American Journal of Scientific Research
ISSN 1450-223X Issue 19(2011), pp. 84-90
© EuroJournals Publishing, Inc. 2011
http://www.eurojournals.com/ajsr.htm

Classifier Ensemble Based Class Weightening

Hamid Parvin
School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran

Hossein Alizadeh
School of Computer Engineering, Iran University of Science and Technology (IUST), Tehran, Iran
E-mail: parvin, halizadeh@iust.ac.ir

Abstract

Many methods have been proposed for combining multiple classifiers in pattern recognition such as Random Forest which uses decision trees for problem solving. In this paper, we propose a weighted vote-based classifier ensemble method. The proposed method is similar to Random Forest method in employing many decision trees and neural networks as classifiers. For evaluating the proposed weighting method, both cases of decision tree and neural network classifiers are applied in experimental results. Main presumption of this method is that the reliability of the prediction of each classifier differs among classes. The proposed ensemble method is tested on a huge Persian data set of handwritten digits and shows improvement in comparison with competitors.

Keywords: Classifier; Classifier Ensembles; Random Forest; Bagging;

1. Introduction

Hybridization of intelligent techniques, coming from different computational intelligence areas, has become popular because of the growing awareness that such combinations frequently perform better than the individual techniques coming from computational intelligence.

Practical experience has indicated that hybrid intelligence techniques might be helpful to solve some of the challenging real world problems. In a hybrid intelligence system, a synergistic combination of multiple techniques is used to build an efficient solution to deal with a particular problem. One field of the hybrid intelligent techniques that has recently been hot for researchers is ensemble algorithms.

Ensemble algorithms train multiple base classifiers and then combine their predictions. Generalization ability of an ensemble could be significantly better than a single classifier for difficult problems [4].

In [12] and [13], the relationship between the ensemble and its components, artificial neural networks (ANN), has been analyzed from the context of both regression and classification, which has revealed that it may be better to ensemble many instead of all of the ANNs at hand. They trained a number of ANNs at first. Then random weights were assigned to those networks and a genetic algorithm (GA) was employed to evolve the weights so that they can characterize to some extent the fitness of the ANNs in constituting an ensemble. Finally some ANNs were selected based on the evolved weights to make up the ensemble.

In contrary, assuming that the reliability of the classifiers differs among classes, an approach based on dynamic selection of the classifiers by taking into account their individual votes, has been proposed in [5]. In particular, a subset of the predictions of each classifier is taken into account during weighted majority voting. Others are considered as unreliable and are not used during combination.

In general, an ensemble is built in two steps: (a) generating multiple base classifiers and then (b) combining their predictions. AdaBoost [8] and Bagging [1] are considered as two famous methods in this field.

AdaBoost sequentially generates a series of base classifiers where the training instances are wrongly predicted by so far trained base classifiers will play more important role in the training of its subsequent classifier. Bagging generates many samples (or bags) from the original training set via bootstrap sampling [7] and then trains a base classifier from each of these samples whose predictions are combined via majority voting. A kind of bagging method is Random Forest, where many decision trees (DT) are trained over distinguished perspectives of training dataset [2].

An ANN has to be configured to be able to produce the desired set of outputs, given an arbitrary set of inputs. Various methods of setting the strength of connections which are considered as ANN learning exist. One way is to set the weights explicitly, using a prior knowledge. Another way which is employed in multi-layer perceptron (MLP) neural network is to 'train' the ANN, feeding it by teaching patterns and then letting it change its weights according to some learning rule [11]. In this paper an MLP neural network is used as classifier.

GA which is considered as one of the optimization paradigms is based on natural processes [3]. A GA can be considered as a composition of three essential elements: first, a set of potential solutions called individuals or chromosomes that will evolve during a number of iterations (generations). This set of solutions is also called population. Second, an evaluation mechanism (fitness function) that allows assessing the quality or fitness of each individual of the population. And third, an evolution procedure that is based on some "genetic" operators such as selection, crossover and mutation. The crossover takes two individuals to produce two new individuals.

