A New Classifiers Ensemble Method for Handwritten Pen Digits Classification

M. Hassanzadeh¹, G. Ardeshir²

¹,². Faculty of electrical and computer engineering, Babol University of Technology, Babol, Iran

Corresponding author email: m.hasanzadeh@nit.ac.ir

ABSTRACT: Recent researches have shown that ensembles of classifiers have more accuracy than a single classifier. Baging, boosting and error correcting output codes (ECOC) are most common ways for creating combination of classifiers. In this paper a new method for ensemble of classifiers has been introduced and performance of this method examined by applying to handwritten pen digits dataset. Experimental results indicate that this method leads to more accurate classification than other existing methods.

Keywords: Accuracy, Bagging, Boosting, ECOC, Ensemble of Classifiers

INTRODUCTION

In recent year, ensemble of classifiers has been known as a method for improving the accuracy of classification. An ensemble (committee) of classifiers is a set of classifiers whose individual decisions are combined in some way (typically by voting) to classify new examples. In literature, the ensemble of classifiers is referred by different names: committees of learners, mixtures of experts, classifier ensembles, multiple classifier systems, consensus theory, etc. (Kuncheva et al., 2003). Hansen and Salamon in 1990 showed that an ensemble of classifiers could be more efficient than a single one if each classifier of the ensemble is different from the others in terms of the classification error (Hansen et al., 1990). This means that one of the main problems in combining classifiers is “creating diverse classifiers”. In the modeling of classifier combination, many researchers believe that the success of classifier ensembles not only depends on a set of appropriate classifiers, but also on the diversity being inherent in the classifiers. Bagging boosting and error correcting output codes are some existing methods for combination of classifiers. In this paper a new method for ensemble of classifiers has been proposed and performance of this method compared with other methods. Handwritten pen digits datasets was used in simulation task.

The rest of the paper organized as follows: Next section describes several existing methods for creating ensemble of classifiers. In another section, proposed method for ensemble of classifiers has been introduced. Results of experiments and conclusion described in another sections.

METHODS OF CLASSIFIERS ENSEMBLE

Using different training sets for each classifier is a method to create an ensemble. In this method, the data available for training is divided into different subsets and each set of data is used as a building block for each classifier. Two important methods of creating these subsets are bagging and boosting (Beriman, 1998. Beriman 1999. Optiz et al., 1999. Rokach, 2009). Since in this work we have used Bagging, it is explained in this section.

Bagging

In this method N elements are selected randomly by replacement from an N-element set known as training set. The selected elements form a new set. It should be noted that all the elements are selected with the same probability and depending to the “replacement”, there is a possibility of one element to be selected multiple times.

The N sets of data are used to build the same number of the classifiers constructing the ensemble. After the training process, each new input data is applied to the trained classifiers and is assigned to a one of the classes by using the majority vote scheme. Figure 1 illustrates the Bagging algorithm.
In 1996, Breiman showed that the Bagging algorithm is more efficient in unstable classifiers in which a small change in the training data results in a considerable variation in classifier. Neural network and decision tree are two examples of unstable classifiers.

Figure 1. Bagging algorithm (Kuncheva, 2004).

**Boosting**

Similar to the Bagging method, different randomly selected sub-sets are used to create the classifiers in an ensemble. However, the probability of selection of the data varies based on the results coming out from the previously created classifiers. In other words, the data that are incorrectly classified by the previous classifiers are more probably selected in the new set. It is performed by considering an appropriate probability distribution for the training data. At the beginning, the specified distribution is uniform and by addition of each classifier to the ensemble, the distribution will change (Kuncheva, 2004). The probability of data selection as well as the method of combination of the classifiers in the ensemble can be done in different ways of which one most common method is described in the next section.

**Ada-Boost**

In this method, the samples are also selected based on the performance of previous classifiers to generate a new subset which used to build a new classifier.

At the beginning, the selection probability of the samples is equal for all of them (uniform distribution). By generating a classifier, the data existing in the training set is applied to the classifiers to change the weight of each sample (selection probability) based on the classifier’s performance. The weights are changed so that the summation of the weights classified incorrectly ($\varepsilon_k$) are added to each other to calculate a factor called beta as (1)

$$\beta_k = \frac{\varepsilon_k}{1 - \varepsilon_k}$$  \hspace{1cm} (1)

The weight of all data classified incorrectly are multiplied by beta and then normalized so that the summation of the weights (which is the selection probability) equals to 1.

