Information Sciences 264 (2014) 135–157

Contents lists available at ScienceDirect

Information Sciences

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

Empowering difficult classes with a similarity-based aggregation in multi-class classification problems

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

[a] Departamento de Automática y Computación, Universidad Pública de Navarra, 31006 Pamplona, Spain
[b] Department of Computer Science, University of Jaén, 23071 Jaén, Spain
[c] Department of Computer Science and Artificial Intelligence, University of Granada, 18071 Granada, Spain
[d] Faculty of Computing and Information Technology – North Jeddah, King Abdulaziz University, 21589 Jeddah, Saudi Arabia

ARTICLE INFO

Article history:
Received 1 June 2013
Received in revised form 15 October 2013
Accepted 29 December 2013
Available online 9 January 2014

Keywords:
Multi-class classification
Pairwise learning
One-vs-One
Decomposition strategies
Tuning
Difficult classes

ABSTRACT

One-vs-One strategy divides the original multi-class problem into as many binary classification problems as pairs of classes. Then, independent base classifiers are learned to face each problem, whose outputs are combined to predict a single class label. This way, the accuracy of the baseline classifiers without decomposition is usually enhanced, aside from enabling the usage of binary classifiers, i.e., Support Vector Machines, to solve multi-class problems. This paper analyzes the fact that existing aggregations favor easily recognizable classes; hence, the accuracy enhancement mainly comes from the higher correct classification rates over these classes. Using other evaluation criteria, the significant improvements of One-vs-One are diminished, showing a weakness due to the presence of difficult classes. Difficult classes can be defined as those obtaining a lower correct classification rate than that obtained by the other classes in the problem. After studying the problem of difficult classes in this framework and aiming to empower these classes, a novel similarity-based aggregation is presented, which generalizes the well-known weighted voting. The experimental analysis shows that the new methodology is able to increase the recognition of difficult classes, obtaining a more balanced performance over all classes, which is a desirable behavior. The methodology is tested within several Machine Learning paradigms and is compared with the state-of-the-art on aggregations for One-vs-One strategy. The results are contrasted by the proper statistical tests, as suggested in the literature.

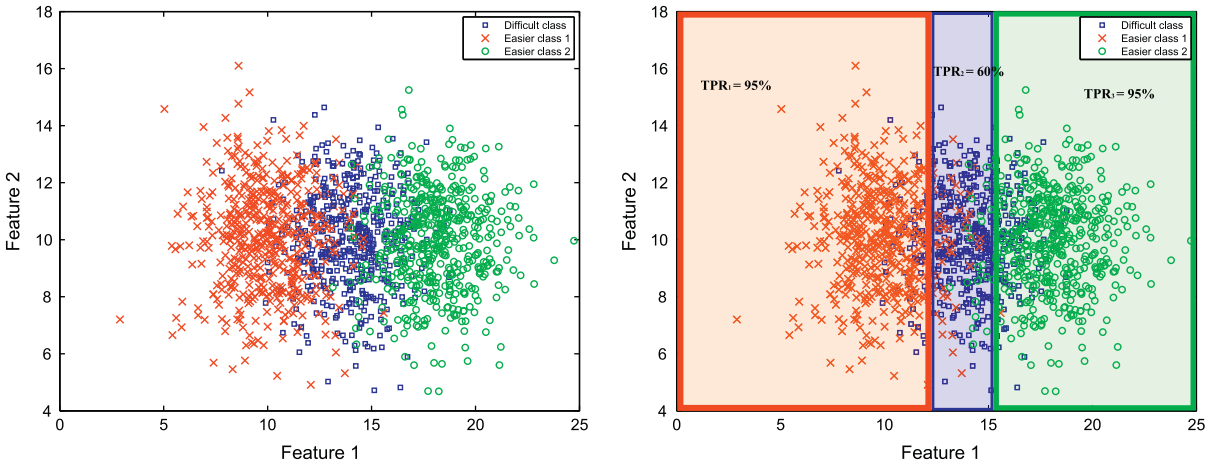
1. Introduction

Classification problems involving multiple categories are more general than their binary counterparts. When multiple classes are present, the complexity of finding the decision boundaries usually increases, making the construction of the classifiers more difficult. A number of real-world problems involve the classification of multiple classes, for instance, the classification of texts [44], microarrays [58] or textures [40].

Decomposition strategies [42] are commonly used to overcome these type of problems. In some cases, because the base classifier cannot deal with multiple classes by itself, whereas in others, because these strategies enhance the results of the baseline classifiers (without using decomposition) [24,56]. These strategies, also called binarization strategies, are based on divide and conquer paradigm, and most of them can be included within Error Correcting Output Codes (ECOC) [14,4]

* Corresponding author. Tel.: +34 948 166048; fax: +34 948 168924.
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).

0020-0255/$ - see front matter © 2014 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ins.2013.12.053



(a) A three class problem with one difficult class.

(b) Classifier maximizing accuracy rate, even though a low TPR is achieved for the difficult class.

Fig. 1. An example of the difficult classes problem. The class in the center is more difficult to be correctly classified due to its overlapping with the other two classes.

framework. One of the most commonly used strategy is One-vs-One (OVO) [37], where the original problem is divided in as many pairs of classes as possible. When a new instance is presented to all the base classifiers, which were independently learned for each pair of classes, an aggregation is used to decide the final class label. In [24], an extensive review of different aggregations was carried out, concluding that the best aggregation depends on the base learner, but the weighted voting [35] and those based on probability estimates [62] were proven to be the most robust, even though no significant differences were often found.

In the specialized literature different studies analyzing the behavior of multi-class learning algorithms have been carried out, such as those learning class structures, which speed-up the test phase in problems with a huge amount of classes [9,64,45], those studying the consistency of multi-class classification [41,59] or works dealing with the calibration of probabilistic classifiers in multi-class problems [38,20].

Within a multi-class problem, the characteristics defining the classes are usually different: for example, the number of instances, the inter-class relations and the overlapping with other classes may vary. Depending on these characteristics, some of the classes might be easier to distinguish than others. *Difficult classes* can be defined as those obtaining a lower correct classification rate; that is, the number of correctly classified examples from the class divided by the total number of examples from that class (True Positive Rate, TPR[1]). The TPR of a class varies depending on both the mentioned characteristics and the classifier used, hence, this definition is subjective, since some classes might be easier or more difficult for a classifier than for another one. However, most of the classifiers are affected by the characteristics of the classes, being this part the most important one.

In this scenario, using the most commonly considered metric, i.e., accuracy rate (percentage of correctly classified examples) as an evaluation criterion, these classes might lose their importance, since it averages the results over all instances without taking into account the TPR over each class [36]. As a consequence, it becomes easier to increase the accuracy improving the classification of the easiest classes in exchange for misclassifying some of the instances from these difficult ones. This problem comprises the well-known class imbalance problem [31,25], where the difficult classes are those under-represented in the data-set; however, the problem of difficult classes is more general, since the hitch is not only caused by the skewed class distribution. For this reason, traditional solutions proposed for class imbalance, such as balancing the data-set, are not useful in this context and different approaches must be studied.

In this paper, we aim to undertake multi-class learning problems from a different perspective and centering on a completely distinct problem, i.e., the problem of difficult classes and its possible solutions from the point of view of decomposition strategies, and more specifically, paying attention to OVO strategy. We tackle the problem with a double study:

1. OVO strategy weakens when the enhancement is sought for the difficult classes. We aim to explain the reason why this occurs, which is mainly due to the way in which the decomposition is carried out and the aggregation used.
2. We introduce a new aggregation model based on similarity measures [10], which enables the modification of the decision boundaries of the base classifiers; in such a way, the classification of the difficult classes can be boosted without changing the underlying base classifiers. Hence, this methodology is independent of the base classifier, allowing the achievement of

[1] We refer to the true values of the TPR to show the difficult classes problem in Sections 2 and 3. Since these values cannot be obtained from data, the estimated TPR, by means of the results on the test sets in the cross-validation procedure, are considered to report the results along the experiments.

solutions that cannot be obtained if the enhancement of the base classifiers is sought separately (as we will show following the example of Fig. 1 in Section 2.2).

In order to study this problem and show the validity of our proposal in the framework of OVO strategy, the experiments carried out in this paper include a set of twenty-eight real-world problems from KEEL [3,2] and UCI [5] data-set repositories and several well-known classifiers from different paradigms: Support Vector Machines (SVMs) [60], decision trees [53] and instance-based learning [1]. Besides the usage of the accuracy rate to evaluate the performance of the classifiers, we include other measures [36] accounting for the problem of difficult classes (introduced in Section 2.2). The comparisons among the results obtained are contrasted using the proper statistical tests, as suggested in the literature [13,28].

The remainder of this paper is as follows. In Section 2, the problem of difficult classes is analyzed, focusing on OVO strategy. Next, Section 3 shows our proposal to empower the difficult classes in OVO. The tuning of the parameters to benefit the difficult classes is presented in Section 4. The experimental framework used to carry out the empirical analysis is presented in Section 5. The experimental analysis, showing the weakness of OVO and studying the validity of our aggregation, is developed in Section 6. Thereafter, Section 7 discusses the results obtained and the future research lines emerged out of this work. Finally, Section 8 concludes the paper.

2. Difficult classes problem in One-vs-One strategy

In this section, we start recalling the basis of OVO strategy and its simplest aggregation, the voting method (SubSection 2.1). Then, we introduce the problem of difficult classes and we analyze the performance measures that are considered in this paper (SubSection 2.2). Finally, we aim to show the difficult classes problem in OVO scheme (SubSection 2.3).

2.1. One-vs-One decomposition

OVO consists of dividing a m-class problem into $m(m-1)/2$ binary subproblems considering all the possible pairs of classes, these subproblems are formed of the instances from the pair of classes considered and are faced by independent base learners. This binarization procedure is supposed to produce simpler subproblems, aside from enabling the classifiers originally designed to deal with two-class problems (e.g., SVM [60,33,55,34,32]) to address multiple classes problems.

In order to label a new instance, it is submitted to all the base classifiers. Each classifier, distinguishing a pair of classes $\{C_i, C_j\}$, outputs a confidence degree $r_{ij} \in [0,1]$ in favor of C_i; thus, the confidence in favor of C_j is computed as $r_{ji} = 1 - r_{ij}$. All the confidence degrees can be organized within a score-matrix R, given by Eq. (1):

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

From the score-matrix, the final output class is inferred. Different aggregations have been presented in the literature to perform this task [24]. The simplest aggregation, yet powerful, is the voting strategy, where each classifier votes for its predicted class, and the class obtaining the largest number of votes is predicted:

$$\text{Class} = \arg \max_{i=1,\ldots,m} \sum_{1 \leqslant j \neq i \leqslant m} s_{ij} \tag{2}$$

where s_{ij} is 1 if $r_{ij} > r_{ji}$ and 0 otherwise.

The aggregation phase is critical in OVO scheme. Given that the decomposition procedure is fixed, the way in which the class is selected can alter the predictions, and therefore, most of the problems of OVO strategy arise at this point. The selection of the aggregation method is the first problem; notice that if all the classifiers distinguishing the real class of the instance make a correct prediction, any aggregation should be able to predict the correct class. However, if any of them fails, the aggregation strategy must decide upon the class to be predicted, which might not be the real class. Despite the number of existing aggregations, no significant differences are usually found among their usage [24], since all of them start from the same score-matrices. Another problem is the so-called non-competence [29,26]. Some of the classifiers giving a confidence for the instance to be classified have no knowledge about the real class of the instance (they have not been trained using instances from that class); hence, they might distort the results of the aggregation. Even though this problem has received low attention in the literature, it has been shown that reducing the number of non-competent classifiers can lead to an enhancement of the results [26]. Hierarchical strategies [45] can also reduce the number of non-competent classifiers evaluated, but these strategies are mainly focused on reducing the computational complexity rather than in improving the results obtained, avoiding non-competent classifiers. In this work, we focus on another problem that we have identified in OVO strategy, the difficult classes problem, which we put forward hereafter.

2.2. Difficult classes problem

The characteristics of each class within a problem are directly translated into their degree of separability. Even though all the classes in a problem could be equally separable, this is not the usual case, since their characteristics generally differ. The simplest way to present the difficult classes problem is by means of the example shown in Fig. 1(a). One can observe that the class in the center is more difficult to distinguish than the other two due to the class overlapping. In this example, a classifier maximizing accuracy could define the three regions shown in Fig. 1(b), from which the classification would be carried out.

In this case, two of the classes have achieved a high TPR (TPR $= 95\%$), whereas the TPR of the difficult one is only of the 60%, being the accuracy rate of the 83.33%. Assuming that the difficult class is the concept of interest of the problem, or at least it is as important as the rest of the classes, it would be interesting to obtain lower TPRs for the easier classes, while increasing that of the difficult one. For example, obtaining a TPR $= 83.33\%$ (homogeneous) for all the classes, which would lead to the same accuracy rate of the 83.33%, which is completely different from the real situation; in this case, all classes would be equally recognized, whereas in the previous, there was a difficult class achieving a low TPR. This last result could be much better than the previous one in many problems that require to equally recognize all classes [52,47]. As we have shown in this example, a very important point when dealing with difficult classes is that the most commonly used metric does not reflect the problem, since the same accuracy rate is achieved in both cases, whereas the results on the difficult class highly differ.

