Knowledge-Based Systems 85 (2015) 298–306

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Dynamic selection of the best base classifier in One versus One

I. Mendialdua [a,*], J.M. Martínez-Otzeta [a], I. Rodriguez-Rodriguez [a], T. Ruiz-Vazquez [b], B. Sierra [a]

[a] Department of Computer Science and Artificial Intelligence, University of the Basque Country UPV/EHU, Donostia-San Sebastian 20018, Spain
[b] Department of Computer Architecture and Technology, University of the Basque Country UPV/EHU, Donostia-San Sebastian 20018, Spain

ARTICLE INFO

Article history:
Received 28 November 2014
Received in revised form 4 May 2015
Accepted 9 May 2015
Available online 19 May 2015

Keywords:
Machine learning
Supervised classification
Decomposition strategies
One against One
Classifier combination
Dynamic classifier selection

ABSTRACT

Class binarization strategies decompose the original multi-class problem into several binary sub-problems. One versus One (OVO) is one of the most popular class binarization techniques, which considers every pair of classes as a different sub-problem. Usually, the same classifier is applied to every sub-problem and then all the outputs are combined by some voting scheme. In this paper we present a novel idea where for each test instance we try to assign the best classifier in each sub-problem of OVO. To do so, we have used two simple Dynamic Classifier Selection (DCS) strategies that have not been yet used in this context. The two DCS strategies use K-NN to obtain the local region of the test-instance, and the classifier that performs the best for those instances in the local region, is selected to classify the new test instance. The difference between the two DCS strategies remains in the weight of the instance. In this paper we have also proposed a novel approach in those DCS strategies. We propose to use the K-Nearest Neighbor Equality (K-NNE) method to obtain the local accuracy. K-NNE is an extension of K-NN in which all the classes are treated independently: the K nearest neighbors belonging to each class are selected. In this way all the classes take part in the final decision. We have carried out an empirical study over several UCI databases, which shows the robustness of our proposal.

1. Introduction

The objective of the supervised classification strategies is to classify the new unlabeled samples in their correct class. To do so, these strategies create a prediction model (also denoted as classifier) based on a training set of well labeled instances.

A classification problem with only two classes is known as a binary classification problem. A simple example of a binary classification problem are the yes/no or true/false problems. On the other hand the problems with more than two classes are known as multi class problems. However for several kind of classifiers, such as SVM, it is easier to build a classifier to distinguish only between two classes. Because of that, two general approaches have been adopted to deal with multi class problems: to create a single decision function that considers all the classes or to decompose the problem into several binary sub-problems (also known as class-binarization).

In the latest years the class-binarization strategies are getting more common in the literature. There are 3 main techniques: One versus All (OVA)[2], One versus One (OVO)[12] and Error Correcting Output Codes (ECOC)[9]. In this work we focus our attention on OVO strategy, which compares the cases belonging to two classes in each sub-problem; the remaining classes are ignored in each sub-problem.

OVO gives the option to consider each sub-problem as independent and to select a different base classifier in each sub-problem, which could be considered as an example of static classifier selection problem. For classification selection scheme two categories exist: static and dynamic. In the first case, regions of competence are defined during the training phase, while in the second case, they are defined during the classification phase taking into account the characteristics of the sample to be classified.

In the literature it is possible to find several works that propose the selection of different base classifiers in each sub-problem statically; however conclusions of these works are contradictory: some works obtain significant improvements, while others reject this hypothesis.

In this paper, we propose to extend this idea trying to assign dynamically the best base classifier in each sub-problem of OVO. We have called to this new approach DYNOVO. We present several variations of DYNOVO using two simple Dynamic Classifier Selection (DCS) strategies from the state-of-the-art. Those strategies select the classifier that obtains the best accuracy in a local region, which is defined by the K-Nearest Neighbor (K-NN) algorithm. In order to adapt those DCS strategies we have made several changes

* Corresponding author at: Robotics and Autonomous System Research Group, University of the Basque Country UPV/EHU, Donostia-San Sebastian 20018, Spain.

http://dx.doi.org/10.1016/j.knosys.2015.05.015
0950-7051/

on the K-NN algorithm, moreover we propose the use of another K-NN version called K-Nearest Neighbor Equality (K-NNE) from the state-of-the-art which fits properly in this problem. For our experiments we have chosen several well-known classifier from the machine learning paradigms: SVM, C4.5, Ripper, Naive Bayes and Bayesian Network. We have carried out our experiments over 22 UCI databases. Experimental results show that DYNOVO obtains very good results.

The rest of the paper is organized as follows: In Section 2 we review the class binarization techniques, focusing on OVO strategy. In Section 3 we review the Dynamic Classifier Selection technique while Section 4 is devoted to related work. Section 5 describes the proposed approach and Section 6 shows the experimental results obtained. Finally, Section 7 states the conclusions of our work and future research lines.

2. Class binarization

Several machine learning techniques, such as SVM, were designed to solve two-class problems. However many real-word problems involve the discrimination of more than two classes. In order to use those algorithms in multi-class problem the class binarization strategies divide the original problem into several two-class problems. It has been proven the benefits to use the binarization techniques in multi-class problems [15] and due to those promising results the use of these strategies has been extended to other base classifiers, such as Ripper [14] or C4.5 [9]. In the recent years the class binarization strategies are receiving more attention in the literature, and one indicative of that is that recently several reviews have been published [29,18,15].

The class binarization techniques are divided by two steps: decomposition and combination.

