−1	1	3	5	10	−1
Autos	**0.7338**	0.7020	0.6775	0.6411	0.6317	0.7546	**0.7548**	0.7475	0.7035	0.6898	**0.8612**	0.8054	0.7988	0.7642	0.7564
Car	0.9219	0.9312	0.9339	**0.9378**	0.7942	0.9674	0.9849	0.9874	**0.9937**	0.9363	0.9238	**0.9267**	0.9217	0.9179	0.8170
Cleveland	0.3070	0.3028	0.2902	0.2885	**0.3097**	0.2868	0.3037	0.3088	**0.3205**	0.3085	0.2986	0.2945	**0.3137**	0.3047	0.3010
Dermatology	0.9539	0.9573	0.9609	0.9609	**0.9611**	**0.9039**	0.8964	0.8964	0.8964	0.8929	0.9407	**0.9442**	0.9407	0.9373	0.9441
Ecoli	0.7326	0.7571	**0.7817**	0.7732	0.7586	0.7656	0.7836	**0.7878**	0.7834	0.7757	0.7418	0.7463	0.7577	**0.7691**	0.7362
Flare	0.6219	0.6387	0.6403	0.6439	**0.6786**	0.6538	0.6588	0.6588	0.6643	**0.6779**	0.6758	**0.6866**	0.6826	0.6817	0.6764
Glass	**0.6519**	0.6308	0.6442	0.6331	0.6197	**0.6258**	0.5871	0.5809	0.5712	0.5519	0.6690	**0.6748**	0.6610	0.6055	0.5943
Led7digit	0.6363	0.6387	0.6458	**0.6699**	0.6578	0.6024	0.6110	0.6139	0.5970	**0.6314**	0.6046	0.6060	0.6003	**0.6311**	0.6189
Lymphography	**0.6442**	0.6053	0.6053	0.5911	0.6109	**0.6686**	**0.6686**	**0.6686**	**0.6686**	**0.6686**	**0.5791**	0.5670	0.5672	0.5539	0.5646
Nursery	**0.9130**	0.9073	0.9073	0.9084	0.9124	0.9649	0.9672	0.9683	0.9695	**0.9718**	**0.9039**	0.8984	0.8894	0.8825	0.8629
Pageblocks	0.6857	0.7319	0.7210	0.7158	**0.7396**	**0.7217**	0.7116	0.7116	0.7116	0.7119	0.7842	0.7791	0.7868	0.7749	**0.8048**
Penbased	**0.9676**	0.9676	0.9656	0.9545	0.8778	0.9789	**0.9809**	0.9789	0.9768	0.8860	**0.9617**	0.9607	0.9556	0.9506	0.8789
Satimage	**0.8690**	0.8498	0.8554	0.8592	0.8463	**0.8442**	0.8423	0.8422	0.8422	0.8271	0.8239	0.8278	**0.8278**	0.8258	0.7977
Segment	**0.9672**	0.9646	0.9631	0.9641	0.9535	**0.9712**	0.9692	0.9682	0.9677	0.9636	0.9753	**0.9758**	0.9748	0.9727	0.9626
Shuttle	0.9859	0.9846	0.9833	0.9833	**0.9859**	**0.9769**	0.9703	0.9652	0.9578	0.9007	**0.9898**	0.9872	0.9872	0.9872	0.9690
Vehicle	**0.6423**	0.6314	0.6297	0.6329	0.6283	0.7921	**0.7936**	0.7920	0.7920	0.7905	**0.6250**	0.6235	0.6173	0.6110	0.6049
Vowel	**0.9878**	0.9744	0.9700	0.9667	0.9689	**0.9922**	0.9878	0.9844	0.9811	0.9811	**0.9689**	0.9522	0.9389	0.9167	0.8033
Yeast	0.4356	0.4551	0.4565	0.4656	**0.4801**	0.4785	0.4879	**0.4888**	0.4808	0.4855	0.4850	0.5006	**0.5069**	0.5038	0.4890
Zoo	0.9411	0.9262	0.9262	0.9260	**0.9412**	0.9563	0.9423	**0.9703**	**0.9703**	**0.9703**	0.9340	0.9200	**0.9481**	**0.9481**	**0.9481**
Average	**0.7684**	0.7662	0.7662	0.7640	0.7556	0.7845	0.7843	**0.7853**	0.7815	0.7695	**0.7761**	0.7725	0.7724	0.7652	0.7437

Table 6
Friedman aligned-rank tests comparing the different values of k in each base classifier with accuracy. A '+' near the p-value means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence).