This is why in this paper, in addition to the accuracy rate, we need to consider other measures accounting for the difficult classes problem [36]. We should recall that the accuracy rate is computed as

$$Acc = \frac{1}{n_T}\sum_{i=1}^{m}\mathrm{TPR}_i \cdot n_i, \tag{3}$$

where n_i is the number of examples of class i and n_T is the total number of examples evaluated. In fact, the issue that the accuracy rate is the weighted mean of the TPRs over each class, where the weights are given by the proportion of examples from each class, makes this measure inadequate to account for the importance of all classes at the same time. The new measures should consider the TPR over each class without taking into account the number of examples. There are two well-known metrics that could be considered:

- The Average Accuracy rate (AvgAcc) [19],

$$\mathrm{AvgAcc} = \frac{1}{m}\sum_{i=1}^{m}\mathrm{TPR}_i. \tag{4}$$

- The Geometric Mean (GM) [6],

$$\mathrm{GM} = \sqrt[m]{\prod_{i=1}^{m}\mathrm{TPR}_i}. \tag{5}$$

The AvgAcc is different from the accuracy rate, since it partially accounts for the TPR over the different classes; nonetheless, a low rate on one class might not severely affect this measure. Moreover, the greater the number of classes is, the easier to hide low TPRs in some classes becomes. Otherwise, the GM accounts for the maximization of the TPR over all classes at the same time, strongly penalizing those cases achieving a low TPR in any of the classes. In fact, a TPR $= 0\%$ in any of the classes will produce a GM $= 0$. In the rest of the paper, we will show that the GM is the measure better accounting for the problem we aim to undertake, whereas the AvgAcc could serve as a complementary measure, but cannot be used alone. All these facts can be easily understood with the example we have presented. Table 1 shows the values obtained for each one of the measures considered in this paper for the two classifiers we have supposed (the one obtaining heterogeneous TPRs, different in each class, and the one obtaining homogeneous TPRs, the same in all classes).

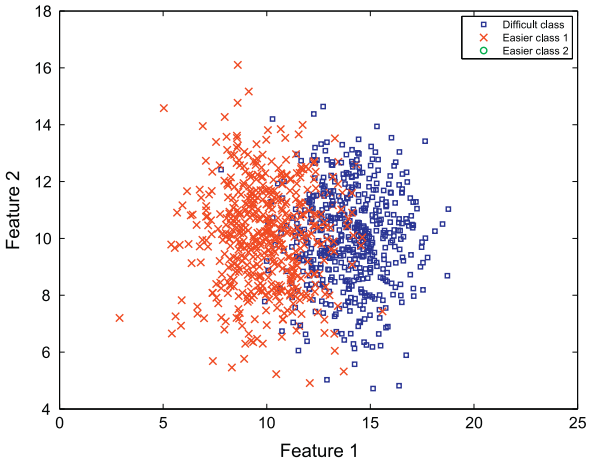
2.3. A weakness of One-vs-One strategy: dealing with difficult classes

The fact that the base (binary) classifiers of OVO are optimal (in terms of Acc, AvgAcc, GM and balance between the TPRs over both classes in the subproblem) need not mean that their combination would lead to the optimal solution in terms of Acc, AvgAcc, GM and balance among all the TPRs over all the classes of the problem. This can be clearly observed following

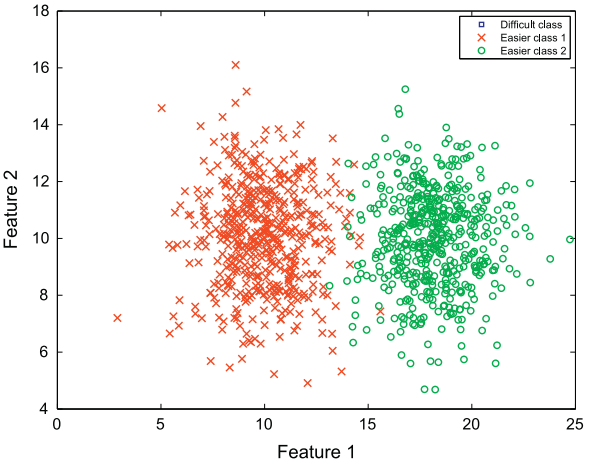
Table 1
On the use of performance measures in the difficult classes problem. Homogeneous classifier refers to that obtaining the same TPR for all classes, whereas Heterogeneous refers to that obtaining a different TPR for each class.

Classifier	TPR_1	TPR_2	TPR_3	Acc (%)	AvgAcc (%)	GM
Heterogeneous	0.95	0.6	0.95	83.33	83.33	0.8151
Homogeneous	0.8333	0.8333	0.8333	83.33	83.33	0.8333

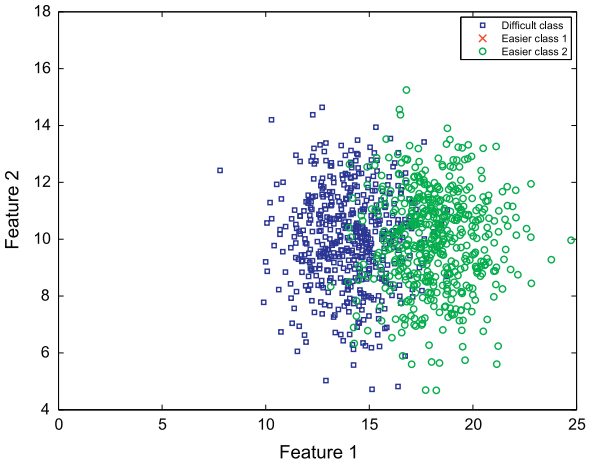
the example of Fig. 1. We divide this problem into three new subproblems considering each pair of classes (Fig. 2(a–c)), which are faced by independent base classifiers whose decision boundaries are shown in Fig. 2(d–f). The TPRs obtained by these classifiers for each one of the two classes could be considered equal for the same classifier (balanced for both



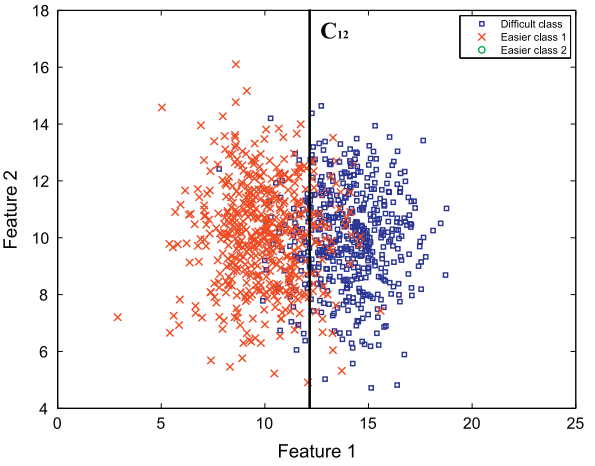
(a) Class 1 vs. Class 2.



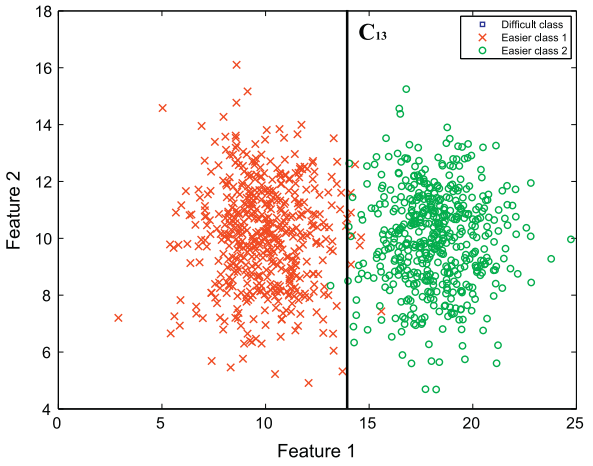
(b) Class 1 vs. Class 3.



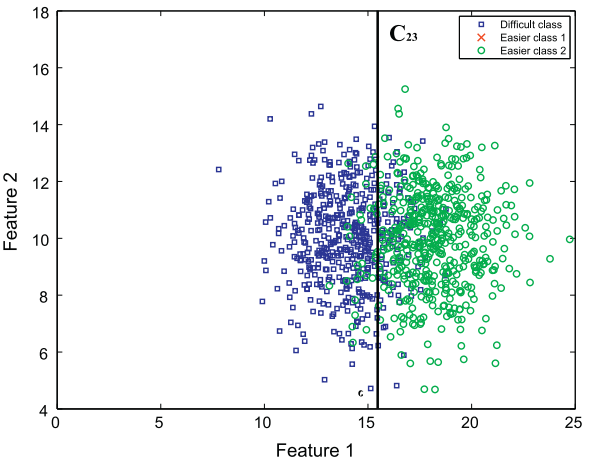
(c) Class 3 vs. Class 2.



(d) Classifier 1 vs. 2.



(e) Classifier 1 vs. 3.



(f) Classifier 2 vs. 3.

Fig. 2. OVO decomposition of the problem in Fig. 1(a) and the base classifiers learned for this decomposition.

classes, i.e., locally optimal in terms of Acc, AvgAcc, GM). However, notice that their combination considering the voting scheme (Eq. (2)) would lead exactly to the class separation in Fig. 1(b), which is optimal in terms of Acc but achieves a low GM due to the low TPR over the difficult class.

For this reason, we aim to study the low TPR achieved over the difficult classes in OVO strategy, or rather the non-existent improvement over those classes. In other words, the usually reported [24,22] accuracy enhancement of OVO strategy over the baseline classifiers (without decomposition) is mainly due to the improvement of the TPR over the easiest classes. In Section 6, we will also show this fact by an exhaustive experimental analysis.

Hereafter, our aim is to formalize why OVO strategy tends to improve the accuracy over the easiest classes, ignoring (not enhancing) the classification of the difficult ones. In order to do so, we consider a simple scenario, where OVO strategy using the voting strategy is considered.

Problem statement and notation.

- m-class problem, $\mathbb{C} = \{C_1, \ldots, C_m\}$.
- There are m_d classes which are much more difficult to classify (for example, due to overlapping, noise, or even imbalance).
- The rest of the classes are easier to be classified.
- Let TPR^i_{ij} be the TPR over class C_i of the classifier distinguishing classes $\{C_i, C_j\}$.

Problem assumptions.

1. Independence of the base classifiers, which is supposed in OVO scheme.
2. An instance is correctly classified if all the competent base classifiers [26] (those considering the real class of the instance in the training phase) correctly classify the instance (although in some cases a fail might not suppose an erroneous prediction).
3. Given a difficult class (C_i) and an easier class (C_j) then, $\text{TPR}^i_{ik} \leqslant \text{TPR}^j_{jt}$ for all $k, t = 1, \ldots, m,\ k \neq i, t \neq j$ and there exist $p, q \in \{1, \ldots, m\},\ p \neq i, q \neq j$ such that $\text{TPR}^i_{ip} < \text{TPR}^j_{jq}$.

The first assumption suppose that the output labels given by the classifiers are not related (one does not depend on the other), which by definition is assumed in OVO classifiers. The second one refers to the fact that all the predictions of the competent classifiers need to be correct in order to correctly classify an instance. We are aware that this might be an over-simplification as there might be cases in which a single fail will not affect the final prediction. However, it could be supposed that an instance is classified into a class whenever there is a total agreement between the base classifiers, which can be considered in systems requiring a high confidence in the decision. In this scenario it will help us showing the difficult classes problem in OVO. Finally, the last assumption deals with the fact that the TPRs obtained in each base classifier by a difficult class are always lower or equal than the corresponding TPRs over the easier classes, being also a classifier in which this relation is strict (the TPR over the difficult class is lower).

Problem description.

Given an instance $\{\mathbf{x}, y\}$ (where $\mathbf{x}$ represents the input attribute values and $y = C_i$ with $i \in \{1, \ldots, m\}$), the probability of being correctly classified, denoted as $P(h_{ovo}(\mathbf{x}) = y)$ (where h_{ovo} stands for the OVO classifier), following Assumptions 1 and 2, is given by the TPR of each one of the base classifiers that used instances from this class to be trained (that is, TPR^i_{ij} for all $j = 1, \ldots, m$ with $i \neq j$):

$$P(h_{ovo}(\mathbf{x}) = y) = \prod_{1 \leqslant j \neq i \leqslant m} \text{TPR}^i_{ij}. \tag{6}$$

Therefore, we can consider an instance $\{\mathbf{x}_1, y_1\}$ (belonging to one of the easier classes, i.e., $y_1 = C_i$) and an instance $\{\mathbf{x}_2, y_2\}$ (belonging to one of the difficult classes, i.e., $y_2 = C_j$) to be classified, whose probabilities of being correctly classified are given by Eq. (6). Following Assumption 3, we have that

$$P(h_{ovo}(\mathbf{x}_1) = y_1) = \prod_{1 \leqslant k \neq i \leqslant m} \text{TPR}^i_{ik} > \prod_{1 \leqslant t \neq j \leqslant m} \text{TPR}^j_{jt} = P(h_{ovo}(\mathbf{x}_2) = y_2), \tag{7}$$

showing that the probability of correctly classifying the instance from the difficult class will always be lower than that of correctly classifying the instance from the easier class due to the differences in the TPRs of the base classifiers. In order to understand how these differences can affect this probability, in Table 2 we show the probability of correctly classifying an instance with increasing number of classes considering different TPRs in the base classifiers dealing with the correct class of the instance (for the sake of simplicity we assume that for the class i, $\text{TPR}^i_{ij} = \text{TPR}^i_{ik}$ for all $j, k = 1, \ldots, m,\ j \neq i$ and $k \neq i$). It can be observed that the probability of correctly classifying a class having low TPRs in the base classifiers decreases much more rapidly than that of a class having greater TPRs.