The quality of the individuals is assessed with a fitness function. The result is a real value for each individual. The best individuals will survive and are allowed to produce new individuals.

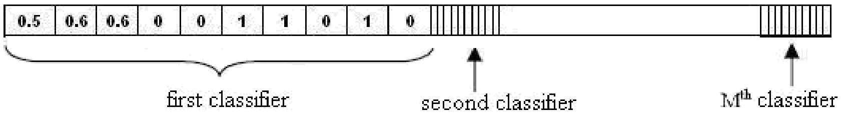
A common and obvious way for classifying an instance is from a sequence of questions, so that next question is asked with regard to this current question. Using trees are the most common representation way for these question-answers. DT is used to create a classifier ensemble, expansively. Also, they are used for the application of data mining and clustering. Their functionality is understandable for human. Besides, unlike other methods such as ANN, they are very quick. It means their learning phase is quicker than other methods [6].

In the next section, we explain the proposed ensemble method in more details.

2. Proposed Method

Let us to assume that total number of obtained classifiers is denoted by M. Let the total number of labels (classes) be denoted by N. The solution of the selection problem encoded in the form of a chromosome which has $N \times M$ genes. First N genes belong to the first classifier. Second subsequent N genes belong to the second classifier. i-th N genes belong to the i-th classifier. The encoding of a chromosome is illustrated in Figure 1, for $N=10$. The genes of each chromosome have real data type. In the exemplary chromosome depicted in Figure 1, the first classifier is not allowed to vote for only fourth, fifth, eighth and tenth classes, and it is allowed to vote for first, second, third, sixth, seventh and ninth classes with coefficients 0.5, 0.6, 0.6, 1, 1 and 1 respectively.

Figure 1: Encoding of a chromosome of used GA, provided the number of classes is *N=10* in the problem



Let us denote a chromosome by *b*, an array of *N*×*M* numbers whose elements belong to closed interval [0,1]. In the Figure 1, *b(i)* is the effect weight of *k*-th classifier to vote for selecting *j*-th class, where *k* is calculated according to the equation 1.

$$k = \lceil i / N \rceil \tag{1}$$

and *j* is calculated according to the equation 2.

$$j = i \mod N \tag{2}$$

Because of non-normalization of the raw *b* chromosome, we first convert it to a normalized version according to the following equation. We denote this normalized version of chromosome *b* by *nb*.

$$nb(b) == \{b \rightarrow nb \mid b \in [0,1]^{N\times M}, nb \in [0,1]^{N\times M}, nb(b)_i = \frac{b_i}{\sum_{q=1}^{M} b_{(q-1)*k+j}}\} \tag{3}$$

where *k* and *j* are the same in the equation 1 and 2. The *nb* is employed in calculating confidence of classifier ensemble for each class per each test data item *x*. These confidences are obtained according to the following equation.

$$\mathrm{conf}(b,x) == (\mathrm{conf}(b,x)_1, \mathrm{conf}(b,x)_2, \ldots, \mathrm{conf}(b,x)_N) \mid b \in [0,1]^c,$$
$$\mathrm{conf}(b,x)_j = \sum_{i=1}^{M} C_{i,j}(x) * \mathrm{nb}(b)_{(i-1)*k+j}\} \tag{4}$$

where *k* and *j* are the same in the equation 1 and 2, *c* is length of chromosome, i.e. *N*×*M*, and $C_{i,j}(x)$ is considered as output of *i*-th classifier for *j*-th class for data item *x*.

Now we define the following terms for the following usage. Normalization of an array of numbers is defined as following equation.

$$\mathrm{normalize}(a) = \{(\mathrm{normalize}(a)_1, \mathrm{normalize}(a)_2, \ldots, \mathrm{normalize}(a)_c) \mid a \in [0,1]^c,$$
$$\mathrm{normalize}(a)_i = \frac{a_i}{\sum_{j=1}^{c} a_j}\} \tag{5}$$

Label is a pre-assigned category of data item ***x***. It is denoted by *l*. $l_i(\boldsymbol{x})$ is a number which is considered as membership of ***x*** to the class *i*. If ***x*** belongs to *i*-th class, $l_i(\boldsymbol{x})$ is 1 and $l_j(\boldsymbol{x})$ is 0 for all $j \neq i$. The fitness of each chromosome (classifier ensemble) is defined as the amount of its accuracy on the evaluation set. The fitness function of a chromosome is computed as equation 6.

$$fitness(b, DV) = \sum_{x \in DV} \| normalize(conf(b,x)) - l(x) \| \tag{6}$$

Where *DV* is validation dataset, *l* is label function. ||.|| is considered as one of norm function like Euclidean distance.