k values	C45	SVM$_{Poly}$	SVM$_{Puk}$	3NN	PDFC	Ripper
1	45.26 (1.00000)	38.87 (1.00000)	45.39 (1.00000)	41.26	47.24 (0.82509)	41.32 (1.00000)
3	40.63 (1.00000)	37.89	47.66 (1.00000)	50.55 (0.89698)	41.79 (0.82509)	40.26 (1.00000)
5	38.76	40.82 (1.00000)	42.18	48.34 (0.89698)	38.74	37.74
10	45.34 (1.00000)	48.47 (0.71071)	43.18 (1.00000)	44.18 (0.89698)	48.50 (0.82509)	47.42 (0.83679)
−1	70.00 (0.00191*)	73.95 (0.00022*)	61.58 (0.12051)	55.66 (0.43013)	63.74 (0.02075*)	73.26 (0.00029*)

Table 7
Friedman aligned-rank tests comparing the different values of k in each base classifier with kappa. A '+' near the p-value means that there are statistical differences with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence).

k values	C45	SVM$_{Poly}$	SVM$_{Puk}$	3NN	PDFC	Ripper
1	42.95 (1.00000)	37.29	42.32	40.63	46.84 (0.92603)	40.21 (1.00000)
3	40.11 (1.00000)	37.47 (1.00000)	46.05 (1.00000)	49.89 (0.93831)	42.39 (0.92603)	40.21 (1.00000)
5	39.79	41.82 (1.00000)	42.55 (1.00000)	48.66 (0.93831)	39.79	37.58
10	48.76 (0.94717)	50.47 (0.42141)	47.39 (1.00000)	49.55 (0.93831)	48.89 (0.92603)	50.74 (0.42379)
−1	68.39 (0.00553*)	72.95 (0.00027*)	61.68 (0.12141)	51.26 (0.93831)	62.08 (0.05080+)	71.26 (0.00066*)

Table 8
Average accuracy results in test of the representative combinations, DCS method and DRCW-OVO method (with $k = 5$) for each base classifier.

Data-set	C45			SVM$_{Poly}$			SVM$_{Puk}$			3NN			PDFC			Ripper		
	WV	DCS	DRCW	PE	DCS	DRCW	PE	DCS	DRCW	ND	DCS	DRCW	PC	DCS	DRCW	WV	DCS	DRCW
Autos	76.24	74.96	**80.96**	73.75	73.81	**79.48**	69.02	70.27	**71.45**	**78.88**	76.96	75.14	78.82	79.40	**80.74**	**85.09**	84.42	84.58
Car	94.68	94.50	**96.99**	93.58	93.58	**97.16**	64.99	**84.84**	81.65	93.57	93.40	**96.93**	99.77	**99.88**	99.42	92.59	93.52	**96.35**
Cleveland	52.55	53.55	**55.23**	58.97	**59.31**	58.66	47.53	47.87	**48.88**	**58.31**	57.96	56.61	53.92	55.93	**56.61**	52.18	54.54	**56.90**
Dermatology	95.24	**98.32**	98.06	94.71	94.99	**95.55**	97.20	97.20	**97.48**	92.14	95.49	**96.90**	84.66	**93.85**	91.90	93.32	94.43	**95.27**
Ecoli	81.06	81.94	**85.58**	79.37	79.63	**82.25**	77.11	77.11	**81.64**	81.66	82.52	**84.30**	84.07	83.78	**84.68**	78.47	78.74	**82.34**
Flare	**75.34**	73.62	75.27	75.43	75.46	**75.86**	69.28	**73.39**	72.04	71.21	71.59	**72.43**	73.64	**73.92**	73.69	75.24	74.83	**75.60**
Glass	72.03	71.63	**74.81**	62.14	63.14	**71.04**	73.72	74.15	**76.19**	73.35	74.27	**74.33**	68.72	**70.12**	**70.12**	68.56	68.12	**75.40**
Led7digit	64.51	**65.35**	65.33	67.90	**68.09**	66.47	61.33	61.57	**62.54**	66.68	67.88	**68.26**	62.17	62.60	**65.42**	63.98	63.86	**64.19**
Lymphography	74.50	**76.44**	**76.44**	82.48	82.48	**83.10**	81.87	81.87	**82.50**	68.19	**79.55**	79.52	**83.19**	**83.19**	**83.19**	75.68	75.68	**77.04**
Nursery	89.66	89.81	**90.90**	92.13	92.13	**94.53**	80.33	89.05	**90.83**	93.29	93.29	**93.68**	**97.92**	**97.92**	97.84	90.66	90.81	**92.44**
Pageblocks	95.64	95.46	**95.82**	94.90	94.53	**95.27**	94.58	94.76	**95.11**	**95.63**	95.46	95.09	**95.09**	94.91	**95.09**	95.45	95.11	**96.00**
Penbased	91.10	91.11	**95.64**	95.92	96.01	**97.01**	97.55	97.64	**98.00**	**97.00**	96.64	96.91	**98.19**	98.10	98.10	91.38	91.11	**96.01**
Satimage	82.15	82.92	**85.41**	84.48	84.16	**86.34**	84.77	85.70	**87.56**	87.58	87.73	**88.34**	86.79	86.95	**87.25**	82.61	82.14	**86.01**
Segment	96.28	96.71	**97.97**	92.68	92.90	**95.58**	97.23	97.36	**97.40**	96.58	96.80	**96.84**	97.32	**97.36**	97.27	96.58	96.88	**97.84**
Shuttle	99.59	99.68	**99.72**	96.55	97.61	**99.50**	99.59	99.63	**99.63**	**99.50**	99.40	99.40	97.43	98.03	**98.76**	99.40	**99.68**	99.54
Vehicle	72.33	72.81	**73.88**	73.53	74.00	**74.48**	81.92	81.92	**82.04**	72.11	72.23	**72.23**	**84.53**	84.40	84.41	69.27	70.20	**71.29**
Vowel	83.43	83.64	**94.75**	71.41	71.82	**95.05**	**99.70**	**99.70**	99.29	**97.78**	97.37	97.27	98.28	98.08	**98.59**	80.20	79.39	**94.44**
Yeast	59.57	59.84	**60.46**	60.52	59.98	**60.92**	59.31	59.51	**62.14**	56.68	56.54	**58.30**	60.25	59.98	**60.92**	58.30	58.10	**61.81**
Zoo	92.17	92.17	**93.22**	95.72	95.72	**96.77**	78.06	84.13	**90.80**	89.90	91.86	**94.64**	96.77	96.77	**97.82**	94.05	94.05	**96.10**
Average	81.48	81.81	**84.02**	81.38	81.55	**84.48**	79.74	81.98	**83.01**	82.63	83.52	**84.06**	84.29	85.01	**85.36**	81.21	81.35	**84.17**