How can this problem be solved or at least alleviated?

1. The improvement of the TPR^i_{ij} for each difficult class i ($j = 1, \ldots, m,\ j \neq i$).

Table 2
Probability of correctly classifying an instance of a class with a specific TPR in all each base classifiers and increasing number of classes.

Classes	TPR 0.5	TPR 0.55	TPR 0.6	TPR 0.65	TPR 0.7	TPR 0.75	TPR 0.8	TPR 0.85	TPR 0.9	TPR 0.95	TPR 1.0
2	0.5	0.55	0.6	0.65	0.7	0.75	0.8	0.85	0.9	0.95	1.0
3	0.250	0.303	0.360	0.423	0.490	0.563	0.640	0.723	0.810	0.903	1.0
4	0.125	0.166	0.216	0.275	0.343	0.422	0.512	0.614	0.729	0.857	1.0
5	0.063	0.092	0.130	0.179	0.240	0.316	0.410	0.522	0.656	0.815	1.0
6	0.031	0.050	0.078	0.116	0.168	0.237	0.328	0.444	0.591	0.774	1.0
7	0.016	0.028	0.047	0.075	0.118	0.178	0.262	0.377	0.531	0.735	1.0
8	0.008	0.015	0.028	0.049	0.082	0.134	0.210	0.321	0.478	0.698	1.0
9	0.004	0.008	0.017	0.032	0.058	0.100	0.168	0.273	0.431	0.663	1.0
10	0.002	0.005	0.010	0.021	0.040	0.075	0.134	0.232	0.387	0.630	1.0
11	0.001	0.003	0.006	0.014	0.028	0.056	0.107	0.197	0.349	0.599	1.0
12	0.0005	0.001	0.004	0.009	0.020	0.042	0.086	0.167	0.314	0.569	1.0

2. The usage of aggregations taking into account the difficult classes problem, without needing to alter the underlying base classifiers.

The former is the straightforward solution, but the enhancement of the TPRs is not easy to accomplish. Moreover, we have shown in the example (Fig. 1) that even though all the pair of classes in the same base classifiers have the same TPRs, the difficult classes problem can be present and hence, the learning of the base classifier should be biased toward the difficult classes (improving the TPR over the difficult class at the expenses of the other class). The problem is that when the learning is going to be carried out the presence of the difficult classes might be unknown and therefore altering the learning of the classifiers would require the difficult classes to be detected a priori (which is not trivial), acting in consequence. Besides, in the case of a base classifier dealing with two difficult classes, the biasing would be even more difficult. For these reasons, we focus on the latter case, which could also be combined with the enhancement of the base classifiers. This approach does not alter the underlying base classifiers, but use them differently. Hence, it could be used with any classifier, since it is independent of the base classifier.

Our aim is to modify the classification of the instances by making those belonging to the difficult classes easier to predict. To do so, we favor the difficult classes in the aggregation by a flexible (parametrized) similarity-based aggregation. As a consequence, the aggregation should identify and empower the difficult classes depending upon the results obtained in the training set. This method can be understood as a post-processing, where the votes of the classifiers are modified depending on the difficulty of classifying each class. One of the advantages of this method is that it performs a global optimization with all the classifiers at the same time, which is not considered by standard OVO combinations. In this manner, the outputs of the classifiers which were independently learned can be analyzed together, and these outputs can be modified in such a way that globally better solutions can be obtained. The global process allows us to reach solutions which could be overlooked when locally seeking for optimal base classifiers configurations.

We believe that there is enough information within the score-matrices of OVO strategy as to obtain significantly different results over the difficult classes only changing the aggregation. For this reason, we fix the score-matrices used in the experiments of this paper (which vary depending on the base classifier), and we aim to learn from the errors committed by each classifier so that we can adjust the aggregation to empower the classification of the difficult classes. Therefore, all the differences shown in this paper are only due to the aggregation and have nothing to do with the base classifiers, which is of great importance in order to properly evaluate the performance of the proposed methodology.

3. Similarity-based aggregation for OVO strategy

In this section, we put forward the new aggregation method for OVO scheme based on similarity measures to account for the difficult classes problem. To do so, we first recall several preliminary concepts in SubSection 3.1 that are needed to understand the origin of the aggregation presented in SubSection 3.2.

3.1. Restricted equivalence functions and similarity measures

We need to recall some concepts and operations before showing the similarity measures considered in our aggregation. A negation models the concept of opposite:

Definition 1. A mapping $n : [0,1] \to [0,1]$ with $n(0) = 1, n(1) = 0$, strictly decreasing, and continuous is called *strict negation*. Moreover, if n is involutive, i.e., if $n(n(a)) = a$ for all $a \in [0,1]$, then n is called a *strong negation*.

Restricted equivalence functions (REFs) [10,11] measure the degree of closeness (equivalence) between two points; in their definition the concept of negation is used.

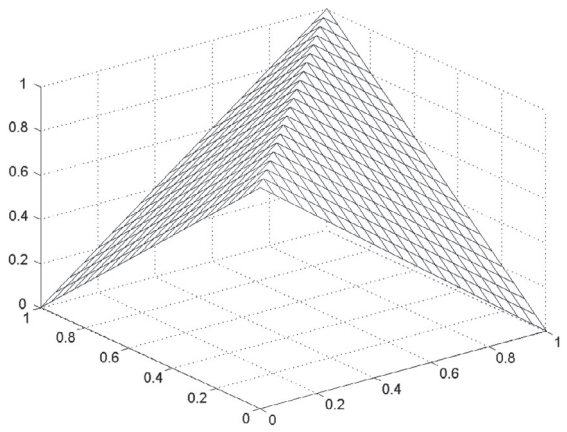
Definition 2. [10,11] A function $\mathrm{REF} : [0,1]^2 \to [0,1]$ is called restricted equivalence function associated with the strong negation n, if it satisfies the following conditions

1. $\mathrm{REF}(a,b) = \mathrm{REF}(b,a)$ for all $a, b \in [0,1]$;
2. $\mathrm{REF}(a,b) = 1$ if and only if $a = b$;
3. $\mathrm{REF}(a,b) = 0$ if and only if $a = 1$ and $b = 0$ or $a = 0$ and $b = 1$;
4. $\mathrm{REF}(a,b) = \mathrm{REF}(n(a), n(b))$ for all $a, b \in [0,1]$;
5. For all $a, b, c \in [0,1]$, if $a \leqslant b \leqslant c$, then $\mathrm{REF}(a,b) \geqslant \mathrm{REF}(a,c)$ and $\mathrm{REF}(b,c) \geqslant \mathrm{REF}(a,c)$.

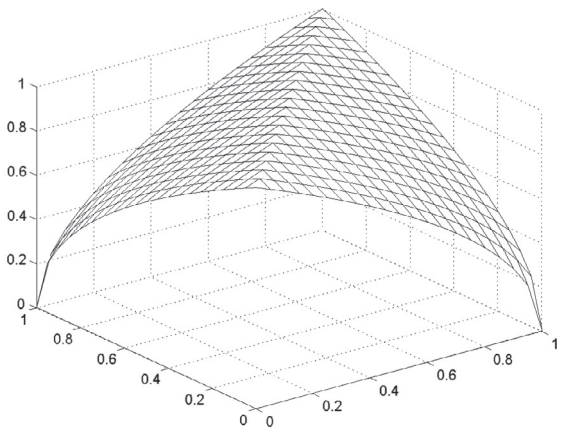
In this paper, the interest of this closeness measure resides in the possibility of its parameterization by means of automorphisms as follows (other construction methods can be found in [10,12]).

Definition 3. A continuous, strictly increasing function $\varphi : [a,b] \to [a,b]$ such that $\varphi(a) = a$ and $\varphi(b) = b$ is called automorphism of the interval $[a,b] \subset \mathbb{R}$.

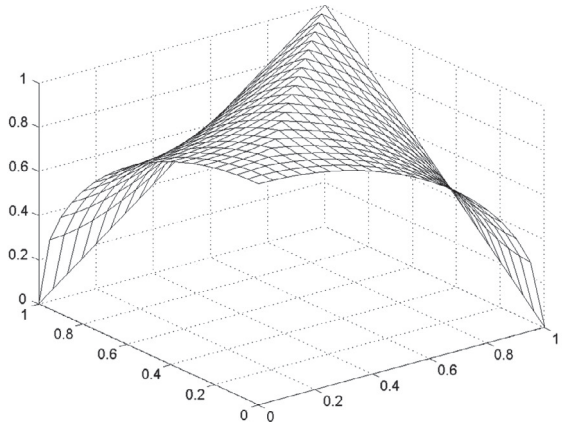
Proposition 1 [10]. *Let φ_1, φ_2 be two automorphisms of the interval $[0,1]$. Then $\mathrm{REF}(a,b) = \varphi_1^{-1}(1 - |\,\varphi_2(a) - \varphi_2(b)\,|)$ is a restricted equivalence function associated with the strong negation $n(a) = \varphi_2^{-1}(1 - \varphi_2(a))$.*



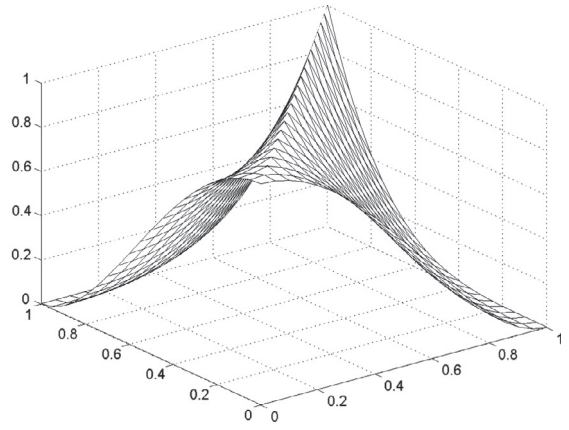
(a) $\varphi_1(a) = a$ and $\varphi_2(a) = a$, $\mathrm{REF}(a,b) = 1 - |a - b|$.



(b) $\varphi_1(a) = a^2$ and $\varphi_2(a) = a$, $\mathrm{REF}(a,b) = \sqrt{1 - |a - b|}$.



(c) $\varphi_1(a) = a^2$ and $\varphi_2(a) = a^2$, $\mathrm{REF}(a,b) = \sqrt{1 - |a^2 - b^2|}$.



(d) $\varphi_1(a) = a^{1/2}$ and $\varphi_2(a) = a$, $\mathrm{REF}(a,b) = (1 - |a - b|)^2$.

Fig. 3. Influence of the automorphisms in the construction of a REF.

An easy way of constructing automorphisms is by means of a parameter $\lambda \in (0,\infty) : \varphi(a) = a^{\lambda}$, and hence, $\varphi^{-1}(a) = a^{1/\lambda}$. In Fig. 3, several examples of REFs are shown, which illustrate the effect of changing the pair of automorphisms used in the construction of REFs. This parameterization allow us to alter the confidences of the base classifiers, changing the automorphisms associated with each confidence position in the score-matrix.

Similarity measures are an extension of the concept of closeness to compare tuples ($\mathbf{a} = (a_1,\ldots,a_N) \in [0,1]^N$) instead of single points.

Definition 4. [10] A function $SM : [0,1]^N \times [0,1]^N \to [0,1]$ is called a similarity measure with respect to the strong negation n if it satisfies the following properties:

(i) $SM(\mathbf{a},\mathbf{b}) = SM(\mathbf{b},\mathbf{a})$;
(ii) $SM(\mathbf{a},\Psi(\mathbf{a})) = 0$ if and only if $a_i = 0$ or $a_i = 1$ for all $i \in \{1,\ldots,N\}$, where $\Psi(\mathbf{a}) = (n(a_1),\ldots,n(a_N))$;
(iii) $SM(\mathbf{a},\mathbf{b}) = 1$ if and only if $a_i = b_i$ for all $i \in \{1,\ldots,N\}$;
(iv) If $\mathbf{a} \leqslant \mathbf{b} \leqslant \mathbf{c}$, then $SM(\mathbf{a},\mathbf{b}) \geqslant SM(\mathbf{a},\mathbf{c})$ and $SM(\mathbf{c},\mathbf{b}) \geqslant SM(\mathbf{c},\mathbf{a})$;
(v) $SM(\Psi(\mathbf{a}),\Psi(\mathbf{b})) = SM(\mathbf{a},\mathbf{b})$.

