Pattern Recognition 46 (2013) 3412–3424

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

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

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

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

ARTICLE INFO

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

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

ABSTRACT

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

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

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

1. Introduction

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

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

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

* Corresponding author. Tel.: +34 948 16 60 48; fax: +34 948 16 89 24.
E-mail addresses: mikel.galar@unavarra.es (M. Galar), alberto.fernandez@ujaen.es (A. Fernández), edurne.barrenechea@unavarra.es (E. Barrenechea), bustince@unavarra.es (H. Bustince), herrera@decsai.ugr.es (F. Herrera).

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.04.018

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

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

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

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

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

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

2. Related works: decomposition strategies and dynamic classifier selection

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

2.1. Binarization for multi-classification

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

OVO [37] and OVA [12] decompositions are known to be the most common approaches. Whereas the former consists of learning a binary classifier to discern between each pair of classes, the latter constructs a binary classifier to separate each single class from all other classes. The simplest combination strategy is to consider the voting strategy, where each classifier gives a vote for a class and that with the largest number of votes is given as output (in OVA only one classifier should give a positive vote). In [4],