Pattern Recognition 46 (2013) 2830–2839

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

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

A genetic-based subspace analysis method for improving Error-Correcting Output Coding

Mohammad Ali Bagheri [a,*], Qigang Gao [a,1], Sergio Escalera [b,c]

[a] Faculty of Computer Science, Dalhousie University, 6050 University Avenue, PO BOX 15000, Halifax, Canada NS B3H4R2
[b] Centre de Visio per Computador, Campus UAB, Edifici O, Bellaterra, 08193 Barcelona, Spain
[c] Department of Mathematics, University of Barcelona, Gran Via de les Corts 585, 08007 Barcelona, Spain

ARTICLE INFO

Article history:
Received 17 March 2012
Received in revised form
27 February 2013
Accepted 16 March 2013
Available online 27 March 2013

Keywords:
Error Correcting Output Codes
Evolutionary computation
Multiclass classification
Feature subspace
Ensemble classification

ABSTRACT

Two key factors affecting the performance of Error Correcting Output Codes (ECOC) in multiclass classification problems are the independence of binary classifiers and the problem-dependent coding design. In this paper, we propose an evolutionary algorithm-based approach to the design of an application-dependent codematrix in the ECOC framework. The central idea of this work is to design a three-dimensional codematrix, where the third dimension is the feature space of the problem domain. In order to do that, we consider the feature space in the design process of the codematrix with the aim of improving the independence and accuracy of binary classifiers. The proposed method takes advantage of some basic concepts of ensemble classification, such as diversity of classifiers, and also benefits from the evolutionary approach for optimizing the three-dimensional codematrix, taking into account the problem domain. We provide a set of experimental results using a set of benchmark datasets from the UCI Machine Learning Repository, as well as two real multiclass Computer Vision problems. Both sets of experiments are conducted using two different base learners: Neural Networks and Decision Trees. The results show that the proposed method increases the classification accuracy in comparison with the state-of-the-art ECOC coding techniques.

1. Introduction

A common task in many real-world pattern recognition problems is to discriminate among instances that belong to multiple classes, known as *multiclass classification*. There are two general approaches to deal with multiclass problems. One approach is to construct a single decision function by considering all classes concurrently and to solve a complex classification problem, known as the single-machine approach [1,2]. Some classification algorithms, such as the k-Nearest Neighbor (kNN) or Multilayer Perceptron (MLP), are inherently based on this approach. The second approach is to recast the multiclass problem into a series of smaller binary classification problems, which is referred to as "class binarization" [3]. In this way, two-class problems can be solved by binary classifiers and the results can then be combined so as to provide a solution to the original multiclass problem. An extensive comparison of the results demonstrates that the class binarization approach generally achieves a better performance, even for powerful learners [3,4]. In addition, many established classification algorithms are specifically designed for binary problems, such as Support Vector Machine (SVM) or AdaBoost. Therefore, to solve multiclass classification problems using these binary classifiers, the class binarization approach should be employed.

Among the proposed methods for approaching class binarization, three techniques are well-known: one-versus-all (OVA) [5], one-versus-one (OVO) [6], and Error Correcting Output Codes [7,8]. In one-versus-all, the multiclass problem is decomposed into several binary problems in the following way: for each class a binary classifier is trained to discriminate among the patterns of the class and the patterns of the remaining classes. In the one-versus-one technique, one classifier is trained to separate each possible pair of classes. In both approaches, the final classification prediction is usually obtained by means of a voting or committee procedure. More recently, a unified framework was introduced to decompose a multiclas problem into a series of different binary problems, which is known as Error Correcting Output Codes (ECOC). In this framework, each classifier is trained on a two meta-class problem, where each meta-class consists of some combinations of the original classes. The ECOC method can be broken down into two stages: encoding and decoding. The aim of the encoding stage is to design a discrete decomposition matrix (codematrix) for the given problem. Each row of the codematrix,

* Corresponding author. Tel.: +1 902 579 9789; fax: +1 902 492 1517.
E-mail addresses: bagheri@cs.dal.ca (M.A. Bagheri), qggao@cs.dal.ca (Q. Gao), sergio@maia.ub.es (S. Escalera).
[1] Tel.: +1 902 494 3356; fax: 1 902 492 1517.

0031-3203/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.patcog.2013.03.014

named *codeword*, is a sequence of bits representing each class, where each bit identifies the membership of the class to a classifier [9]. In the decoding stage, the final classification decision is obtained based on the outputs of binary classifiers. Given an unlabeled test sample, each binary classifier casts a vote for one of the two meta-classes used in its training. The output vector is compared to each class codeword of the matrix and the test sample is assigned to the class whose codeword is closest to the output vector, according to a distance measure. Because of the ability of the ECOC framework to correct the bias and variance errors of the base classifiers [10–12], it has been successfully applied to a wide range of applications [13–16].