comparison with $\alpha = 0.1$ (90% confidence) and a '*' with $\alpha = 0.05$ (95% confidence).

The results of the statistical analysis put out the superiority of DRCW-OVO against the state-of-the-art combinations for OVO strategy, also outperforming the DCS method which, similar to DRCW-OVO, considers information from the neighborhood of the instance to remove the non-competent classifiers. Statistically significant differences are found in all base classifiers in favor of DRCW-OVO both with accuracy and kappa measure.

The relative weighting helps in improving the classification in OVO strategy taking into account the competence of each classifier in the classes it must distinguish. More importantly, the proposed combination is robust, achieving a great performance in all the base classifiers considered. In addition, these results show that there was margin for improvement when considering the problem of non-competent classifiers, showing the importance of this problem and improving the previous approach [22] with similar characteristics.

5.3. Explaining the relative weighting mechanism in DRCW-OVO

The aim of this section is to complement Section 3.3, where we have shown an illustrative example to explain the behavior of the proposed method. We have also shown that in order to work properly, the weights assigned to each output should depend on whether the classifier is competent or not. That is, outputs from a competent classifier should be weighted differently by assigning a high value to the correct class (which should be nearer to the instance than the other one), whereas giving a low value to the other one. Otherwise, the outputs of the non-competent classifiers should be weighted similarly, assigning a value as similar as possible to 0.5 (not empowering these outputs).

Table 9
Average kappa results in test of the representative combinations, DCS method and DRCW-OVO method (with $k=5$) for each base classifier.