In all experiment, GA parameters are kept fixed. In this study, tournament selection is used for the reproduction phase with tournament size of 5. The crossover operator that has an important role in evolutionary computing, allowing them to explore the problem space by sharing different chromosomes information is set to two-point crossover. The mutation operator, allowing evolutionary computing algorithm to exploiting the problem space, is applied to each entry of the offspring chromosomes with a probability $p_{mut} = 0.01$. Probability of selection for crossover operator is $p_{cross} = 0.8$. In the simulation experiments, the population size is selected as 200. It means that 200 different ensemble candidates evolved simultaneously. Pseudo-code of the GA used in the proposed method for evolving the classifier ensembles is shown in Figure 2.

Figure 2: The GA used in the proposed method

```
Generate randomly an initial population of size POP_SIZE
For each chromosome in the population
    Compute fitness of the chromosome (as it will bementioned below)
For Iteration_Num = 1 .. GENERATION_NUM
    For Chromosome_Num = 1 .. POP_SIZE
        1-Select two parents from the old population
        2-Crossover the two parents to produce two offspring with
        probability $P_{cross}$
        3-Mutate each bit of each offspring with          probability $P_{mut}$
        4-Apply weighted majority to each of the          offspring
        5-Compute fitness of each offspring (as it will be mentioned below)
    End for
    Replace the original population with the offsprings to form the new
    population
End for
Select the best chromosome as the resultant ensemb
```

We use two types of classifier in the ensembles: ANN, DT. The first classifier is ANN with *N* outputs which each of the outputs corresponds to a class.

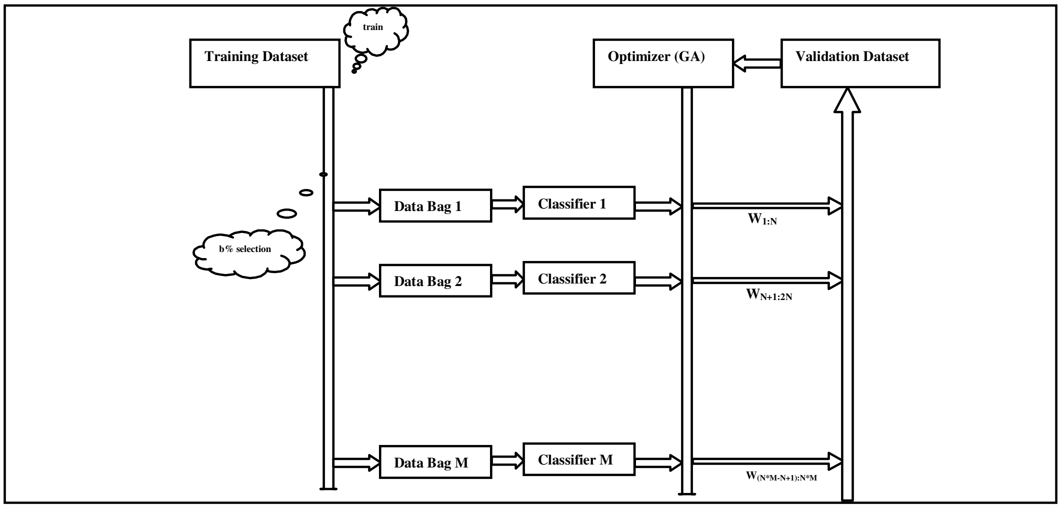
The accuracy of each classifier ensembles is defined as the number of true estimation on the test data set.