These similarity measures satisfy similar properties to those of REFs. This fact allows one to construct them using REFs and an aggregation function [8,27].

Definition 5. An aggregation function is a mapping $M : [0,1]^N \to [0,1]$ such that:

(A_1) $M(a_1,\ldots,a_N) = 0$ if and only if $a_1 = \ldots = a_N = 0$;
(A_2) $M(a_1,\ldots,a_N) = 1$ if and only if $a_1 = \ldots = a_N = 1$;
(A_3) M is nondecreasing.

Proposition 2 [10]. *Let $M : [0,1]^N \to [0,1]$ be an aggregation function and let $REF : [0,1]^2 \to [0,1]$ be a restricted equivalence function associated with the strong negation n. Then*

$$SM(\mathbf{a},\mathbf{b}) = M(REF(a_0,b_0),\ldots,REF(a_N,b_N))$$

is a similarity measure associated with the strong negation n.

The most commonly used aggregation function, also considered in this paper, is the arithmetic mean. Hence, the similarity measures considered along this paper are as follows (using Propositions 1 and 2, with $\varphi_1(x) = x^{\lambda_1}$ and $\varphi_2(x) = x^{\lambda_2}$):

$$SM(\mathbf{a},\mathbf{b}) = \frac{1}{N}\sum_{i=1}^{N} REF(a_i,b_i) = \frac{1}{N}\sum_{i=1}^{N} \varphi_1^{-1}(1 - |\,\varphi_2(a_i) - \varphi_2(b_i)\,|) = \frac{1}{N}\sum_{i=1}^{N} (1 - |\,(a_i)^{\lambda_2} - (b_i)^{\lambda_2}\,|)^{1/\lambda_1} \tag{8}$$

3.2. Generalizing the weighted voting method: an aggregation based on similarity measures

Our similarity-based aggregation generalizes the well-known Weighted Voting strategy (WV), whose robustness has been both theoretically [35] and empirically [24] proven. In WV, the confidences of the base classifiers are used as weights to vote for the classes and the class with the largest total confidence is given as final output class:

$$Class = \arg\max_{i=1,\ldots,m} \sum_{1 \leqslant j \neq i \leqslant m} r_{ij} \tag{9}$$

In our case, instead of summing up the confidences of the classifiers in each row, we compare these confidences to the certain vote (i.e., 1.0), since it is the case in which the highest vote should be given. Hence, the more similar the confidence r_{ij} to 1.0 is, the more importance the vote has. In order to compare both values, we make use of REFs; therefore, instead of voting using r_{ij}, we consider the weight of the vote given by $REF(r_{ij},1)$, which indicates how close is r_{ij} to the certain vote. Recalling the construction of REFs from Proposition 1, the operations and parameters needed for the comparison can be simplified as follows:

$$REF(a,1) = (1 - |\,a^{\lambda_2} - 1^{\lambda_2}\,|)^{1/\lambda_1} = a^{\lambda_2/\lambda_1} = a^{\lambda} \tag{10}$$

Therefore, both parameters of the REF (λ_1,λ_2) are reduced to a single equivalent one (λ). Fig. 4 depicts the influence of λ when comparing a single value to 1 using REFs. Notice that $\lambda = 1$ does not modify the vote of the classifier, whereas values below one ($\lambda < 1$) empowers the weights and the contrary occurs with $\lambda > 1$. We should notice that using different construction of REFs (with other automorphisms) different behaviors could be achieved. Anyway, from our point of view, the proposed one is the most appropriate, in the sense that it is the most suitable to model the empowering of the difficult classes and the weakening of the easier ones.

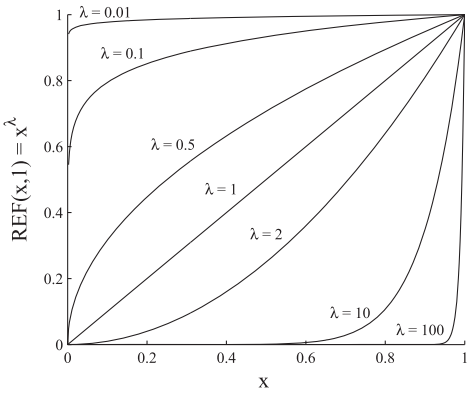


Fig. 4. Influence of the parameter λ in REF$(x,1)$.

Remark 1. Hence, looking at Fig. 4, one can observe that the outputs in favor of the difficult classes should use a REF with a low λ, whereas those of the easier classes might consider a higher value of λ in order to seek for a balance between their predictions. The estimation of these parameters is not trivial, since their values directly affect the final predictions; for this reason, Section 4 is devoted to their adjustment, which is carried out globally considering all the base classifiers predictions at the same time.

We have outlined how the confidences given by the base classifiers are empowered or weakened; then, similarly to WV, these confidences are aggregated in each row; formally, the similarity measure between the confidences in each row and the tuple $\mathbf{1} = (1, 1, \ldots, 1)$ is maximized:

$$\text{Class} = \arg\max_{i=1,\ldots,m} \text{SM}_i(\mathbf{r}_i, \mathbf{1}) = \arg\max_{i=1,\ldots,m} \frac{1}{m-1} \sum_{1 \leqslant j \neq i \leqslant m} \text{REF}_{ij}(r_{ij}, 1) = \arg\max_{i=1,\ldots,m} \frac{1}{m-1} \sum_{1 \leqslant j \neq i \leqslant m} (r_{ij})^{\lambda_{ij}} \tag{11}$$

where $\mathbf{r}_i$ corresponds to the ith row of the score matrix and λ_{ij} is the corresponding parameter used in $\text{REF}_{ij}(r_{ij}, 1)$. We denote $\text{SM}_i(\mathbf{r}_i, \mathbf{1})$ and $\text{REF}_{ij}(r_{ij}, 1)$ to indicate that each similarity measure for each class, and each REF within each similarity measure can use different values in their parameter setting. Finally, notice that WV method is recovered when $\lambda_{ij} = 1$ for all $i, j = 1, \ldots, m$ and $i \neq j$ (see Proposition 2 in [10]). As we will show in Section 4, we will consider a single parameter for each base classifier (that is, we have as many parameters as degrees of freedom in the score-matrix) and hence, each parameter λ_{ij} is related to the corresponding λ_{ji} as follows: $\lambda_{ji} = \frac{1}{\lambda_{ij}}$ for all $i, j = 1, \ldots, m$ and $i < j$.

4. Adapting the similarity-based aggregation to enhance the classification of difficult classes in One-vs-One strategy

In the previous section we have presented the aggregation that will allow us to deal with the difficult classes problem in OVO strategy. Nevertheless, the aggregation does not solve the problem by itself, it needs to be adjusted depending upon the difficulty of classifying each class in each problem. This difficulty is analyzed depending on the outputs encoded in the score-matrices given by the base classifiers for each training instance. In this post-processing stage, any optimization algorithm maximizing the objective function established could be used; in our case, we consider a GA [30], and more specifically, the real-coded CHC algorithm [16] (similarly to other works [57]).

It is important to emphasize that such a mechanism is required due to the complex global adjustment that is carried out. Recall that the global adjustment is needed because locally adjusting each base classifier without taking into account the rest of the classifiers need not translate in a global improvement. Otherwise, adjusting the classifiers globally allows the search algorithm to observe the interactions between the base classifiers, leading to a better global adjustment. Moreover, there are cases in which small changes in a parameter of a base classifier could imply a change in the predicted class, but also the contrary might occur (there are base classifiers which do not have influence in the classification of some instances).

Once the importance of the adjustment has been stated, the rest of the section is organized as follows: in SubSection 4.1, we present the most important part of the optimization procedure, i.e., the fitness function, which is maximized aiming to benefit the difficult classes. Then, in SubSection 4.2, we recall the operations of the CHC algorithm and we introduce the representation considered to codify the parameters of the similarity measures. Finally, in SubSection 4.3, we discuss the computational complexity of the optimization procedure.

4.1. Objective function

The key factor of the parameter tuning is the fitness function to be optimized, since depending upon this function different aggregations are obtained. Notice that the standard accuracy rate must not be further optimized, as it is usually done [18,57], because it does not account for the difficult classes. For this reason, we consider other measures based on those

presented in Section 2.2 in order to determine the quality of the system obtained with a given set of parameters $\boldsymbol{\lambda} = \left(\lambda_1, \lambda_2, \ldots, \lambda_{\frac{m(m-1)}{2}}\right)$ (a parameter is encoded for each base classifier in the score-matrix, according to its degrees of freedom). We propose the usage of the following fitness function (Eq. (12)) to perform the evaluation.

$$\text{Fitness}(\boldsymbol{\lambda}) = \text{Margin}(\boldsymbol{\lambda}) + \frac{\text{GM}(\boldsymbol{\lambda}) + \text{AvgAcc}(\boldsymbol{\lambda})}{2}, \tag{12}$$

where the *Margin* quantifies how well is the real class of an instance separated from the second class with the highest similarity, that is, the margin of separation, a similar concept to that of SVMs [60]. The concept of margin can be defined for each instance, but the value used in the objective function must comprise all the instances. Thus, the global margin is computed as follows.

1. For each correctly classified instance $\{\mathbf{x}, y\}$ (with $y = C_i$) the margin is computed as

$$\text{Margin}(\mathbf{x}) = \frac{\text{SM}_i(\mathbf{r}_i, 1) - \text{SM}_j(\mathbf{r}_j, 1)}{n_{Tr} \cdot m}, \tag{13}$$

where C_j is the second class with the largest similarity value. Since C_i is the correctly predicted class, it is the one with the largest similarity, and therefore, the margin is always positive. The margin is normalized by the number of classes and training instances (n_{Tr}) to reduce its influence in the fitness function with respect to the other factors.
2. Among all the margins computed, we take the minimum one, since it is the value better representing how well separated are the most conflictive (difficult) classes

$$\text{Margin} = \min_{k=1,\ldots,n_c} \text{Margin}(\mathbf{x}_k), \tag{14}$$

where n_c is the number of correctly classified instances.

The most important part of the fitness function, and our main objective, is the GM, since it is the measure which better balances the accuracy over all classes. Nevertheless, the other factors are needed according to the following facts:

1. AvgAcc has a priori the same weight, but even though its value is higher, its variations depending upon the correctly classified instances are generally lower, and hence, it has less influence when comparing different evaluations of the fitness function. However, it is a very important factor in cases where GM value is low, since it serves as guide for the search (being a class with initial TPR = 0, GM = 0). In these cases, AvgAcc helps in improving the classification over the rest of the classes, which eventually can lead to correctly classify an instance of the class with a low TPR.
2. Margin has a low weight in the fitness function (due to its normalization), but it is a key component serving as a stabilization process once the best GM and AvgAcc combination has been found. At this point, increasing the minimum margin of separation is useful to properly establish the separation boundaries between classes, and hence, it is an over-fitting prevention mechanism.

We should emphasize the importance of the margin in the fitness function, since we are developing a global adjustment aiming to increase the GM and AvgAcc the system can easily overfit using the training set in this procedure. This way of avoiding overfitting is more effective than considering an additional data partition for validation. As we are dealing with multi-class problems, there are many cases where there are a few number of examples from some classes and hence, further partitioning the data will lead to use not enough number of examples to learn the base classifiers, which will directly compromise the performance of the system (the performance decrease in the base classifiers due to the use of a separate validation partition cannot be recovered by the methodology proposed). For this reason, the same training set used to train the OVO system is used in this optimization procedure.

This objective function can be related to an asymmetric loss case [51,65]. In our case, the instances of each class contribute differently to the function depending upon the difficultly of classifying the corresponding class. This way, the proposed method allow one to empirically determine the skew (costs) and hence, the asymmetric loss ratios from the data itself, avoiding the biggest problem of cost-sensitive classification, the definition of the costs.

4.2. CHC algorithm and representation of the parameters

The objective function defined (Eq. (12)) can be optimized with any optimization method capable of handling such a complex search space. The real-coded CHC algorithm [16] was selected for this purpose due to its successful application in similar tuning approaches [18,57]. CHC holds a good balance between exploration (looking into the whole search space) and exploitation (finding an accurate solution), being an appropriate meta-heuristic for complex search spaces.

In this GA, all the M chromosomes in the population (encoded solutions) and their offsprings (which are found by the crossover operator over the chromosomes in the population) are put together; then, the M best individuals (in terms of the fitness function, i.e., Eq. (12)) make up the next population. Instead of using a mutation operator as most of the GAs do, an incest prevention mechanism combined with a re-initialization of the population is used to promote diversity

(as explained hereafter). The necessary components to design the whole process are: representation of the solutions (how the parameters are encoded within a chromosome, which is a key factor), initialization of the initial population, crossover operator, incest prevention and restarting mechanisms.