Data-set	C45			SVM_{Poly}			SVM_{Puk}			3NN			PDFC			Ripper		
	WV	DCS	DRCW	PE	DCS	DRCW	PE	DCS	DRCW	ND	DCS	DRCW	PC	DCS	DRCW	WV	DCS	DRCW
Autos	0.6857	0.6676	**0.7495**	0.6575	0.6584	**0.7307**	0.5974	0.6144	**0.6266**	**0.7256**	0.6970	0.6775	0.7231	0.7299	**0.7475**	**0.8053**	0.7962	0.7988
Car	0.8845	0.8804	**0.9355**	0.8578	0.8577	**0.9388**	0.4342	**0.7053**	0.6656	0.8607	0.8562	**0.9339**	0.9950	**0.9975**	0.9874	0.8430	0.8622	**0.9217**
Cleveland	0.2431	0.2524	**0.2673**	0.3342	**0.3399**	0.3227	0.2640	0.2662	**0.2752**	**0.3282**	0.3179	0.2902	0.2818	0.2888	**0.3088**	0.2637	0.2951	**0.3137**
Dermatology	0.9402	**0.9790**	0.9756	0.9339	0.9374	**0.9444**	0.9649	0.9649	**0.9683**	0.8997	0.9429	**0.9609**	0.8011	**0.9219**	0.8964	0.9161	0.9301	**0.9407**
Ecoli	0.7405	0.7517	**0.7997**	0.7132	0.7158	**0.7516**	0.6911	0.6911	**0.7492**	0.7445	0.7563	**0.7817**	0.7801	0.7751	**0.7878**	0.7095	0.7115	**0.7577**
Flare	**0.6765**	0.6547	0.6759	0.6785	0.6791	**0.6844**	0.6055	**0.6549**	0.6387	0.6253	0.6295	**0.6403**	0.6594	**0.6628**	0.6588	0.6794	0.6742	**0.6826**
Glass	0.6195	0.6117	**0.6495**	0.4667	0.4826	**0.5986**	0.6417	0.6491	**0.6755**	0.6326	**0.6445**	0.6442	0.5645	0.5797	**0.5809**	0.5736	0.5683	**0.6610**
Led7digit	0.6032	**0.6127**	0.6124	0.6414	**0.6436**	0.6254	0.5685	0.5710	**0.5817**	0.6282	0.6414	**0.6458**	0.5775	0.5822	**0.6139**	0.5980	0.5967	**0.6003**
Lymphography	0.5167	0.5447	**0.5462**	0.6606	0.6606	**0.6732**	0.6469	0.6469	**0.6581**	0.5516	0.6044	**0.6053**	**0.6686**	**0.6686**	**0.6686**	0.5387	0.5400	**0.5672**
Nursery	0.8468	0.8491	**0.8653**	0.8847	0.8847	**0.9198**	0.7307	0.8441	**0.8678**	0.9018	0.9018	**0.9073**	**0.9695**	**0.9695**	0.9683	0.8634	0.8657	**0.8894**
Pageblocks	0.7711	0.7576	**0.7718**	0.7036	0.6734	**0.7242**	0.7337	0.7400	**0.7400**	**0.7556**	0.7436	0.7210	**0.7117**	0.6929	0.7116	0.7739	0.7532	**0.7868**
Penbased	0.9011	0.9012	**0.9515**	0.9547	0.9557	**0.9668**	0.9728	0.9738	**0.9778**	**0.9667**	0.9626	0.9656	**0.9799**	0.9789	0.9789	0.9042	0.9012	**0.9556**
Satimage	0.7796	0.7896	**0.8198**	0.8077	0.8037	**0.8308**	0.8137	0.8253	**0.8474**	0.8460	0.8480	**0.8554**	0.8364	0.8383	**0.8422**	0.7863	0.7810	**0.8278**
Segment	0.9566	0.9616	**0.9763**	0.9147	0.9172	**0.9485**	0.9677	0.9692	**0.9697**	0.9601	0.9626	**0.9631**	0.9687	**0.9692**	0.9682	0.9601	0.9636	**0.9748**
Shuttle	0.9885	0.9910	**0.9923**	0.9016	0.9313	**0.9859**	0.9886	**0.9898**	0.9898	**0.9859**	0.9833	0.9833	0.9281	0.9441	**0.9652**	0.9834	**0.9910**	0.9872
Vehicle	0.6309	0.6372	**0.6517**	0.6470	0.6534	**0.6597**	0.7588	0.7588	**0.7604**	0.6281	0.6297	**0.6297**	**0.7936**	0.7920	0.7920	0.5903	0.6028	**0.6173**
Vowel	0.8178	0.8200	**0.9422**	0.6856	0.6900	**0.9456**	**0.9967**	**0.9967**	0.9922	**0.9756**	0.9711	0.9700	0.9811	0.9789	**0.9844**	0.7822	0.7733	**0.9389**
Yeast	0.4724	0.4760	**0.4848**	0.4850	0.4779	**0.4889**	0.4764	0.4811	**0.5089**	0.4346	0.4325	**0.4565**	0.4798	0.4753	**0.4888**	0.4645	0.4631	**0.5069**
Zoo	0.8970	0.8966	**0.9106**	0.9418	0.9420	**0.9561**	0.7036	0.7848	**0.8753**	0.8597	0.8906	**0.9262**	0.9563	0.9563	**0.9703**	0.9209	0.9209	**0.9481**
Average	0.7353	0.7387	**0.7672**	0.7300	0.7318	**0.7735**	0.7135	0.7436	**0.7562**	0.7532	0.7588	**0.7662**	0.7714	0.7790	**0.7853**	0.7346	0.7363	**0.7724**