In order to evaluate the proposed classifier selection approach, we compare it with weighted and unweighted static classifier selection for Hoda data set. Each chromosome is encoded as a string having *M* entries, one for each classifier with data types real and binary respectively in those two methods. In unweighted static classifier selection which has data type binary, if the value of a gene is 1; this means that the classifier is selected for being used in the corresponding ensemble. It is worthy to note that all the design parameters of the above-mentioned algorithm including population size, number of iterations, crossover and mutation rate etc. are kept fixed. The Figure 3 illustrates the proposed method generally.

3. Experimental Results

Hoda data set [9] is a handwritten OCR data set. This data set contains 100000 data points. Some data instances are depicted in the Figure 4. HODA dataset is the first dataset of handwritten Farsi digits that has been developed during an MSc. project in Tarbiat Modarres University entitled: Recognizing Farsi Digits and Characters in SANJESH Registration Forms. This project has been carried out in cooperation with Hoda System Corporation. It was finished in summer 2005 under supervision of Prof. Ehsanollah Kabir. Samples of the dataset are handwritten characters extracted from about 12000 registration forms of university entrance examination in Iran. The reader is referred to [9] for more details about the data set such as the process of feature generation.

Figure 3: Scheme of the weighted ensemble of classifiers



We have divided our data set into training, validation and test sets containing 60000, 20000 and 20000 data points, respectively. The validation data set acts as pseudo-testing for obtaining fitness of each chromosome as it was explained above. The ensemble is produced by bagging mechanism. Ensemble size is also set to 201. So, the training process is iterated 201 times for performing 201 different base classifiers, ANN and DT. Each classifier is trained over 10% of training dataset. As it is shown in Table 1, the proposed method outperforms other methods and full ensemble. It shows that a full ensemble classification method can be optimized as well.

Figure 4: Some instances of Farsi OCR data set, with different qualities

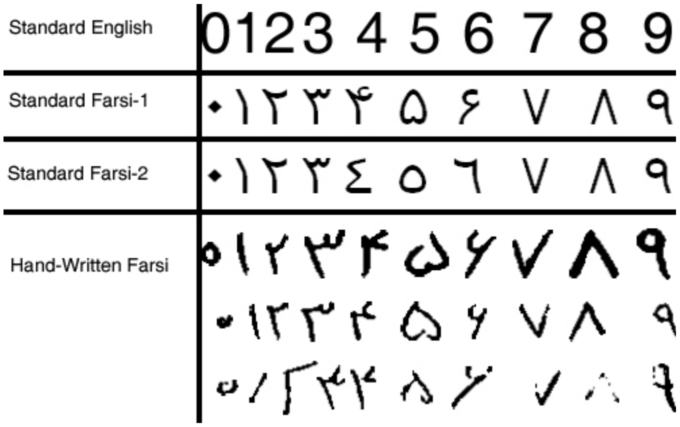


Table 1: The results of proposed ensemble method

Classification Scheme	Accuracy with ANN base classifier	Accuracy with DT base classifier
Unweighted Full Ensemble	98.11	98.22
Unweighted Static Classifier Selection	98.15	98.13
Weighted Static Classifier Selection	98.21	98.34
The Method in [10]	98.27	98.41
The Method in [5] using Soft Error	98.51	98.45
The proposed Method	98.99	99.03

The first row in the Table 1 stands for the accuracy of ensemble of all 201 classifiers without classifier weighting or selection. The majority vote is employed for making final decision in the unweighted full ensemble.

The method in rows 2 in Table 1, unweighted static classifier selection, only focuses on the selected classifiers which are allowed to vote unweightedly based on majority vote mechanism. Indeed this method uses a binary chromosome with the length size which is equaled to the number of classifiers. *i*-th bit of the chromosome stands for being-absent/participating *i*-th classifier. For example, value 1 for *i*-th bit of the chromosome means that the classifier number *i* must participate in the ensemble, else it must be absent in the ensemble and it is prevented for voting.

The method in rows 3 in Table 1, weighted static classifier selection, uses a real chromosome again with the length size which is equaled to the number of classifiers. *i*-th bit of the chromosome stands for amount or weight of being-absent/participating *i*-th classifier.

