



A genetic encoding approach for learning methods for combining classifiers

Loris Nanni *, Alessandra Lumini

DEIS, IEIIT – CNR, Università di Bologna, Viale Risorgimento 2, 40136 Bologna, Italy

ARTICLE INFO

Keywords:

Ensemble of classifiers
Learning-based fusion
Correspondence analysis
Genetic algorithm

ABSTRACT

Several studies have reported that the ensemble of classifiers can improve the performance of a stand-alone classifier. In this paper, we propose a learning method for combining the predictions of a set of classifiers.

The method described in this paper uses a genetic-based version of the correspondence analysis for combining classifiers. The correspondence analysis is based on the orthonormal representation of the labels assigned to the patterns by a pool of classifiers. In this paper instead of the orthonormal representation we use a pool of representations obtained by a genetic algorithm. Each single representation is used to train a different classifiers, these classifiers are combined by vote rule.

The performance improvement with respect to other learning-based fusion methods is validated through experiments with several benchmark datasets.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

The ensemble of classifiers have been intensively studied in the literature, both theoretical and empirical studies have demonstrated that a good ensemble can improve the performance of a stand-alone classifier, in particular if the individual classifiers in the ensemble are both accurate and independent. Several different approaches have been proposed in the literature for combining classifiers (Garcia-Pedrajas, Hervás-Martínez, & Ortiz-Boyer, 2005; Zhou, Wu, & Tang, 2002), some examples are

- weighted average (Cooke, 1991);
- Bayesian fusion (Cooke, 1991);
- majority vote (Chen & Cheng, 2001; Kittler & Alkoot, 2003; Lad, 1996);
- fuzzy logic, possibility theory (Dubois, Grabisch, Prade, & Smets, 1999; Klir & Folger, 1988);
- correspondence analysis (Merz, 1999);
- maximum entropy modelling (Levy & Deli, 1994; Myung, Ramamoorti, Andrew, & Bailey, 1996).

The method named Stacking (Dzeroski & Zenko, 2004; Seewald, 2002) was one of the first learning method for combining multiple classifiers. In this method a classifier (named meta-level classifier) is trained using the outputs of the “base-level” classifiers. For example, in Ting and Witten (1999) the meta-level classifier is trained using the probabilities of each of the class values returned by each of the base level classifiers. In (Todorovski & Dzeroski,

2000, 2002) the stacking approaches based on meta decision trees have been studied.

In Merz (1999) the ideas of stacking and correspondence analysis are combined for classifiers fusion, in this method the orthonormal representations of the labels assigned to the test patterns by the set of classifiers are projected onto an uncorrelated space, finally the projected patterns are classified by a nearest neighbour classifier.

The main problem of the learning methods for combining classifiers (Kuncheva, Bezdek, & Duin, 2001) is that given insufficient training samples simple fusion functions may outperform trained fusion functions (Duin, 2002; Raudys, 2003; Raudys & Janeliunas, 2002).

The most typical approaches for building a pool of classifiers are the followings (Zhou & Yu, 2005):

- Bagging (Breiman, 1996), each classifier is trained using a different training set;
- Random subspace (Ho, 1998), each classifier is trained using a different feature set;
- Rotation forest (Rodríguez, Kuncheva, & Alonso, 2006), the main idea consists in divide the feature set in subsets of a given dimension and to apply the principal component analysis (PCA) on the features that belong to a given subset. Finally these projections are combined to build a projection matrix that build a modified training set;
- Each classifier has different values for its parameters (Altuncay, 2006).

Also the evolutionary methods (Yu & Cho, 2006) are studied for building the ensemble of classifiers. In Lardi (1994) and Siedlecki

* Corresponding author.

E-mail address: lnanni@deis.unibo.it (L. Nanni).

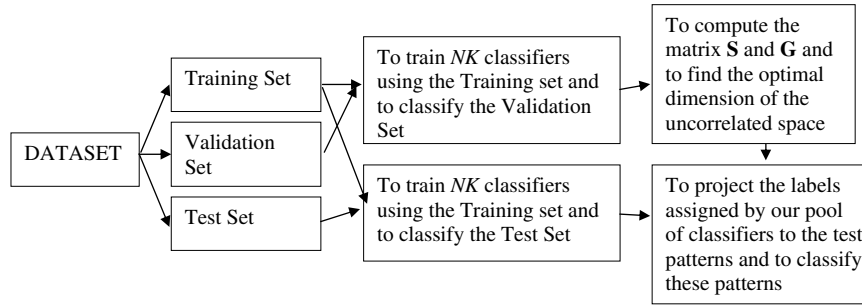


Fig. 1. Correspondence analysis for combining classifiers.

and Sklansky (1989) the genetic algorithms (GA) are used for creating a pool of classifiers where each classifier is trained using a different feature set. In Gabrys and Ruta (2006) the genetic algorithm is used for the simultaneous selection of feature subspaces, classifier prototypes and combination rule. In Altıncay (2006) the genetic algorithm is used for creating a pool of classifiers where each classifier has different values for its parameters.