On this account, we want to analyze the weights assigned by DRCW-OVO so as to explain that the weighting mechanism is suitable. With this objective, we have computed three values for each one of the configurations of k tested in Section 5.1 and for each data-set, which are shown in Table 12:

1. The average weight assigned to the output of the correct class in a competent classifier (the one that should be predicted), denoted as Wc.
2. The average weight assigned to the class that should not be predicted in a competent classifier, denoted as WnC (hence, Wc+WnC=1).
3. The average standard deviation of the weights assigned to the non-competent classifiers in each score-matrix with respect to 0.5 (denoted as D0.5), which would be the ideal weight. Therefore, the lower the average deviation is, the better the non-competent classifiers modeled are, since it means that the weights are close to 0.5, and hence the outputs that should not be predicted are not empowered.

From Table 12, we can observe that the methods achieving the best results in Section 5.1 ($k=1,3,5$) are also the ones obtaining the highest average weight values for the real class (and consequently the lowest for the other one). Moreover, we should stress that there must also be a balance between the deviation (the lower the better) and the weights Wc. For example, with $k=-1$ we obtain a very low deviation, but it also produces too similar Wc and WnC values, which makes the influence of the proposed method low, as we have already shown. It should be noticed that this is due to the fact that more similar distances between the instance and the different classes are found as k increases, loosing the capability of giving different weights to Wc and WnC. As a result, the influence of the proposed methodology in OVO with too large k values becomes low.

6. Concluding remarks

In this paper, we have presented a DCW procedure to deal with the non-competent classifiers problem in OVO strategy, the DRCW-OVO method. To do so, we have proposed to use the nearest neighbor of each class from the instance to be classified in order to dynamically weight the outputs of the classifiers by these distances. The novelty of this approach resides both in the management of the non-competent classifiers and in the application of DCW to OVO strategy, since DCW techniques have not been previously used for decomposition-based techniques.

The new combination has shown its validity and usefulness in a number of real-world problems, being a simple strategy able to outperform the state-of-the-art methods in exchange for a low increase of the computational cost. The robustness of the method with respect to the value of the parameter k (the number of neighbors from each class used) has been analyzed. We have also explained the reason why the method is working, which is due to the greater relative weights assigned to the outputs of the correct class, whereas those of the competent classifiers are equally weighted, penalizing those classes. Furthermore, we must stress that all the differences found in this paper are due to the combinations studied and not due to differences in the base classifiers, since all the combinations base their decision on the same score-matrices.

This paper opens up new possibilities to adapt DCW, DCS and DES techniques to decomposition-based strategies, which could enhance the results obtained by classical combination methods. On this account, as a future work, we intend to introduce these strategies in the more general ECOC framework. Furthermore, it will be interesting to study different models to weight the competence such as the usage of clustering models to compute the distance to the clusters instead of using the k nearest neighbors approach proposed.

Table 10
Friedman aligned-rank tests comparing the representative combinations (Repr.), DCS and DRCW-OVO (with $k=5$) with accuracy. A '+' near the p-value means that there are statistical differences with $\alpha=0.1$ (90% confidence) and a '*' with $\alpha=0.05$ (95% confidence).

Method	C45	SVM$_{Poly}$	SVM$_{Puk}$	3NN	PDFC	Ripper
Repr.	41.08 (0.00000*)	38.26 (0.00000*)	42.97 (0.00000*)	36.92 (0.00389*)	37.87 (0.00050*)	39.58 (0.00000*)
DCS	34.63 (0.00001*)	36.11 (0.00001*)	30.03 (0.00292*)	29.84 (0.07448+)	30.97 (0.01732*)	36.68 (0.00000*)
DRCW	11.29	12.63	14.00	20.24	18.16	10.74

Table 11
Friedman aligned-rank tests comparing the representative combinations (Repr.), DCS and DRCW-OVO (with $k=5$) with kappa. A '+' near the p-value means that there are statistical differences with $\alpha=0.1$ (90% confidence) and a '*' with $\alpha=0.05$ (95% confidence).