Because of unbalanced accuracy of classifiers in the ensemble, generally, the static classifier selection can give better results than the simple full ensemble. Usually the weighted approaches are doing better than unweighted ones. However, the results of weighted and unweighted approaches are close to each other, the weighted method slightly outperforms the unweighted one. It improves the result achieved by the full ensemble. Even though, a classifier is not able to achieve good accuracy in all classes; it may obtain a good accuracy on one special class. So, the method which is introduced in [5] has a good result.

It is also worthy to mention that using the absolute error as fitness function rather than equation 6 of the neural networks available in the ensemble, results in less improvement as in the row 4.

Another aspect of the proposed approach is that its computational cost is very low. Although we can train just one MLP to reach to a good accuracy, it consumes many days for large data sets like Hoda. We need to train an MLP for some weeks to reach the accuracy approximately 98% on Hoda data set. The weak learners can converge to a good accuracy very soon, but the subsequent small improvements are very slow. In this approach, we have some weak base classifiers that are under-trained but ensemble of them is not under-trained. We provided 201 individual weak base MLPs or DTs as members in the ensemble. Also it is notable that both ensembles of MLPs and DTs are comparable, with fairly superior of DTs ensemble.

4. Conclusion

Because of their robustness and high performance, classifier ensemble methods are used for difficult problem solving. In this paper, a new ensemble algorithm is proposed, which is designed for building ensembles of bagging classifiers. The proposed method is a weighted vote-based classifier ensemble like Random Forest method which employs DT and ANN as classifiers.

The empirical study on the very large dataset of Persian handwritten digits, Hoda shows that the proposed approach is superior to other combination methods of classifiers, as it is discussed. It effectively improves the accuracy of full ensemble of ANN or DT classifiers.

References

[1] Breiman L.: Bagging predictors. Machine Learning, (1996) vol.24, no.2, 123–140.

[2] Breiman L.: Random forests. MachineLearning, (2001) 45:5–32.

[3] Davis L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York, (1991).

[4] Dietterich T.G.: Ensemble learning. In The Handbook of Brain Theory and Neural Networks, 2nd edition, M.A. Arbib, Ed. Cambridge, MA: MIT Press, (2002).

[5] Dimililer N., Varo˘glu E., Altıncay H.: Vote-Based Classifier Selection for Biomedical NER Using Genetic Algorithms. In Proceedings of the 3nd Iberian Conference on Pattern Recognition and Image Analysis, Girona, Spain, (2007) 202-209.

[6] Duda R.O., Hart P. E., Stork D. G.: Pattern Classification, 2nd ed. John Wiley & Sons, NY (2001).

[7] Efron B., Tibshirani R.: An Introduction to the Bootstrap. New York: Chapman & Hall, (1993).

[8] Freund Y., Schapire R.E.: A decision-theoretic generalization of online learning and an application to boosting. In Proceedings of the 2nd European Conference on Computational Learning Theory, Barcelona, Spain, (1995) 23–37.

[9] Khosravi H., Kabir E.: Introducing a very large dataset of handwritten Farsi digits and a study on the variety of handwriting styles. Pattern Recognition Letters, (2007) vol 28, issue 10, 1133-1141.

[10] Parvin H., Alizadeh H., Minaei-Bidgoli B.: A New Approach to Improve the Vote-Based Classifier Selection. Fourth International Conference on Networked Computing and Advanced Information Management, NCM 2008, (2008).

[11] Sanchez A., Alvarez R., Moctezuma J.C., Sanchez S.: Clustering and Artificial Neural Networks as a Tool to Generate Membership Functions. Proceedings of the 16th IEEE International Conference on Electronics, Communications and Computers (2006).

[12] Zhou Z.H., Wu J.X., Jiang Y., Chen S.F.: Genetic Algorithm based Selective Neural Network Ensemble. In: Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI'01), Seattle, WA, (2001), vol.2, 797-802.

[13] Zhou Z.H., Wu J.X., Tang W.: Ensembling Neural Networks: Many Could Be Better Than All. Artificial Intelligence, (2002), 137(1-2): 239-263.