In the decomposition step, the multi-class problem is decomposed into several binary sub-problems. The most popular strategies consist on grouping classes into two groups in each sub-problem, in this way each binary classifier compares two groups of classes between them. The code-matrix is an easy way to represent how the classes are grouped.

In the code matrix each class takes values in the set of {+1, −1, 0}, where +1 indicates that the class is associated to the positive class, −1 indicates that the class is associated to the negative class and 0 indicates that the class is ignored in this binary sub-problem. In Fig. 1 an example of a code matrix can be seen; it shows how a 5-class problem $\{\theta_1, \theta_2, \theta_3, \theta_4, \theta_5\}$ is decomposed into a 6 binary sub-problems $\{f_1, f_2, f_3, f_4, f_5, f_6\}$. For instance, it can be seen that in the sub-problem f_1, the classifier is constructed in such manner that the cases belonging to θ_1 and θ_2 are grouped in class +1 and the cases of θ_3 and θ_5 in class −1. So this classifier compares θ_1 and θ_2 classes with θ_3 and θ_5, whereas the cases that belong to θ_4 are ignored.

Each of these sub-problems returns an output with a prediction. The combination step consists on combining these predictions to made the final decision. The simplest combination is the majority vote, where each sub-problem returns a vote and the class with the largest number of votes is predicted.

$$
\begin{array}{c}
 \\
 \\
classes
\end{array}
\left\{
\begin{array}{c}
\overbrace{}^{classifiers} \\
f_1\ \ f_2\ \ f_3\ \ f_4\ \ f_5\ \ f_6 \\
\begin{array}{c}\theta_1 \\ \theta_2 \\ \theta_3 \\ \theta_4 \\ \theta_5\end{array}
\left(\begin{array}{cccccc}
+1 & 0 & -1 & -1 & 0 & +1 \\
+1 & +1 & -1 & -1 & +1 & 0 \\
-1 & +1 & +1 & -1 & 0 & 0 \\
0 & -1 & 0 & +1 & 0 & +1 \\
-1 & -1 & 0 & -1 & -1 & -1
\end{array}\right)
\end{array}
\right.
\qquad
\begin{array}{l}
f_1 \rightarrow \theta_1, \theta_2\ vs\ \theta_3, \theta_5 \\
f_2 \rightarrow \theta_2, \theta_3\ vs\ \theta_4, \theta_5 \\
f_3 \rightarrow \theta_3\ vs\ \theta_1, \theta_2 \\
f_4 \rightarrow \theta_4\ vs\ \theta_1, \theta_2, \theta_3, \theta_5 \\
f_5 \rightarrow \theta_2\ vs\ \theta_5 \\
f_6 \rightarrow \theta_1, \theta_4\ vs\ \theta_5
\end{array}
$$

Fig. 1. Example of a code matrix.

Different decomposition strategies have been developed where One Vs One (OVO) is one of the strategies that has received more attention in the literature.

2.1. One versus One (OVO)

OVO decomposition scheme decomposes a K class multiclass problem into a $K(K-1)/2$ sub-problems. Each sub-problem is responsible to differentiate one pair of classes (θ_i, θ_j), where $\theta_i \neq \theta_j$; the remaining classes are ignored.

Fig. 2 illustrates a code matrix of how a 5-class problem is decomposed in OVO: in each sub-problem one class is represented as +1 class, another one is represented as −1 and the remaining classes are represented as 0.

There are different aggregations of combining the output predictions of the sub-problems. The simplest combination strategy is the majority vote [14,12]. An immediate extension is the Weighted Voting, where the vote of each output is weighted based on the confidence level returned by the classifier [22]. Hastie and Tibshirani [21] propose another combination that tries to find the best approximation of the class posterior probabilities given the posterior probabilities of the pairwise sub-problems.

Although OVO requires a high number of sub-problems (specially when the number of classes is high), it is worth mentioning that each classifier is trained only with the samples from the corresponding pair of classes, hence the decision boundaries to distinguish the classes are simpler and the required time is not high. However there are several proposals that try to reduce the number of sub-problems, where most of these works are based on a hierarchical structure [32,11].

3. Dynamic Classifier Selection (DCS)

As different classifiers usually make different error on different samples, Dynamic Classifier Selection (DCS) based methods attempt to predict the single classifier which is most likely to be correct for a given sample. To do so, the best classifier for each partition is determined on a validation process. For classification, an unknown sample is assigned to a partition, and the output of the best classifier for that partition is the one used to make the final decision.

The first dynamic classification approaches are introduced by Woods [39] and are based on K-NN algorithm. He proposes two methods: Overall Local Accuracy (OLA) and local class accuracy: both methods obtain the classifiers' accuracy in local regions in the surroundings of the unknown test sample, the classifier with the best accuracy is selected to classify the unknown sample. Smith [36] proposes an immediate extension of OLA applying the Distance Weighted K-NN (DW-OLA). Giacinto and Roli [19] also extend Woods's work incorporating distance weighted and classifiers confidence levels to two new methods called A Priori and A Posteriori. On the other hand, there are also other works which are not based on the K-NN method, for instance, Liu and Yuan [28] propose to use clustering: they divide the feature space into several clusters for each base classifier. The unknown sample is assigned to a cluster for each base classifier, and the classifier of the most accurate cluster is selected to classify the unknown sample.

$$
\begin{pmatrix}
+1 & +1 & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1 & 0 & 0 & 0 & +1 & +1 & +1 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & -1 & 0 & 0 & +1 & +1 & 0 \\
0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & 0 & +1 \\
0 & 0 & 0 & -1 & 0 & 0 & -1 & 0 & -1 & -1
\end{pmatrix}
$$

Fig. 2. OVO code-matrix.