Method	C45	SVM$_{Poly}$	SVM$_{Puk}$	3NN	PDFC	Ripper
Repr.	40.42 (0.00000*)	37.68 (0.00001*)	43.58 (0.00000*)	35.97 (0.01118*)	36.37 (0.00108*)	38.82 (0.00000*)
DCS	35.21 (0.00001*)	36.47 (0.00001*)	29.32 (0.00474*)	29.97 (0.09760+)	32.89 (0.00488*)	37.13 (0.00000*)
DRCW	11.37	12.84	14.11	21.05	17.74	11.05

Table 12
Statistics obtained from the weights assigned to the different outputs of the base classifiers. Wc is the average weight for the outputs of the correct class, whereas WnoC corresponds to the incorrect one in a competent classifier. D0.5 is the average of the standard deviation of the weights with respect to 0.5 (ideal value) in the non-competent classifiers.

Data-set	$k=1$			$k=3$			$k=5$			$k=10$			$k=-1$		
	Wc	WnC	D0.5	Wc	WnC	D0.5	Wc	WnC	D0.5	Wc	WnC	D0.5	Wc	WnC	D0.5
Autos	0.8151	0.1849	0.1448	0.7314	0.2686	0.1265	0.6876	0.3124	0.1167	0.6408	0.3592	0.1078	0.5787	0.4213	0.0745
Car	0.9172	0.0828	0.1252	0.9000	0.1000	0.1139	0.8824	0.1176	0.1094	0.8489	0.1511	0.1024	0.5315	0.4685	0.0713
Cleveland	0.6097	0.3903	0.1040	0.6046	0.3954	0.0864	0.6030	0.3970	0.0819	0.6028	0.3972	0.0833	0.5462	0.4538	0.0424
Dermatology	0.8128	0.1872	0.1179	0.7945	0.2055	0.1133	0.7850	0.2150	0.1113	0.7710	0.2290	0.1090	0.7168	0.2832	0.1022
Ecoli	0.7793	0.2207	0.2227	0.7704	0.2296	0.2193	0.7594	0.2406	0.2110	0.7427	0.2573	0.1981	0.6576	0.3424	0.1631
Flare	0.7588	0.2412	0.1512	0.7673	0.2327	0.1482	0.7680	0.2320	0.1464	0.7656	0.2344	0.1406	0.6729	0.3271	0.0859
Glass	0.6953	0.3047	0.1321	0.6825	0.3175	0.1330	0.6757	0.3243	0.1336	0.6661	0.3339	0.1374	0.5892	0.4108	0.0995
Led7digit	0.6013	0.3987	0.1399	0.6264	0.3736	0.1375	0.6379	0.3621	0.1355	0.6572	0.3428	0.1275	0.6535	0.3465	0.1103
Lymphography	0.7769	0.2231	0.2304	0.7650	0.2350	0.2228	0.7535	0.2465	0.2112	0.7344	0.2656	0.1915	0.6541	0.3459	0.1196
Nursery	0.8660	0.1340	0.1708	0.8407	0.1593	0.1606	0.8260	0.1740	0.1544	0.8032	0.1968	0.1446	0.6278	0.3722	0.1011
Pageblocks	0.8962	0.1038	0.2838	0.8880	0.1120	0.2887	0.8866	0.1134	0.2784	0.8936	0.1064	0.2585	0.7656	0.2344	0.2220
Penbased	0.8010	0.1990	0.1090	0.7840	0.2160	0.1031	0.7729	0.2271	0.1004	0.7546	0.2454	0.0960	0.6416	0.3584	0.0698
Satimage	0.6988	0.3012	0.1181	0.6953	0.3047	0.1130	0.6932	0.3068	0.1100	0.6913	0.3087	0.1069	0.6439	0.3561	0.1006
Segment	0.8634	0.1366	0.1817	0.8479	0.1521	0.1761	0.8380	0.1620	0.1724	0.8218	0.1782	0.1662	0.6812	0.3188	0.1113
Shuttle	0.9591	0.0409	0.2328	0.9496	0.0504	0.2237	0.9433	0.0567	0.2202	0.9305	0.0695	0.2162	0.7097	0.2903	0.1509
Vehicle	0.6434	0.3566	0.1177	0.6317	0.3683	0.1037	0.6251	0.3749	0.0964	0.6143	0.3857	0.0863	0.5404	0.4596	0.0577
Vowel	0.8431	0.1569	0.1118	0.8023	0.1977	0.1054	0.7613	0.2387	0.0998	0.6799	0.3201	0.0866	0.5475	0.4525	0.0440
Yeast	0.6894	0.3106	0.2090	0.6823	0.3177	0.1969	0.6806	0.3194	0.1916	0.6835	0.3165	0.1879	0.6023	0.3977	0.1295
Zoo	0.8930	0.1070	0.0980	0.8766	0.1234	0.0836	0.8654	0.1346	0.0797	0.8482	0.1518	0.0836	0.8131	0.1869	0.0888
Average	0.7853	0.2147	0.1579	0.7706	0.2294	0.1503	0.7603	0.2397	0.1453	0.7448	0.2552	0.1384	0.6407	0.3593	0.1023

Conflict of interest

None declared.