1. *Representation of the parameters:* the set of parameters ($\boldsymbol{\lambda}$) to be optimized are real parameters, so they are the elements (called genes) of a chromosome. Recall that the value of λ ranges from 0 to ∞, which cannot be directly encoded within a chromosome. Therefore, we use the following chromosome ($\Phi(\boldsymbol{\lambda})$) to encode $\boldsymbol{\lambda}$:

$$\Phi(\boldsymbol{\lambda}) = \left(\phi(\lambda_1), \phi(\lambda_2), \ldots, \phi\left(\lambda_{\frac{m(m-1)}{2}}\right)\right) = \left(c_{\lambda_1}, c_{\lambda_2}, \ldots, c_{\lambda_{\frac{m(m-1)}{2}}}\right) \tag{15}$$

where each gene $c_{\lambda_i} \in (0,1)$, $i = 1, \ldots, \frac{m(m-1)}{2}$. Since the values of the genes are in $(0,1)$, they need to be translated to the values of the parameters in order to evaluate the fitness function, which is done as follows

$$\lambda_i = \phi^{-1}(c_{\lambda_i}) = \begin{cases} (2 \cdot c_{\lambda_i})^2 & \text{if } c_{\lambda_i} \leqslant 0.5 \\ \frac{1}{(2\cdot(1-c_{\lambda_i}))^2} & \text{otherwise.} \end{cases} \tag{16}$$

In this manner, the square operation allows us to homogeneously explore the whole search space (Fig. 4). That is, 0.5 in the gene value (c_{λ_i}) is equal to 1 in the parameter value (λ_i); the upper part of the expression corresponds to the upper part of the parameter values in Fig. 4 and the same occurs with the lower part of the expression and that of Fig. 4, respectively.

2. *Initialization:* All the chromosomes are randomly initialized in $(0,1)$ except for the first individual, which is initialized with 0.5 in all each genes. This way, the search is started with an individual (solution) representing the original WV, that is, the proposed aggregation with $\boldsymbol{\lambda} = \mathbf{1}$ (following Eq. (16)).
3. *Crossover operator:* This operator allows one to combine two chromosomes of the population to generate their offspring. We use the Parent Centric BLX operator [43], which works as follows. Let $\mathbf{a} = (a_a, \ldots, a_N)$ and $\mathbf{b} = (b_1, \ldots, b_N)$ ($\mathbf{a}, \mathbf{b} \in [0,1]^N$, being N) be two real-coded chromosomes of length N. Their crossover generates two offspring.
   (a) $\mathbf{o}_1 = (o_{11}, \ldots, o_{1N})$, where o_{1i} is randomly (uniformly) chosen number from the interval $\left[l_i^1, u_i^1\right]$, with $l_i^1 = \max\{0, a_i - I_i\}$, $u_i^1 = \min\{1, a_i + I_i\}$, and $I_i = \mid a_i - b_i \mid$.
   (b) $\mathbf{o}_2 = (o_{21}, \ldots, o_{2N})$, where o_{2i} is randomly (uniformly) chosen number from the interval $\left[l_i^2, u_i^2\right]$, with $l_i^2 = \max\{0, b_i - I_i\}$ and $u_i^2 = \min\{1, b_i + I_i\}$.
4. *Incest prevention:* It promotes diversity among solutions (which is important to properly search the whole search space). This mechanism prevents the crossover of parents if their Hamming distance (divided by two) is below a threshold value L (i.e., they are too similar). Since we consider real-coded chromosomes, they need to be transformed in order to compute the distance. A Gray Code using *#BITS* bits per gene is used with this purpose. The initial value of the threshold is computed as $L = (\#\text{Genes} \cdot \#\text{BITS})/4.0$, where #Genes stands for the number of genes (in our case, $m(m-1)/2$). As in the original CHC algorithm, L is decreased by one (in this case, by #BITS), when no new individuals (offspring) are created (no parents have been crossed due to this mechanism).
5. *Restarting mechanism:* In the CHC algorithm, the mutation operator is replaced by this mechanism in order to avoid local optima (improve the exploration). When the threshold value L is lower than zero, all the chromosomes in the population are randomly regenerated in the $(0,1)$ interval. Besides, the current best solution is included in the population.

There are two stopping criteria to finish the optimization process: the number of evaluations and the number of restarting procedures without improvements. Their set-up is detailed in SubSection 5.1. Finally, recall that the optimization procedure is carried out over the whole training set.

4.3. Computational complexity

Regarding the computational complexity, we should note that logically, our proposal is computationally more expensive than standard OVO aggregations, since an optimization phase using a GA is introduced, which depends on the number of classes and instances. However, as well as in any other post-processing technique, the training time is just taken into account once per data-set, being the testing time of the proposed aggregation equivalent to those of the other methods. Hence, its application in classification problems that do not require on-line training might not be compromised, offering the possibility of increasing the classification over the difficult classes, as we show in Section 6.

More specifically, the computational complexity of the method is bounded by the maximum number of evaluations allowed (which we set to $1000 \cdot m^2$, see Table 3). In each evaluation all the instances are classified following Eq. (11). But notice that since this is a post-processing method, the base classifiers are only trained once before starting the process, and hence, the cost of each evaluation is no more than that of the aggregation. Therefore the computational complexity of the optimization process is of $\mathcal{O}(n \cdot m^2)$. In Section 6, we will report the execution time of the method with each base classifier, showing that effectively, the number of classes is the most important factor when considering the computational complexity, whereas increasing number of instances does not imply a severe increase in the execution time.

Table 3
Parameter specification.

Algorithm	Parameters
3NN	$k = 3$, Distance metric = Heterogeneous Value Difference Metric (HVDM)
C4.5	Prune = True, Confidence level = 0.25, Minimum number of item-sets per leaf = 2
SVM_{Poly}	C = 1.0, Tolerance = 0.001, Epsilon = 1.0E−12, Kernel = Polynomial Polynomial Degree = 1, Fit Logistic Model = True
SVM_{Puk}	C = 100.0, Tolerance = 0.001, Epsilon = 1.0E−12, Kernel = Puk PukKernel $\omega = 1.0$, PukKernel $\sigma = 1.0$, Fit Logistic Model = True
CHC	Population size $(M) = 50$ individuals, Evaluations = $1000 \cdot m^2$ $\#BITS = 30$, Restarting procedures without improvement = 3

5. Experimental framework

In this section, the set-up of the experimental framework used to carry out the empirical analysis in Section 6 is presented. First in SubSection 5.1, the base classifiers and their parameter setting are described. Next, in SubSection 5.2, we recall the best aggregations for each base classifier found in [24], which are the base for the comparisons. Then in SubSection 5.3, we show the real-world problems tested in the experimentation. Finally, we present the performance measures considered in the evaluation and the statistical tests applied in SubSection 5.4.

5.1. Base classifiers and parameter configuration

In order to show the problem of OVO with difficult classes and to test our proposed solution, we have selected several well-known Machine Learning algorithms as base learners. We should mention that the whole experimental set-up is similar to that in [24], where the state-of-the-art on aggregations for the OVO strategy were compared. The algorithms used in the comparison are the following ones:

- *kNN - k-Nearest Neighbors* [1].
- *C4.5* – decision tree [53].
- *SVM – Support Vector Machine* [60,48].

These learning algorithms were selected due to their good performance in a large number of real problems, being all included in the top ten Data Mining algorithms [63]. Moreover, in case of SVM there is not an established multi-class extension yet, although there are several attempts [33].

Most of the aggregation methods for OVO classification make use of the confidences given by the base classifiers, which are obtained as follows:

- *kNN: Confidence* $= \frac{\sum_{l=1}^{k} \frac{e_l}{d_l}}{\sum_{l=1}^{k} \frac{1}{d_l}}$ 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. When $k > 1$, the probability estimate depends on the distance from the neighbors, hence the estimation is not restricted to a few values. This approach can also be considered as weighted k-Nearest Neighbors [15].
- *C4.5:* The confidence is obtained from the accuracy of the leaf making the prediction, that is, the percentage of correctly classified training instances reaching the leaf (preliminary experiments considering Laplace smoothing [50] produced similar results).
- *SVM:* The probability estimates from the SVM logistic model [49] are used as confidence degrees.

There are aggregations where ties could occur, in those cases, as usual, the majority class is predicted, if the tie continues, the class is randomly selected.

The configuration parameters considered to train the base classifiers are shown in Table 3, along with the parameters used in the CHC algorithm. These values are common for all problems, and they were selected according to the recommendation of the corresponding authors, which is the default parameter setting included in KEEL software [3,2] used to develop our experiments. In the case of SVMs, we considered two configurations, varying the parameter C and the kernel function, to study the behavior of the aggregations with different set-ups, which addresses for the robustness of the proposal (in the sense that in spite of the fine-tuning of the base classifiers, its behavior is maintained). At last, we treat nominal attributes in SVM as scalars to fit the data into the systems using a polynomial kernel.

Even though tuning the parameters of each method on each particular problem could lead to better results (mainly in SVM), we prefer to maintain a baseline performance of each method as the basis for comparison. We are not comparing base classifiers among them; hence, our hypothesis is that the methods winning on average on all problems would also perform

better if a more optimal setting would be performed. Moreover, in a framework where no method is tuned, the best methods tend to correspond to the most robust ones, which is also a desirable characteristic.

5.2. Aggregations considered

We use a different aggregation depending upon the base classifier used to analyze their results with respect to the new aggregation methodology. We follow the findings in our previous work [24], where a representative aggregation was selected for each base classifier (the best one). The unique exception is the case of SVMs, where the best performer aggregation was Nesting OVO [39], but without showing significant differences with respect to the rest of the aggregations. Furthermore, despite constructing several OVO ensembles one inside the other recursively, this strategy does not perform better than other simpler methods such as the probability estimates method by Wu et al. [62], which use is much more extended. For this reason, being the latter method equivalent but also simpler, we use it as a representative; this way, we are also able to compare all methods using exactly the same score-matrices in all aggregations, only focusing the comparison on the differences between the aggregations themselves. The following aggregations are considered:

- *kNN* – ND (Non-Dominance criterion [17]).
- *C4.5* – WV (Weighted Voting strategy).
- *SVM* – PE (Wu et al. Probability Estimates [62]).

For the sake of brevity, we do not recall the operating procedure of PE and ND; their description can be found in their original source papers, but also an extensive and detailed description is available in [23].

5.3. Data-sets

We have used twenty-eight data-sets from KEEL [2] and UCI [5] data-set repositories. Data-sets with a large representation of different number of classes and attributes have been considered. Table 4 summarizes the properties of these data-sets: the number of examples (#Ex.), the number of attributes (#Atts.), the number of numerical (#Num.) and nominal (#Nom.) attributes, and the number of classes (#Cl.) are shown. Some of the largest data-sets (nursery, page-blocks, penbased, satimage and shuttle) were stratified sampled at 10% in order to reduce the computational time required for training the base classifiers (reduced data-set properties are shown). In the case of missing values (autos, cleveland and dermatology), we removed those instances from the data-set before doing the partitions. The information of the data-sets is

Table 4
Summary description of data-sets.

Data-set	#Ex.	#Atts.	#Num.	#Nom.	#Cl.
Balance	625	4	4	0	3
Contraceptive	1473	9	9	0	3
Hayes-roth	132	4	4	0	3
Iris	150	4	4	0	3
NewThyroid	215	5	5	0	3
Splice	319	60	0	60	3
Tae	151	5	5	0	3
Thyroid	720	21	21	0	3
Wine	178	13	13	0	3
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

Table 5
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}
Balance	625	3	288	49	288								
Contraceptive	1473	3	629	333	511								
Hayes-roth	132	3	51	51	30								
Iris	150	3	50	50	50								
NewThyroid	215	3	30	35	150								
Splice	319	3	77	77	165								
Tae	151	3	49	50	52								
Thyroid	720	3	17	37	666								
Wine	178	3	59	71	48								
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

completed with the number of instances per class in each data-set (Table 5). 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 performance of the classifiers was estimated by means of a stratified 5-fold cross-validation. The data partitions used in this paper can be found in KEEL-dataset repository [2] and in the website associated with our previous work [24] (http://sci2s.ugr.es/ovo-ova/), which makes the experimental study to be easily reproducible.

5.4. Performance measures and statistical tests

As we have previously stated, we consider the accuracy rate, GM and AvgAcc, as performance measures to evaluate the results. These measures allow us to show that the benefit of OVO in terms of accuracy comes from the easier classes and to properly analyze the performance over all classes (mainly, the GM as we have previously explained, but the AvgAcc serves as a complementary analysis).

The comparison of the performance of the classifiers must be done using the proper statistical analysis to find whether significant differences exist or not among them. In order to carry out this process appropriately non-parametric tests should be considered, according to the recommendations made in [13,28]. These tests are needed because the conditions guaranteeing the reliability of the parametric tests may not be satisfied, losing the credibility of the statistical analysis [13]. Any interested reader can find additional information on the thematic website http://sci2s.ugr.es/sicidm/, where software for the application of the statistical tests is provided.