In the classification step, by applying the input data to all of the classifiers along with using the majority vote scheme in its weighted mode, the class associated with each data is assigned. The weight considered for kth classifier is equal to $\ln(1/\beta_k)$ (Kuncheva, 2004).

Figures 2 and 3 show the different steps of learning and classification used in this algorithm.

Figure 2. Ada-Boost algorithm, training phase (Kuncheva, 2004).

Classification based on ECOC

In ECOC framework, all base classifiers are trained independently. This training scheme ignores the dataset distribution and the performance of each base classifier (Dijun Luo et al. 2008).

Let $T_r = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ denotes a set of training data where each instance $x_i$ belongs to a domain $X$, each label $y_i$ belongs to a set of labels representing categories $Y$, $Y = \{1, 2, ..., k\}$, and $N$ is the number of instances. A multi-class classifier $H: X \rightarrow Y$ is a function that maps an instance $x$ in $X$ into a label $y$ in $Y$. A typical ECOC method is conducted as follows:

1. **Initialization**: The parameters $\alpha_j$ are initialized by $\alpha_j = \frac{1}{N}$, where $N$ is the number of instances.
2. **Training phase**: For $k = 1, 2, ..., L$:
   - Take a sample $S_k$ from $Z$ using distribution $w^k$.
   - Build a classifier $D_k$ using $S_k$ as the training set.
   - Calculate the weighted ensemble error at step $k$ by $e_k = \frac{1}{N} \sum_{i=1}^{N} w_i^k \cdot \lambda_i$, where $\lambda_i = 1$ if $D_k$ misclassifies $z_i$ and $\lambda_i = 0$ otherwise.
   - If $e_k = 0$ or $e_k \geq 0.5$, ignore $D_k$, reinitialize the weights $w_i^k$ to $\frac{1}{N}$ and continue.
   - Else, calculate $\beta_k = \frac{e_k}{1 - e_k}$, where $e_k \in (0, 0.5)$.
   - Update the individual weights $w_i^{k+1} = \frac{w_i^k \beta_k^{1-e_i}}{\sum_{j=1}^{N} w_j^k \beta_j^{1-e_j}}$, $j = 1, 2, ...$.
3. Return $D$ and $\beta_1, ..., \beta_L$.

Figure 3. Ada-Boost algorithm, classification phase (Kuncheva, 2004).

Classification based on ECOC

In ECOC framework, all base classifiers are trained independently. This training scheme ignores the dataset distribution and the performance of each base classifier (Dijun Luo et al. 2008).

Let $T_r = \{(x_1, y_1), (x_2, y_2), ..., (x_N, y_N)\}$ denotes a set of training data where each instance $x_i$ belongs to a domain $X$, each label $y_i$ belongs to a set of labels representing categories $Y$, $Y = \{1, 2, ..., k\}$, and $N$ is the number of instances. A multi-class classifier $H: X \rightarrow Y$ is a function that maps an instance $x$ in $X$ into a label $y$ in $Y$. A typical ECOC method is conducted as follows:

1. **Initialization**: The parameters $\alpha_j$ are initialized by $\alpha_j = \frac{1}{N}$, where $N$ is the number of instances.
2. **Training phase**: For $k = 1, 2, ..., L$:
   - Take a sample $S_k$ from $Z$ using distribution $w^k$.
   - Build a classifier $D_k$ using $S_k$ as the training set.
   - Calculate the weighted ensemble error at step $k$ by $e_k = \frac{1}{N} \sum_{i=1}^{N} w_i^k \cdot \lambda_i$, where $\lambda_i = 1$ if $D_k$ misclassifies $z_i$ and $\lambda_i = 0$ otherwise.
   - If $e_k = 0$ or $e_k \geq 0.5$, ignore $D_k$, reinitialize the weights $w_i^k$ to $\frac{1}{N}$ and continue.
   - Else, calculate $\beta_k = \frac{e_k}{1 - e_k}$, where $e_k \in (0, 0.5)$.
   - Update the individual weights $w_i^{k+1} = \frac{w_i^k \beta_k^{1-e_i}}{\sum_{j=1}^{N} w_j^k \beta_j^{1-e_j}}$, $j = 1, 2, ...$.
3. Return $D$ and $\beta_1, ..., \beta_L$.
Hsu et. al (2002), such as, comparing each category against the rest, comparing all pairs of categories (one-against-one), employing the random code, and employing the Hadamard code.