Acknowledgments

This work was partially supported by the Spanish Ministry of Education and Science under Projects TIN2011-28488, TIN-2012-33856, TIN-2013-40765-P and the Andalusian Research Plan P10-TIC-6858, P11-TIC-9704.

References

[1] D.W. Aha, D. Kibler, M.K. Albert, Instance-based learning algorithms, Mach. Learn. 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, J. Mult. Valued Logic Soft Comput. 17 (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 Comput. 13 (3) (2009) 307–318.

[4] E.L. Allwein, R.E. Schapire, Y. Singer, Reducing multiclass to binary: a unifying approach for margin classifiers, J. Mach. Learn. Res. 1 (2000) 113–141.

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

[6] M.A. Bautista, S. Escalera, X. Baró, O. Pujol, On the design of an ecoc-compliant genetic algorithm, Pattern Recognit. 47 (2) (2014) 865–884.

[7] H. Cevikalp, R. Polikar, Local classifier weighting by quadratic programming, IEEE Trans. Neural Netw. 19 (10) (2008) 1832–1838.

[8] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, ACM Trans. Intell. Syst. Technol. 2 (2011) 27:1–27:27, software available at ⟨http://www.csie.ntu.edu.tw/~cjlin/libsvm⟩.

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

[10] J. Cohen, A coefficient of agreement for nominal scales, Educat. Psychol. Meas. 20 (1) (1960) 37–46.

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

[12] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[13] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.

[14] S.A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Syst. Man Cybern. SMC-6 (4) (1976) 325–327.

[15] F. Enríquez, F.L. Cruz, F.J. Ortega, C.G. Vallejo, J.A. Troyano, A comparative study of classifier combination applied to NLP tasks, Inf. Fusion 14 (3) (2013) 255–267.

[16] B. Fei, J. Liu, Binary tree of SVM: a new fast multiclass training and classification algorithm, IEEE Trans. Neural Netw. 17 (3) (2006) 696–704.

[17] A. Fernández, M. Calderón, E. Barrenechea, H. Bustince, F. Herrera, Solving mult-class problems with linguistic fuzzy rule based classification systems based on pairwise learning and preference relations, Fuzzy Sets Syst. 161 (23) (2010) 3064–3080.

[18] G.P.C. Fung, J.X. Yu, H. Wang, D.W. Cheung, H. Liu, A balanced ensemble approach to weighting classifiers for text classification, in: Sixth International Conference on Data Mining, ICDM, 2006.

[19] J. Fürnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.

[20] 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⟩.

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

[22] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for one-vs-one strategy: avoiding non-competent classifiers, Pattern Recognit. 46 (12) (2013) 3412–3424.

[23] S. García, J. Derrac, J. Cano, F. Herrera, Prototype selection for nearest neighbor classification: taxonomy and empirical study, IEEE Trans. Pattern Anal. Mach. Intell. 34 (3) (2012) 417–435.

[24] S. García, F. Herrera, An extension on "statistical comparisons of classifiers over multiple data sets" for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.

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

[26] T. Hastie, R. Tibshirani, Classification by pairwise coupling, Ann. Stat. 26 (2) (1998) 451–471.

[27] J.L. Hodges, E.L. Lehmann, Rank methods for combination of independent experiments in analysis of variance, Ann. Math. Stat. 33 (1962) 482–497.

[28] S. Holm, A simple sequentially rejective multiple test procedure, Scand. J. Stat. 6 (1979) 65–70.

[29] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.

[30] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recognit. 43 (1) (2010) 128–142.

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

[32] 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, Springer, Berlin, Germany, 1990, pp. 41–50.

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

[34] L.I. Kuncheva, Switching between selection and fusion in combining classifiers: an experiment, IEEE Trans. Syst. Man Cybern. B Cybern. 32 (2) (2002) 146–156.

[35] L.I. Kuncheva, Combining Pattern Classifiers: Methods and Algorithms, John Wiley & Sons, Inc., Hoboken, New Jersey, United States, 2004.

[36] B. Liu, Z. Hao, E.C.C. Tsang, Nesting one-against-one algorithm based on SVMs for pattern classification, IEEE Trans. Neural Netw. 19 (12) (2008) 2044–2052.

[37] A.C. Lorena, A.C. Carvalho, J.M. Gama, A review on the combination of binary classifiers in multiclass problems, Artif. Intell. Rev. 30 (1-4) (2008) 19–37.