In this paper, we consider the Wilcoxon paired signed-rank test [61] as a non-parametric statistical procedure to perform comparisons between two algorithms, since we carry out the comparisons in a pairwise manner (comparing our methodology in each base classifier against the best state-of-the-art aggregation).

6. Experimental study

Hereafter, we carry out the experiments using different base learners with a twofold objective:

1. To show the weakness of OVO to classify the difficult classes, that is, to analyze how the significant differences that are usually found when accuracy measure is used [24] vanishes when measures accounting for the difficult classes problem are considered. Hence, we compare OVO strategy (with classic aggregations) to the baseline classifier (without decomposition) using accuracy, GM and AvgAcc. We will show that the significant differences obtained with accuracy disappear when the other measures are considered.

2. To study the validity of our aggregation proposal based on similarity measures to enhance the classification of the difficult classes in OVO strategy. Therefore, we compare our methodology based on the similarity measure (from this point denoted as SM) with OVO using the previous aggregation and with the baseline classifier, considering the same three performance measures.

In order to develop this study, this section is divided into three subsections (one for each base classifier) from which the main conclusions extracted will be discussed in Section 7. In addition, we include another subsection showing the execution times of the methodology. Within each one of the first three subsections, the structure is defined as follows: first, we address the former point and then we tackle the latter point. We should recall that all the differences shown among the methods using OVO are only due to the aggregation phase, since the same score-matrices are used in each base classifier.

6.1. 3NN as base classifier

In Table 6, we show the test results for each method and performance measure considered using 3NN as base classifier (note that, accuracy and AvgAcc are presented as percentages, as usual). Base refers to the baseline classifier without using OVO, whereas ND and SM correspond to OVO strategy using ND and SM aggregations, respectively. OVO methods achieve the highest accuracy values (SM does not hinder the accuracy results, despite focusing on difficult classes). In terms of GM, the differences of ND with respect to the baseline 3NN decrease. Otherwise, SM is able to increase the performance of ND, achieving the highest result. Lastly, AvgAcc shows that SM outstands among the three methods, but ND also excels with respect to the baseline 3NN. Anyway, these facts must be contrasted with the proper statistical tests in order to extract meaningful conclusions. Hence, we carry out the Wilcoxon tests, whose results are shown in Table 7.

Table 7 is divided into two parts. The first one compares the previous OVO aggregation with the baseline classifier, whereas the second one is devoted to the comparison of our methodology with the other two methods. In the former comparison, only significant differences are found in case of AvgAcc (with $\alpha = 0.1$), which was not expected, but can be explained as follows: OVO allows one to improve the classification over the easier classes, when this improvement is large enough, their influence in the AvgAcc can also be significant if the accuracy over the difficult classes is not hindered (which does not usually occur). That is, AvgAcc does not properly account for difficult classes problem, as previously stated, since low rates in some classes can be unnoticed due to high TPRs in others. In other respects, the ranks in terms of accuracy rate are in favor of ND, but the p-value is not low enough to reject the null hypothesis of equivalence. Moreover, when

Table 6
Results using 3NN as base classifier.

Data-set	Accuracy			GM			AvgAcc		
	Base	OVO		Base	OVO		Base	OVO	
		ND	SM		ND	SM		ND	SM
Autos	71.05	**79.23**	72.94	**.5741**	.3821	.4021	61.33	**72.24**	68.15
Balance	**83.52**	82.72	80.48	.0000	.0000	**.0868**	**60.43**	59.83	58.77
Car	**96.07**	93.11	92.13	.9050	.8390	**.9264**	**93.21**	90.11	92.72
Cleveland	54.55	**57.25**	51.53	.0663	.0657	**.1600**	26.93	33.70	**36.49**
Contraceptive	46.16	**48.27**	46.98	.4175	.4203	**.4718**	43.12	46.69	**47.54**
Dermatology	**96.38**	90.24	89.96	**.9593**	.8265	.8232	**95.65**	85.19	84.92
Ecoli	79.77	79.77	**79.79**	.1556	.1319	**.1609**	68.64	68.50	**70.83**
Flare	72.23	**72.42**	71.29	.5089	.3962	**.5405**	59.32	60.65	**63.40**
Glass	71.03	**72.44**	66.37	.5551	**.5625**	.5553	64.75	65.67	**69.55**
Hayes-Roth	31.03	74.96	**77.24**	.3829	.7631	**.7769**	35.09	73.35	**79.29**
Iris	**95.33**	94.67	93.33	.9240	**.9378**	.9275	**95.33**	94.67	93.33
Led7digit	42.60	**72.00**	70.60	.0000	.2274	**.6834**	41.18	**72.05**	70.75
Lymphography	**85.08**	83.72	82.39	**.7087**	.5223	.5084	**74.94**	74.31	73.52
NewThyroid	**96.28**	95.35	95.81	**.9484**	.9423	.9423	91.59	90.25	**94.60**
Nursery	92.36	92.52	**93.13**	.6992	.6492	**.7359**	83.50	84.28	**88.89**
Pageblocks	94.52	94.88	**95.07**	.3111	.4800	**.6405**	67.18	73.18	**82.25**
Penbased	**96.91**	96.64	96.36	.9681	**.9692**	.9624	**96.90**	96.66	96.37
Satimage	86.16	86.16	**86.78**	.8248	.8223	**.8463**	82.00	82.30	**85.39**
Segment	96.10	96.62	**96.75**	.9662	**.9712**	.9668	96.10	96.62	**96.75**
Shuttle	99.54	**99.63**	**99.63**	.5657	**.7665**	.3665	73.65	76.23	**80.21**
Splice	89.65	91.24	**94.05**	.8723	.9095	**.9406**	91.51	93.46	**94.15**
Tae	35.74	37.08	**39.81**	**.3829**	.3786	.3622	35.58	37.04	**40.03**
Thyroid	93.61	**94.58**	92.92	.3003	.4813	**.6288**	50.21	57.42	**69.52**
Vehicle	70.69	**71.40**	69.62	**.6515**	.6460	.6505	70.99	**71.71**	69.98
Vowel	95.86	**96.06**	95.35	.9894	**.9915**	.9504	95.86	**96.06**	95.35
Wine	96.05	97.17	**98.87**	.9760	.9708	**.9882**	96.76	97.68	**98.86**
Yeast	**55.33**	54.18	52.02	**.2916**	.1721	.0925	**53.29**	52.01	50.87
Zoo	93.05	90.14	**93.10**	**.3782**	.3484	**.3782**	**85.24**	79.84	84.05
Average	79.17	**81.94**	81.23	.5815	.5919	**.6241**	71.08	74.35	**76.30**

Table 7
Wilcoxon tests for 3NN as base classifier.

Comparison	Measure	R^+	R^-	Hypothesis	p-Value
ND vs. base	Accuracy	256.0	150.0	Not rejected	0.227476
	GM	188.5	217.5	Not rejected	0.754794
	AvgAcc	289.0	117.0	Rejected for ND at 90%	0.050182
SM vs. ND	Accuracy	145.5	260.5	Not rejected	0.186360
	GM	300.0	106.0	Rejected for SM at 95%	0.027187
	AvgAcc	315.0	91.0	Rejected for SM at 95%	0.010760
vs. Base	Accuracy	188.0	218.0	Not rejected	0.732674
	GM	251.5	154.5	Not rejected	0.279642
	AvgAcc	329.0	77.0	Rejected for SM at 95%	0.004115

R^+ are ranks in favor of the first algorithm in the comparison and R^- in favor of the second one.

considering the GM, the weakness of OVO shows up, in spite of being no differences, the sum of ranks turns in favor of the baseline classifier.

The latter comparison shows the influence of our methodology. SM statistically outperforms the previous OVO aggregation in terms of GM and AvgAcc (both with $\alpha = 0.05$), being not at the expenses of losing accuracy (which comparison shows that are almost equivalent). Hence, it has accomplished its objectives. However, in comparison with the baseline 3NN, its behavior is similar to that of ND, accuracy and GM tests are not rejected, but in this case, the ranks of GM are in favor of SM on the contrary to the case of ND. Besides, AvgAcc is now rejected at ($\alpha = 0.05$), showing that SM has further improved ND.

6.2. C4.5 as base classifier

The test results with C4.5 as base learner are presented in Table 8. The largest accuracy is achieved by WV, whereas SM reaches the highest GM and AvgAcc values with a large advantage, which was the expected behavior. These results are contrasted using the appropriate statistical analysis in Table 9, where the results of the Wilcoxon tests are shown.

Table 8
Results using C4.5 as base classifier.

Data-set	Accuracy			GM			AvgAcc		
	Base	OVO		Base	OVO		Base	OVO	
		WV	SM		WV	SM		WV	SM
Autos	76.73	81.17	**81.19**	.6048	.6514	**.6671**	75.55	79.22	**80.05**
Balance	77.28	**80.00**	70.56	.0000	.0000	**.5012**	55.92	57.88	**58.50**
Car	90.80	93.00	**93.58**	.7871	.9197	**.9402**	80.21	92.20	**94.19**
Cleveland	**51.82**	51.53	49.82	**.0000**	**.0000**	**.0000**	**28.14**	26.39	24.87
Contraceptive	51.93	**52.48**	52.07	.4962	.4976	**.5108**	50.34	50.51	**51.33**
Dermatology	92.46	**96.37**	96.08	.8922	**.9541**	.9450	90.29	**95.70**	94.87
Ecoli	78.28	**79.47**	77.68	**.1605**	.1564	.1564	63.94	**65.89**	63.10
Flare	**74.48**	74.20	69.79	.1014	.0000	**.4376**	60.51	58.08	**63.01**
Glass	68.73	**70.53**	68.22	.5174	.5073	**.6258**	66.85	**67.50**	67.20
Hayes-Roth	**83.30**	**83.30**	**83.30**	**.8379**	**.8379**	**.8379**	**85.45**	**85.45**	**85.45**
Iris	**93.33**	**93.33**	**93.33**	**.9289**	**.9289**	**.9289**	**93.33**	**93.33**	**93.33**
Led7digit	70.60	**72.20**	71.20	.6783	**.6939**	.6869	70.72	**72.22**	71.20
Lymphography	**75.01**	73.63	67.63	**.6638**	.4689	.4716	**74.22**	67.59	69.63
NewThyroid	91.16	**93.95**	93.49	.8835	**.9130**	.9109	89.30	**91.68**	91.46
Nursery	**89.04**	89.04	83.57	.3909	.0000	**.6907**	69.29	65.76	**84.01**
Pageblocks	95.07	**95.61**	91.78	.3168	.4932	**.6515**	72.47	78.52	**80.80**
Penbased	89.36	**90.64**	90.27	.8903	**.9032**	.9008	89.31	**90.63**	90.31
Satimage	80.09	**81.65**	81.34	.7499	**.7594**	.7556	77.12	**78.07**	77.70
Segment	96.32	**97.06**	96.93	.9622	**.9700**	.9687	96.32	**97.06**	96.93
Shuttle	99.54	**99.72**	**99.72**	.0000	**.5997**	**.5997**	68.87	**91.85**	**91.85**
Splice	79.31	**89.02**	**89.02**	.7235	**.8726**	**.8726**	75.61	**87.60**	**87.60**
Tae	**57.66**	51.08	49.05	**.5638**	.4943	.4356	**57.46**	51.25	49.00
Thyroid	**98.75**	98.33	98.33	**.9682**	.8887	.8887	**97.03**	90.92	90.92
Vehicle	71.87	71.39	**72.93**	.6776	.6285	**.6946**	72.09	71.65	**73.20**
Vowel	79.49	80.00	**80.91**	.7852	.7891	**.7993**	79.49	80.00	**80.91**
Wine	**94.90**	92.13	92.13	**.9479**	.9158	.9158	**94.85**	91.98	91.98
Yeast	55.80	**59.91**	55.53	.0000	.0000	**.1057**	54.06	**57.06**	55.55
Zoo	**94.10**	93.10	92.10	**.3782**	**.3782**	**.3782**	**85.48**	85.12	85.12
Average	80.62	**81.57**	80.06	.5681	.5793	**.6528**	74.08	75.75	**76.57**

Table 9
Wilcoxon tests for C4.5 as base classifier.

Comparison	Measure	R^+	R^-	Hypothesis	p-Value
WV vs. base	Accuracy	297.5	108.5	Rejected for WV at 95%	0.028026
	GM	224.5	181.5	Not rejected	0.757760
	AvgAcc	259.5	146.5	Not rejected	0.218023
SM vs. WV	Accuracy	64.5	341.5	Rejected for WV at 95%	0.001729
	GM	284.5	121.5	Rejected for SM at 95%	0.017583
	AvgAcc	219.0	187.0	Not rejected	0.497915
vs. Base	Accuracy	182.5	223.5	Not rejected	0.602590
	GM	322.0	84.0	Rejected for SM at 95%	0.009322
	AvgAcc	290.5	115.5	Rejected for SM at 90%	0.055168

R^+ are ranks in favor of the first algorithm in the comparison and R^- in favor of the second one.