2) Base classifier construction: A dichotomy of samples is created for each base classifier. The dichotomies vary according to classifiers. If \( M_{y,s} = -1 \), take all the instances labeled \( y \) as negative samples in training set of the base learner \( h_s \). The instances labeled \( y \) are considered as positive or negative samples in training set of the base learner \( h_s \), if \( M_{y,s} = 1 \) or \( M_{y,s} = -1 \), respectively. If \( M_{y,s} = 0 \), the instances are ignored.

3) Decoding: Given an instance \( x \), a vector of binary labels is generated from all the base classifiers \( \mathbf{H}(x) = (h_1(x), h_2(x), ..., h_n(x)) \). Then the vector is compared with each row of the matrix \( \mathbf{M} \) (each category). A final classification decision is made using the discriminate function as follows:

\[
H(x) = \arg \min_{y \in Y} F(x, y)
\]

\[
F(x, y) = D(M_{y}, \mathbf{H}(x))
\]

where \( D(u, v) \) is distance function between vectors \( u \) and \( v \), and \( M_y \) is the row \( y \) of the code matrix \( \mathbf{M} \). Consequently, the label of \( x \) is predicted to be \( y \) if the output of base classifiers is the ‘closest’ to the row of \( M_y \).

**PROPOSED METHOD**

The main idea for our proposed method is combination of ECOC and other algorithms such as bagging and boosting. Creation of codeword matrix is one of the important steps in classification based on error correcting output codes. There are several methods for creating codeword matrix according to number of classes. When number of classes is greater than 7, random codeword generation is a suitable method. Creating subsets of training set(according to bagging method) is the first step in bagging-ECOC. Then, a codeword matrix is produced randomly for each subset, and finally ensemble of classifiers are created based on error correcting output codes. Required criteria are considered in order to reach the maximum accuracy. Boosting-ECOC follows the same procedure as bagging-ECOC. But the probability of selection of the data varies based on the results coming out from the previously created classifiers.

**EXPERIMENT SETUP AND RESULTS**

The results of the proposed method described in previous section performed on handwritten pen digits dataset, are presented in this part. Handwritten pen digits dataset was used for simulation. This dataset which is presented in UCI machin learning library has 10992 samples. Each sample has 16 features and all samples are classified into 10 classes. 80% of samples were used as training samples and 20% as test samples. We used decision tree (C4.5) and neural networks (with one hidden layer) as base classifiers. Results are presented in figures 4 To 7.

![Figure 4](image_url) - plots of accuracy vs. number of base classifiers for comparing bagging-ECOC against bagging, ECOC and single classifiers. Neural networks used as based classifiers.

![Figure 5](image_url) - plots of accuracy vs. number of base classifiers for comparing boosting-ECOC against boosting, ECOC and single classifiers. Neural networks used as based classifiers.
Comparing with the previous methods, the proposed method gives a more accurate ensemble for classification. This result can be observed clearly in figures 4 to 7. Figure 4 shows plots of accuracy vs. number of base classifiers for comparing bagging-ECOC against bagging, ECOC and single classifiers. Neural networks used as based classifiers. Maximum accuracy within bagging method is 66.46%, ECOC is 76.21% and single classifier is 56.66% but using bagging-ECOC leads the maximum accuracy of 85.72%. Figures 5 to 7 display the same features as figure 4 but for boosting-ECOC and different base classifier. For all of these figures the same trend as figure 4 i.e. the increasing of accuracy can be diagnosed.

CONCLUSIONS

A new method for creating ensemble of classifiers has been proposed and a numerical study was performed to assess the effect of this method on changing the accuracy of classification of handwritten pen digits dataset. This effect was investigated by comparing the results in the form of accuracy vs. number of base classifiers with those of existing methods. The results show that generally, bagging-ECOC and boosting-ECOC increases the accuracy up to 96.11% which is related to the double combination of classifiers.

REFERENCES