The priority when designing ECOC matrices is to improve the error correcting capability of the codematrix, mainly by maximizing a separability criterion between any pair of rows and/or any pair of columns. In general, optimizing row separation criteria directly leads to more error-correcting capability. According to error-correcting theory, it can easily be shown that a matrix having d bits error-correcting capability implies that there is a minimum Hamming distance of $2d+1$ between any pair of rows (codewords). Assuming that each codebit is transmitted independently, it is then possible to correctly classify a received test codeword having fewer than d bits in error, by assigning that codeword to the closest codeword based on the Hamming distance. Therefore, it is desirable to design a codematrix with a high minimum Hamming distance between any pair of codewords. However, the capability to detect and possibly correct errors is "*dependent on the assumption that each error is independently produced*" [17,12]. Therefore, the independence of binary classifiers is the cornerstone of the design of ECOC matrices, without which the ECOC method would be ineffective. The intuition is that if each binary classifier makes different errors, then the ECOC's ability to detect and possibly correct errors would be improved.

The conventional strategy in the ECOC literature for designing independent classifiers is to optimize the distance between ECOC dichotomizers. This property is generally achieved by maximizing the Hamming distance between each column and the others, including their complementaries. Several methods have been proposed that aim to simultaneously optimize row and column separation, such as BCH coding [18], CHC coding [19], and evolutionary techniques [20]. Interestingly, the extensive experimental results show that the codes designed using only a row separation criterion almost performed as well as codes designed using column and row separation. However, codes designed using only column separation criteria performed significantly worse [3]. In addition, many researchers agree that a pseudo-random generation of a codematrix is a reasonably good method, and that "*more sophisticated methods might have only marginal effect on testing error*" [7,3,21]. These results reveal that conventional strategies to design a codematrix will not promote independence among binary classifiers.

One efficient approach to increase diversity among an ensemble of classifiers is to train each learner with data that consist of different feature subsets, leading to uncorrelated errors by base learners [22]. This idea, usually called *subspace* approach, can effectively make use of the diversity of base learners to reduce the variance as well as the bias errors [23,24]. Inspired by this idea, we design a new method for the ECOC framework, named Subspace ECOC. The strategy consists of using different feature subsets for each dichotomizer, leading to more independent classifiers and, consequently, increasing the overall system accuracy. In addition to the design of more independent classifiers, the new technique allows for the design of larger codes in comparison to classical methods.

Some previous studies have proposed the use of bagging and boosting within the ECOC framework, mainly by selecting a sampling of data for each dichotomizer in order to increase the diversity of binary problems. In this sense, Schapire proposed a new technique by combining the boosting algorithm with the idea of output codes [21]. Similarly, Windeatt and Ardeshir proposed to combine the AdaBoost, a version of boosting, with output coding using the decision tree as a base learner [25]. Although the previous methods have performed sampling of data, they used the same set of available features, so it is likely that some classification errors will be common, arising from noisy or non-discriminant features. To our knowledge, there is no related work that performs feature selection within the ECOC framework independently of the base classifier.

Another relevant factor for a codematrix to achieve a good performance is the problem-dependent design of the codes. That is, for a given problem we need to take into account the characteristics of the problem at hand. While the most previous work tried to design a generic codematrix for any classification problem, few studies tried to develop the codematrix by considering the problem characteristics or the classification performance. In this paper, we attempt to tackle this issue using an evolutionary algorithm-based optimization approach in order to guide the feature selection of the three-dimensional ECOC matrix for each problem. More specifically, the Genetic Algorithm (GA) is employed, which has been shown to provide an efficient trade-off between the quality of the solution and the search complexity. In this way, the efficiency of the whole ensemble for the problem at hand is considered in the optimization process of the Genetic Algorithm. As a result, our problem-dependent coding design in the ECOC framework based on the feature subspace approach not only provides more independent classifiers, but also increases the overall classification accuracy.

The rest of this paper is organized as follows. Section 2 provides a brief introduction to the ECOC framework. The proposed method based on the feature subspace is explained in detail in Section 3. Sections 4 and 5 report the experiments we performed with data from two different environments: benchmark and image vision datasets. Finally, Section 6 draws the main conclusions of the paper.

2. Error Correcting Output Codes

First, we briefly describe some notations used in this paper:

- $T=\{(\mathbf{x_1},y_1),(\mathbf{x_2},y_2),\ldots,(\mathbf{x_m},y_m)\}$: A training set, where $\mathbf{x_i}\in R^n$; and each label, y_i, is an integer belonging to $Y=\{1,2,\ldots,N_c\}$, where N_c is the number of classes.
- $h=\{h_1,h_2,\ldots,h_L\}$: A set of L binary classifiers.

2.1. ECOC overview

The basis of the ECOC framework consists of designing a codeword for each of the classes. This method uses a matrix M of $\{1,-1\}$ values of size $N_c \times L$, where L is the number of codewords codifying each class. This matrix is interpreted as a set of L binary learning problems, one for each column. That is, each column corresponds to a binary classifier, called *dichotomizer* h_j, which separates the set of classes into two metaclasses. Instance $\mathbf{x}$, belonging to class i, is a positive instance for the jth classifier if and only if $M_{ij}=1$ and is a negative instance if and only if $M_{ij}=-1$. Table 1 shows a possible binary coding matrix for a 4-class problem $\{c_1,\ldots,c_4\}$ with respective codewords $\{M(\text{r},.)\}$ that uses six dichotomizers $\{h_1,\ldots,h_6\}$. In this table, each column is associated with a dichotomy classifier, h_j, and each row is a unique codeword that is associated with an individual target class. The