In this case, the difficult classes problem in OVO is clearly shown. WV statistically outperforms C4.5 in terms of accuracy, but this difference vanishes when GM and AvgAcc are tested. In the comparison of the proposed method against WV and the baseline C4.5, the outstanding behavior of SM in terms of GM (our main objective) excels, statistically outperforming both methods with very low p-values. In terms of AvgAcc, the hypothesis of equivalence in the comparison with C4.5 is also rejected (which WV was not able to achieve), whereas the same does not occur when comparing with WV. Finally, accuracy has not suffer a large decrease with respect to C4.5 (being equivalent), but WV outperforms SM considering this measure. This means that the benefit in terms of GM has come along with a loss of accuracy due to the large GM enhancement.

6.3. SVM as base classifier

Finally, in the case of SVM as base classifier, since there is not an established extension to multi-class problems, we study the benefit of applying our methodology with respect to the commonly used aggregation (PE). To do so, we consider two different configurations as explained in SubSection 5.1. The results obtained with SVM_{Poly} as base classifier are shown in Table 10, whereas their corresponding statistical analysis is presented in Table 11.

Table 10
Results using SVM_{Poly} as base classifier.

Data-set	Accuracy OVO		GM OVO		AvgAcc OVO	
	PE	SM	PE	SM	PE	SM
Autos	74.80	**75.38**	.5479	**.5624**	**72.69**	71.99
Balance	90.40	**91.68**	.8310	**.9156**	85.35	**91.79**
Car	92.71	**93.34**	.8651	**.9364**	87.18	**93.71**
Cleveland	**58.25**	51.16	.0000	**.0756**	30.88	**34.52**
Contraceptive	49.83	**50.71**	.4604	**.5102**	47.34	**51.54**
Dermatology	**94.13**	93.85	**.9408**	.9362	**94.58**	94.30
Ecoli	**77.69**	76.49	**.1544**	.1517	**68.18**	67.77
Flare	**74.67**	72.79	.4517	**.5914**	61.02	**65.54**
Glass	**61.26**	59.81	.2045	**.4596**	55.40	**61.78**
Hayes-Roth	52.22	**71.14**	.4985	**.7069**	55.05	**72.30**
Iris	**96.00**	**96.00**	.9580	**.9583**	**96.00**	**96.00**
Led7digit	**73.00**	71.80	**.7110**	.7014	**73.01**	71.90
Lymphography	81.68	**83.77**	**.3348**	.3325	64.87	**73.13**
NewThyroid	**97.21**	95.81	.9599	**.9621**	96.16	**96.38**
Nursery	**91.90**	91.43	.6529	**.6990**	82.22	**85.39**
Pageblocks	**94.70**	86.49	.3042	**.6658**	68.23	**78.89**
Penbased	95.27	**95.64**	.9513	**.9554**	95.29	**95.66**
Satimage	**84.14**	83.67	.7703	**.8015**	79.55	**81.36**
Segment	92.55	**93.85**	.9197	**.9359**	92.55	**93.85**
Shuttle	96.37	**96.92**	.3477	**.3631**	80.67	**83.30**
Splice	79.59	**80.22**	.8325	**.8374**	84.29	**84.69**
Tae	51.72	**55.72**	.4869	**.5407**	51.91	**55.57**
Thyroid	95.69	**96.94**	.4445	**.8817**	67.88	**89.29**
Vehicle	72.46	**73.05**	**.6970**	.6892	72.82	**73.49**
Vowel	69.90	**72.22**	.6822	**.7050**	69.90	**72.22**
Wine	**97.16**	**97.16**	**.9684**	**.9684**	**96.99**	**96.99**
Yeast	**59.10**	54.58	.0000	**.4088**	**56.74**	56.69
Zoo	**95.05**	**95.05**	**.0000**	**.0000**	**85.24**	**85.24**
Average	80.34	**80.60**	.5706	**.6519**	74.00	**77.69**

Table 11
Wilcoxon tests for SVM_{Poly} as base classifier.

Comparison	Measure	R^+	R^-	Hypothesis	p-Value
SM vs. PE	Accuracy	219.0	187.0	Not rejected	0.756995
	GM	364.5	41.5	Rejected for SM at 95%	0.000220
	AvgAcc	361.0	45.0	Rejected for SM at 95%	0.000266

R^+ are ranks in favor of SM and R^- in favor of PE.

Using this configuration SM achieves the highest values in the three measures. In the case of accuracy, the difference is low, but in the other two are remarkable, mainly in GM, which is highly enhanced. Similar conclusions are drawn from the statistical tests. Both methods achieve equivalent accuracies, but SM behavior in terms of GM and AvgAcc is superb, rejecting the null hypotheses of equivalence with very low p-values.

Regarding the second configuration, SVM_{Puk}, the results (shown in Table 12) are similar, but not so large differences are shown at first glance. In this case, SM continues achieving the highest values in the three measures, maintaining its excellent performance in terms of GM and AvgAcc. These results are contrasted with the proper statistical analysis in Table 13. The superiority of SM is remarkable as shown by the results of the tests. Whereas the accuracy remains similar, both GM and AvgAcc are improved, rejecting the null hypotheses of equivalence with low p-values. The different behaviors between both

Table 12
Results using SVM_{Puk} as base classifier.

Data-set	Accuracy OVO		GM OVO		AvgAcc OVO	
	PE	SM	PE	SM	PE	SM
Autos	**68.53**	61.51	**.2544**	.2156	**65.06**	59.78
Balance	**88.00**	87.84	**.8660**	.8497	**86.93**	85.71
Car	63.60	**71.18**	.7452	**.7763**	77.58	**80.37**
Cleveland	**45.09**	44.75	**.0000**	**.0000**	**29.78**	29.41
Contraceptive	**48.41**	45.01	.4406	**.4555**	45.70	**46.31**
Dermatology	**96.09**	95.26	**.9574**	.9478	**96.03**	95.29
Ecoli	**75.31**	75.01	.1381	**.1550**	67.35	**67.64**
Flare	**69.42**	64.35	.3277	**.5188**	59.43	**60.10**
Glass	70.60	**70.61**	.5372	**.5533**	68.04	**68.59**
Hayes-Roth	79.54	**81.05**	.8072	**.8163**	82.30	**83.58**
Iris	94.00	**94.67**	.9375	**.9442**	94.00	**94.67**
Led7digit	70.20	**70.80**	.6840	**.6928**	70.32	**71.01**
Lymphography	80.34	**81.01**	.1557	**.3374**	54.98	**61.65**
NewThyroid	**97.67**	**97.67**	**.9811**	**.9811**	**98.16**	**98.16**
Nursery	81.33	**83.33**	.6793	**.6902**	82.28	**83.72**
Pageblocks	**94.16**	93.43	**.2757**	.2666	**67.40**	65.41
Penbased	**97.82**	**97.82**	**.9781**	**.9781**	**97.85**	**97.85**
Satimage	84.92	**85.23**	.8315	**.8434**	84.16	**85.08**
Segment	97.10	**97.23**	.9704	**.9717**	97.10	**97.23**
Shuttle	**99.72**	99.22	.7650	**.9648**	93.14	**97.17**
Splice	64.56	**72.10**	.3787	**.7575**	51.44	**78.68**
Tae	56.30	**57.63**	.5513	**.5649**	56.24	**57.51**
Thyroid	**92.64**	92.50	.4971	**.5364**	62.44	**66.66**
Vehicle	80.49	**80.61**	.7873	**.7887**	80.71	**80.83**
Vowel	**99.39**	**99.39**	**.9936**	**.9936**	**99.39**	**99.39**
Wine	**98.30**	**98.30**	**.9857**	**.9857**	**98.60**	**98.60**
Yeast	**56.54**	54.18	.0000	**.0954**	**55.37**	55.14
Zoo	84.19	**93.05**	.0000	**.2000**	64.05	**80.00**
Average	79.80	**80.17**	.5902	**.6386**	74.49	**76.63**

Table 13
Wilcoxon tests for SVM_{Puk} as base classifier.

Comparison	Measure	R^+	R^-	Hypothesis	p-Value
SM vs. PE	Accuracy	220.0	186.0	Not rejected	0.710304
	GM	336.5	69.5	Rejected for SM at 95%	0.003502
	AvgAcc	307.0	99.0	Rejected for SM at 95%	0.022264

R^+ are ranks in favor of SM and R^- in favor of PE.

Table 14
Executions times in seconds of the proposed methodology for each base classifier and data-set.

Data-set	#Ex.	#Cl.	3NN	C45	SVM_{Poly}	SVM_{Puk}
Autos	159	6	6.0	16.8	34.6	43.6
Balance	625	3	4.2	3.6	3.8	3.6
Car	1728	4	14.2	6.2	52.4	15.2
Cleveland	297	5	19.4	22.8	34.0	22.6
Contraceptive	1473	3	8.8	8.2	8.4	10.4
Dermatology	358	6	3.6	14.0	60.8	86.2
Ecoli	336	8	23.8	128.4	233.6	262.0
Flare	1066	6	39.4	128.0	206.0	243.8
Glass	214	7	31.6	22.6	67.8	80.8
Hayes-roth	132	3	1.2	0.6	1.2	1.4
Iris	150	3	1.0	0.8	1.0	1.4
Led7digit	500	10	414.4	713.8	887.4	947.6
Lymphography	148	4	1.6	4.2	5.4	8.0
Newthyroid	215	3	1.4	1.2	1.4	2.0
Nursery	1296	5	7.8	72.4	99.0	85.4
Pageblocks	548	5	14.8	10.6	54.6	55.0
Penbased	1100	10	142.0	626.2	1 676.0	102.6
Satimage	643	7	31.6	92.0	198.6	86.8
Segment	2310	7	49.2	241.0	795.8	64.8
Shuttle	2175	5	15.6	32.2	188.4	158.4
Splice	319	3	2.0	1.0	2.0	2.6
Tae	151	3	1.6	1.2	1.2	1.6
Thyroid	720	3	2.6	1.4	4.2	4.4
Vehicle	846	4	11.4	18.0	37.8	3.0
Vowel	990	11	236.8	787.0	2 505.8	114.4
Wine	178	3	1.2	0.8	0.8	1.0
Yeast	1484	10	1 093.6	1 599.2	2 706.8	2 595.2
Zoo	101	7	2.2	1.2	42.0	43.6

configurations of SVMs can be explained by the confidences produced by each configuration. In the case of SVM_{Puk}, the value of C considered makes it to produce too extreme confidence values, which do not allow for greater improvements.

6.4. Analyzing the computational complexity

The execution times (in s) of the proposed methodology[2] are shown in Table 14 for each base classifier and data-set. These running times are the average times needed to train a partition from the stratified 5-fold cross-validation for each data-set. It can be observed that the method is not computationally very expensive, requiring few seconds to learn the parameters when the number of classes is low (despite the number of examples in the data-set, e.g., Car data-set). Otherwise, the computational cost increases with the number of classes but the executions are maintained in reasonable times. Moreover, it is interesting to note that, in each data-set, the better the confidences are, the greater the time needed to be executed is, since the adjustment has to deal with smoother confidence values, which require more training time (e.g., SVM_{Poly}).

7. Discussion and future research lines

We have empirically studied the weakness of OVO strategy to deal with difficult classes and tested our proposal to alleviate this problem enhancing their classification. In this section, we aim to first discuss the main conclusions extracted from this analysis (SubSection 7.1). Then, we put forward the future research lines that have come out from this work, both considering the same framework with decomposition strategies (SubSection 7.2) and considering the difficult classes problem from a different perspective (SubSection 7.3).

7.1. Discussion

From the experimental analysis carried out, the following points are highlighted:

1. The difficult classes problem has been pointed out showing that the statistical differences that are usually found between OVO and the baseline classifiers vanish when GM is considered.
2. AvgAcc is not an appropriate measure to account for difficult classes problem on its own. It can be biased by easy classes achieving high TPRs, unnoticing low TPRs in the most difficult ones.

[2] Experiments were carried out in a Intel Core i7 930 with 4 GB RAM.

3. The proposed methodology is able to properly learn the parameters for the similarity-based aggregation, statistically outperforming the corresponding OVO version of the classifier in terms of GM, which was our main objective. Recall that the differences shown between OVO methods are only due to the aggregation, since the score-matrices are the same. Besides, this methodology is generic, in the sense that it can be used with any base classifier.
4. The proposed fitness function allows us to carry out a global adjustment of the base classifiers considering the GM, the AvgAcc and the concept of margin. This last component is important in order to prevent the search algorithm from overfitting, which could occur due to the usage of the same training set.
5. The GM improvement with respect to the previous OVO aggregation has not been at the expenses of accuracy. Hence, we have shown that the base classifiers can be managed in such a way that different objectives can be obtained, without altering them.
6. It has been shown that there is much margin for improvement in terms of GM and AvgAcc, which could be more important than accuracy in many applications.
7. The classifiers giving the best confidence degrees (such as SVM_{Poly}) have more margin for improvement, since they can provide more information to the classification process. Other confidence estimations, such as those of C4.5 based on the number of instances in the predictions, are not so useful, since in many cases a finite number of values are given. The case of SMV_{Puk} is different, because the configuration (parameter C) used produces too borderline (close to 0 or 1) values, which are not as useful as those given by SVM_{Poly}.