[38] V. López, A. Fernández, F. Herrera, On the importance of the validation technique for classification with imbalanced datasets: addressing covariate shift when data is skewed, Inf. Sci. 257 (2014) 1–13.

[39] E. Monta nés, J. Barranquero, J. Díez, J.J. del Coz, Enhancing directed binary trees for multi-class classification, Inf. Sci. 223 (2013) 42–55.

[40] J.G. Moreno-Torres, J.A. Saez, F. Herrera, Study on the impact of partition-induced dataset shift on *k*-fold cross-validation, IEEE Trans. Neural Netw. Learn. Sys. 23 (8) (2012) 1304–1312.

[41] X.-X. Niu, C.Y. Suen, A novel hybrid CNN-SVM classifier for recognizing handwritten digits, Pattern Recognit. 45 (4) (2012) 1318–1325.

[42] G. Ou, Y.L. Murphey, Multi-class pattern classification using neural networks, Pattern Recognit. 40 (1) (2007) 4–18.

[43] J.C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, MIT Press, Cambridge, MA, USA, 1999.

[44] J.C. Platt, Probabilistic outputs for support vector machines and comparison to regularized likelihood methods, in: A. Smola, P. Bartlett, B. Schölkopf, D. Schuurmans (Eds.), Advances in Large Margin Classifiers, Cambridge, MA, 2000.

[45] J.C. Platt, N. Cristianini, J. Shawe-taylor, Large margin DAGs for multiclass classification, in: Advances in Neural Information Processing Systems, MIT Press, New York, 2000.

[46] J.R. Quinlan, C4.5: Programs for Machine Learning, 1st ed., San Morgan Kaufmann Publishers, Mateo-California, 1993.

[47] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.

[48] L. Rokach, Ensemble-based classifiers, Artif. Intell. Rev. 33 (2010) 1–39.

[49] J.A. Sáez, M. Galar, J. Luengo, F. Herrera, Analyzing the presence of noise in multi-class problems: alleviating its influence with the one-vs-one decomposition, Knowl. Inf. Syst. 38 (1) (2014) 179–206.

[50] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[51] D.R. Wilson, T.R. Martinez, Improved heterogeneous distance functions, J. Artif. Intell. Res. 6 (1997) 1–34.

[52] K. Woods, W. Philip Kegelmeyer, K. Bowyer, Combination of multiple classifiers using local accuracy estimates, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 405–410.

[53] M. Woźniak, M. Graña, E. Corchado, A survey of multiple classifier systems as hybrid systems, Inf. Fusion 16 (2014) 3–17.

[54] T.F. Wu, C.J. Lin, R.C. Weng, Probability estimates for multi-class classification by pairwise coupling, J. Mach. Learn. Res. 5 (2004) 975–1005.

[55] J. Zhou, H. Peng, C.Y. Suen, Data-driven decomposition for multi-class classification, Pattern Recognit. 41 (1) (2008) 67–76.

[56] 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 Recognit. 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 an assistant professor 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 from the University of Granada, Granada, Spain, in 2005 and 2010, respectively.

He is currently an Assistant Professor with the Department of Computer Science, University of Jaén, Spain. His research interests include data mining, classification in imbalanced domains, fuzzy rule learning, evolutionary algorithms, multiclassification problems with ensembles and decomposition techniques, and Big Data applications by means of Cloud Computing.

Dr. Fernández received the "Lofti A. Zadeh Prize" of the International Fuzzy Systems Association for the "Best paper in 2009–2010" for his work of hierarchical fuzzy rule based classification system with genetic rule selection for imbalanced data-sets. , and University of Granada prize for the best publication in the area of Engineering for his work "Advanced non-parametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining"

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 publications comprise more than 30 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 member of the board of the European Society for Fuzzy Logic and Technology (EUSFLAT).

Francisco Herrera received his M.Sc. in Mathematics in 1988 and Ph.D. in Mathematics in 1991, both from the University of Granada, Spain.

He is currently a Professor in the Department of Computer Science and Artificial Intelligence at the University of Granada. He has been the supervisor of 34 Ph.D. students. He has published more than 280 papers in international journals. He is coauthor of the book "Genetic Fuzzy Systems: Evolutionary Tuning and Learning of Fuzzy Knowledge Bases" (World Scientific, 2001).

He currently acts as Editor in Chief of the international journals “Information Fusion” (Elsevier) and “Progress in Artificial Intelligence” (Springer). He acts as an area editor of the International Journal of Computational Intelligence Systems and associated editor 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.