All the previous methods create an ensemble of classifiers perturbing: the feature set (e.g. random subspace and the rotation forest); the training set (e.g. the Bagging); or changing the parameters of the classification method. We try to improve the performance of the standard methods including the perturbation also in the method used to combine the pool of classifiers. In this paper, we combine correspondence analysis (for the full explanation of this method please read Section 2) and genetic algorithm, instead of the orthonormal representation of the label of the classes we use a pool of representations obtained by a genetic algorithm. We train a different classifier for each new representation, finally we combine by vote rule this pool of classifiers. The objective function of GA is the minimization of the error rate on the validation set.

In Section 4.1, with the help of the Q-statistic (Kuncheva, 2004) (that measures the independence among a pool of classifiers) we show why our methods work well.

The paper is organized as follows: in Section 2 the correspondence analysis to combine classifiers is explained, in Section 3 the new technique is reported, in Section 4 experimental results are presented. Finally, in Section 5 some concluding remarks are given.

2. Correspondence analysis to combine classifiers

In this section we briefly explain the standard correspondence analysis (CA) for combining NK stand-alone classifiers. We named as \mathbf{N} the indicator matrix, this matrix contains for each pattern the prediction of each single classifier and the truth label. An orthonormal representation is used for each class. For example, given a pattern \mathbf{x} (in a 3-class problem), where the true label is 2, assigned by three classifiers to the class 1, 2 and 3; the row in the matrix \mathbf{N} of the pattern \mathbf{x} is given by: 0 0 1 | 0 1 0 | 1 0 0 | 0 1 0 (the orthonormal representation of the classes 1, 2, 3 and 2). We define row and column masses (\mathbf{r} and \mathbf{c}) the values obtained summing each row/column in \mathbf{N} and dividing by: (number of the validation patterns) \times (number of classes + 1). Starting from \mathbf{r} and \mathbf{c} two diagonal matrices are build: \mathbf{D}_r and \mathbf{D}_c .

The standardized residual matrix is defined as,

$$\mathbf{A} = \mathbf{D}_r^{-1/2} (\mathbf{P} - \mathbf{r}\mathbf{c}^T) \mathbf{D}_c^{-1/2}$$

The probability matrix \mathbf{P} is obtained dividing \mathbf{N} by: (number of the validation patterns) \times (number of classes + 1). Now the singular value decomposition of \mathbf{A} is calculated:

$$\mathbf{A} = \mathbf{U}\mathbf{S}\mathbf{V}^T$$

Now the principal coordinates of the rows and of the columns are derived:

$$\mathbf{F} = \mathbf{D}_r^{-1/2} \mathbf{U}\mathbf{S} \quad \mathbf{G} = \mathbf{D}_c^{-1/2} \mathbf{V}\mathbf{S}$$

Given a test example we calculate a set of indicator vectors \mathbf{IV} . The i -th indicator vector \mathbf{IV}_i is build concatenating the predictions of each single classifier and the i -th class, we have an indicator vector for each class of the problem. The labels of the classes are codified using the orthonormal representation. We define \mathbf{f}_i the projection of \mathbf{IV}_i in the uncorrelated space, before the projection \mathbf{IV}_i is normalized dividing it by (number of classes + 1):

$$\mathbf{f}_i = \mathbf{IV}_i * \mathbf{G} * (\text{pinv}(\mathbf{S}))$$

For each class i we calculate the distance between \mathbf{f}_i and the row of \mathbf{G} that represents the class i (see Merz, 1999 for details), the pattern is assigned to the class for which the distance is minimum. The distance between \mathbf{f}_i and the row of \mathbf{G} is calculated using the first d dimension of these vectors. Using the validation set we find the optimal value for d . In Fig. 1 correspondence analysis to combine classifiers is detailed.

3. A genetic version of correspondence analysis

Genetic Algorithms¹ operate iteratively on a population of chromosomes. A randomly generated set of such strings forms the initial population from which the GA starts its search. Three basic genetic operators guide this search: selection, crossover and mutation.

Selection: Our selection strategy was cross generational. Assuming a population of size N ($N = 25$ in this paper), the offspring double the size of the population and we select the best N individuals from the combined parent-offspring population. The selection is performed using as fitness function the minimization of the error rate in the validation set.

Crossover: Uniform crossover is used here. The crossover probability used in our experiments was 0.96.

Mutation: The mutation probability used here was 0.02.

In our genetic version of correspondence analysis we use the genetic algorithm for finding a set of representations alternative to the orthonormal representation for combining NK stand-alone classifiers. Each class is codified with random values between 0 and 1, for example in a 3-class problem the orthonormal representation of the class 2 is given by 0 1 0, in our method the representation of the class 2 is given by 0 x 0 where x is a gene of a chromosome. The fitness function is the minimization of the error rate in the validation set.

¹ We have used the GAOT matlab toolbox (Houck, Joines, & Kay, 1995).

Download English Version:

<https://daneshyari.com/en/article/387187>

Download Persian Version:

<https://daneshyari.com/article/387187>

[Daneshyari.com](https://daneshyari.com)