7.2. Future research lines under this framework

After these considerations, some future research lines in the same working direction can be pointed out:

1. The hybridization of this method with others accounting for difficult classes problem, and more specifically for class imbalance problem. It should be studied whether this hybridization could provide further improvements.
2. The presence of the difficult classes problem in other decomposition strategies should be analyzed, i.e., in the ECOC framework, looking at the cases such as OVO, which are more prone to suffer from it, or analyzing which codifications could avoid it.
3. Different aggregations aside from the proposed one should be studied in order to favor difficult classes. Besides, different ways of parametrization of the REFs could be considered.
4. The fitness function and the GA used could be improved in order to further enhance the results obtained, e.g., different optimization techniques should be analyzed.
5. Non-competent classifiers [29,26] are another weak point of OVO strategy. The combination strategies dealing with this problem should be analyzed in the context of difficult classes, and the combination with the proposed solution could be implemented. Similarly, hierarchical models [45] could help in reducing the number of non-competent classifiers and the computational time required for testing in large scale problems.

7.3. Additional future research lines on difficult classes

In addition to the future lines emerged in the framework of decomposition strategies, from our point of view, the difficult classes problem should also be analyzed from a different perspective.

1. New measures accounting for difficult classes should be studied, since a proper performance evaluation of the classifiers is of great importance. In this sense, this problem might be related with quantification [21] and calibration [38,20] problems, which would help to understand the problem.
2. The detection of difficult classes prior to the classifier learning would be helpful for the learning of the classifiers. It would allow one to provide the classifiers with additional information in order to increase the TPRs over the difficult classes. As we have shown, difficult classes are strongly related with data characteristics and hence, its presence could be studied using data-complexity metrics [7].
3. The problem of difficult classes may be strongly related with data-set shift problem (when the training data and the test data do not follow the same distribution) [54,46] in some cases. Hence, difficult classes could be analyzed from this different viewpoint.

These future research lines would clarify the problem of difficult classes and would help in developing new methods to deal with it. Anyway, in this paper all this issues cannot be covered and are out of its scope, but they could serve other researches in new developments in the field.

8. Concluding remarks

In this paper we have dealt with a weak point of OVO strategy, that was not addressed before, which we have identified as the difficult classes problem. We have shown that the improvements usually attributed to OVO are mainly due to its

classification enhancement over the easier classes, whereas difficult ones are not empowered as well. The justification of this work was based on the usage of different performance measures taking into account the individual TPR of each class.

We have proposed a new aggregation methodology based on similarity measures, which generalizes the well-known weighted voting strategy, as a possible solution. This aggregation considers a set of parameters which are able to alter the decision rules from the score matrices. In order to find the optimal values for these parameters, we have proposed a fitness function considering the concepts of margin, GM and AvgAcc which, on a whole, allows one to improve the behavior of OVO scheme from the point of view of the difficult classes. In particular, our new methodology has shown its effectiveness, statistically outperforming the previous aggregations in terms of GM, and in most of the cases without hindering the accuracy.

Finally, we have carried out a thorough discussion on the results obtained for a better understanding of the problem and the solution presented, and we have introduced several research lines for future work in this framework but also in the more generic scenario of difficult classes as an interesting problem in machine learning.

Acknowledgements

This work was partially supported by the Spanish Ministry of Education and Science under Projects TIN2010-15055 and TIN2011-28488 and the Andalusian Research Plans P11-TIC-7765 and P10-TIC-6858.

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. Multiple-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] A. Asuncion, D.J. Newman, UCI Machine Learning Repository, 2007. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[6] R. Barandela, J.S. Sánchez, V. García, E. Rangel, Strategies for learning in class imbalance problems, Pattern Recogn. 36 (3) (2003) 849–851.
[7] M. Basu, T.K. Ho, Data Complexity in Pattern Recognition, Springer, 2006.
[8] G. Beliakov, A. Pradera, T. Calvo, Aggregation Functions: A Guide for Practitioners, first ed., Springer Publishing Company, Incorporated, 2008.
[9] S. Bengio, J. Weston, D. Grangier, Label embedding trees for large multi-class tasks, in: NIPS, 2010.
[10] H. Bustince, E. Barrenechea, M. Pagola, Restricted equivalence functions, Fuzzy Sets Syst. 157 (17) (2006) 2333–2346.
[11] H. Bustince, E. Barrenechea, M. Pagola, Image thresholding using restricted equivalence functions and maximizing the measures of similarity, Fuzzy Sets Syst. 158 (5) (2007) 496–516.
[12] H. Bustince, E. Barrenechea, M. Pagola, Relationship between restricted dissimilarity functions, restricted equivalence functions and normal en-functions: image thresholding invariant, Pattern Recogn. Lett. 29 (4) (2008) 525–536.
[13] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[14] T.G. Dietterich, G. Bakiri, Solving multiclass learning problems via error-correcting output codes, J. Artif. Intell. Res. 2 (1995) 263–286.
[15] S.A. Dudani, The distance-weighted k-nearest-neighbor rule, IEEE Trans. Syst., Man, Cybernet. 6 (4) (1976) 325–327.
[16] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: D.L. Whitley (Ed.), Foundation of Genetic Algorithms 2, Morgan Kaufmann, San Mateo, CA, 1993.
[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] A. Fernández, M.J. del Jesus, F. Herrera, On the 2-tuples based genetic tuning performance for fuzzy rule based classification systems in imbalanced data-sets, Inform. Sci. 180 (8) (2010) 1268–1291.
[19] C. Ferri, J. Hernández-Orallo, R. Modroiu, An experimental comparison of performance measures for classification, Pattern Recogn. Lett. 30 (1) (2009) 27–38.
[20] P.A. Flach, N. Lachiche, Naive Bayesian classification of structured data, Mach. Learn. 57 (3) (2004) 233–269.
[21] G. Forman, Quantifying counts and costs via classification, Data Min. Knowl. Discov. 17 (2) (2008) 164–206.
[22] J. Fürnkranz, Round robin classification, J. Mach. Learn. Res. 2 (2002) 721–747.
[23] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Aggregation Schemes for binarization techniques. Methods' Description, Tech. Rep., Research Group on Soft Computing and Intelligent Information Systems (2011). <http://sci2s.ugr.es/ovo-ova/AggregationMethodsDescription.pdf>.
[24] 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 Recogn. 44 (8) (2011) 1761–1776.
[25] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches, IEEE Trans. Syst., Man, Cybernet., Part C: Appl. Rev. 42 (4) (2012) 463–484.
[26] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, F. Herrera, Dynamic classifier selection for One-vs-One strategy: avoiding non-competent classifiers, Pattern Recogn. 46 (12) (2013) 3412–3424.
[27] M. Galar, J. Fernandez, G. Beliakov, H. Bustince, Interval-valued fuzzy sets applied to stereo matching of color images, IEEE Trans. Image Process. 20 (7) (2011) 1949–1961.
[28] 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.
[29] N. Garcia-Pedrajas, D. Ortiz-Boyer, Improving multiclass pattern recognition by the combination of two strategies, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1001–1006.
[30] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[31] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[32] P. Honeine, Z. Noumir, C. Richard, Multiclass classification machines with the complexity of a single binary classifier, Signal Process. 93 (5) (2013) 1013–1026.
[33] C.W. Hsu, C.J. Lin, A comparison of methods for multiclass support vector machines, IEEE Trans. Neural Netw. 13 (2) (2002) 415–425.
[34] G.-B. Huang, H. Zhou, X. Ding, R. Zhang, Extreme learning machine for regression and multiclass classification, IEEE Trans. Syst., Man, Cybernet., Part B: Cybernet. 42 (2) (2012) 513–529.

[35] E. Hüllermeier, S. Vanderlooy, Combining predictions in pairwise classification: an optimal adaptive voting strategy and its relation to weighted voting, Pattern Recogn. 43 (1) (2010) 128–142.
[36] N. Japkowicz, M. Shah, Evaluating Learning Algorithms: A Classification Perspective, Cambridge University Press, 2011.
[37] S. Knerr, L. Personnaz, G. Dreyfus, Single-layer learning revisited: a stepwise procedure for building and training a neural network, in: F. Fogelman Soulié, J. Hérault (Eds.), Neurocomputing: Algorithms, Architectures and Applications, vol. F68 of NATO ASI Series, Springer-Verlag, 1990, pp. 41–50.
[38] N. Lachiche, P.A. Flach, Improving accuracy and cost of two-class and multi-class probabilistic classifiers using ROC curves, in: Proc. 20th International Conference on Machine Learning (ICML'03), AAAI Press, 2003.
[39] 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.
[40] L. Liu, P. Fieguth, Texture classification from random features, IEEE Trans. Pattern Anal. Mach. Intell. 34 (2012) 574–586.
[41] Y. Liu, Fisher consistency of multicategory support vector machines, in: 11th Int. Conf. Artificial Intelligence and Statistics (AISTAT'07), 2007.
[42] 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.
[43] M. Lozano, F. Herrera, N. Krasnogor, D. Molina, Real-coded memetic algorithms with crossover hill-climbing, Evol. Comput. 12 (2004) 273–302.
[44] H.H. Malik, D. Fradkin, F. Moerchen, Single pass text classification by direct feature weighting, Knowl. Inform. Syst. 28 (2011) 79–98.
[45] E. Montañés, J. Barranquero, J. Díez, J.J. del Coz, Enhancing directed binary trees for multi-class classification, Inform. Sci. 223 (2013) 42–55.
[46] J.G. Moreno-Torres, T. Raeder, R. Alaiz-Rodríguez, N.V. Chawla, F. Herrera, A unifying view on dataset shift in classification, Pattern Recogn. 45 (1) (2012) 521–530.
[47] T.K. Paul, H. Iba, Prediction of cancer class with majority voting genetic programming classifier using gene expression data, IEEE/ACM Trans. Comput. Biol. Bioinformatics 6 (2) (2009) 353–367.
[48] J.C. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimization, MIT Press, Cambridge, MA, USA, 1999.
[49] J.C. Platt, Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods, in: Advances in Large Margin Classifiers, MIT Press, 1999.
[50] F. Provost, P. Domingos, Tree induction for probability-based ranking, Mach. Learn. 52 (3) (2003) 199–215.
[51] F. Provost, T. Fawcett, Robust classification for imprecise environments, Mach. Learn. 42 (2001) 203–231.
[52] O. Pujol, P. Radeva, J. Vitria, Discriminant ECOC: a heuristic method for application dependent design of error correcting output codes, IEEE Trans. Pattern Anal. Mach. Intell. 28 (6) (2006) 1007–1012.
[53] J.R. Quinlan, C4.5: Programs for Machine Learning, first ed., Morgan Kaufmann Publishers, San Mateo-California, 1993.
[54] J. Quionero-Candela, M. Sugiyama, A. Schwaighofer, N.D. Lawrence, Dataset Shift in Machine Learning, The MIT Press, 2009.
[55] R. Rifkin, A. Klautau, In defense of one-vs-all classification, J. Mach. Learn. Res. 5 (2004) 101–141.
[56] 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, Knowledge and Information Systems 38 (1) (2014) 179–206.
[57] J. Sanz, A. Fernández, H. Bustince, F. Herrera, A genetic tuning to improve the performance of fuzzy rule-based classification systems with interval-valued fuzzy sets: degree of ignorance and lateral position, Int. J. Approx. Reason. 52 (6) (2011) 751–766.
[58] M. Shah, M. Marchand, J. Corbeil, Feature selection with conjunctions of decision stumps and learning from microarray data, IEEE Trans. Pattern Anal. Mach. Intell. 34 (1) (2012) 174–186.
[59] A. Tewari, P.L. Bartlett, On the consistency of multiclass classification methods, J. Mach. Learn. Res. 8 (2007) 1007–1025.
[60] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.
[61] F. Wilcoxon, Individual comparisons by ranking methods, Biometrics Bull. 1 (6) (1945) 80–83.
[62] 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.
[63] X. Wu, V. Kumar, J.R. Quinlan, J. Ghosh, Q. Yang, H. Motoda, G.J. McLachlan, A. Ng, B. Liu, P.S. Yu, Z.-H. Zhou, M. Steinbach, D.J. Hand, D. Steinberg, Top 10 algorithms in data mining, Knowl. Inform. Syst. 14 (2007) 1–37.
[64] J. Yang, I.W. Tsang, Hierarchical maximum margin learning for multi-class classification, in: Proc. 27th Conf. Uncertainty in Artificial Intelligence (UAI 2011), 2011.
[65] Z.-H. Zhou, X.-Y. Liu, Training cost-sensitive neural networks with methods addressing the class imbalance problem, IEEE Trans. Knowl. Data Eng. 18 (1) (2006) 63–77.