

A Scalable Heuristic Classifier for Huge Datasets: A Theoretical Approach

Hamid Parvin, Behrouz Minaei-Bidgoli, and Sajad Parvin

Islamic Azad University, Nourabad Mamasani Branch, Nourabad Mamasani, Iran
{parvin,b_minaei,s.parvin}@iust.ac.ir

Abstract. This paper proposes a heuristic classifier ensemble to improve the performance of learning in multiclass problems. Although the more accurate classifier leads to a better performance, there is another approach to use many inaccurate classifiers while each one is specialized for a few data in the problem space and using their consensus vote as the classifier. In this paper, some ensembles of classifiers are first created. The classifiers of each of these ensembles jointly work using majority weighting votes. The results of these ensembles are combined to decide the final vote in a weighted manner. Finally the outputs of these ensembles are heuristically aggregated. The proposed framework is evaluated on a very large scale Persian digit handwritten dataset and the experimental results show the effectiveness of the algorithm.

Keywords: Genetic Algorithm, Optical Character Recognition, Pairwise Classifier, Multiclass Classification.

1 Introduction

In practice, there may be problems that one single classifier can not deliver a satisfactory performance [7], [8] and [9]. In such situations, employing ensemble of classifying learners instead of single classifier can lead to a better learning [6]. Although obtaining the more accurate classifier is often targeted, there is an alternative way to obtain it. Indeed one can use many inaccurate classifiers each of which is specialized for a few data items in the problem space and then employ their consensus vote as the classification. This can lead to better performance due to reinforcement of the classifier in error-prone problem spaces.

In General, it is ever-true sentence that "combining the diverse classifiers which are better than random results in a better classification performance" [2], [6] and [10]. Diversity is always considered as a very important concept in classifier ensemble methodology. It refers to being as much different as possible for a typical ensemble. Assume an example dataset with two classes. Indeed the diversity concept for an ensemble of two classifiers refers to the probability that they produce dissimilar results for an arbitrary input sample. The diversity concept for an ensemble of three classifiers refers to the probability that one of them produces dissimilar result from the two others for an arbitrary input sample. It is worthy to mention that the diversity can converge to 0.5 and 0.66 in the ensembles of two and three classifiers respectively. Although reaching the

more diverse ensemble of classifiers is generally handful, it is harmful in boundary limit. It is very important dilemma in classifier ensemble field: the ensemble of accurate-diverse classifiers can be the best. It means that although the more diverse classifiers, the better ensemble, it is provided that the classifiers are better than random.

An Artificial Neural Network (ANN) is a model which is to be configured to be able to produce the desired set of outputs, given an arbitrary set of inputs. An ANN generally composed of two basic elements: (a) neurons and (b) connections. Indeed each ANN is a set of neurons with some connections between them. From another perspective an ANN contains two distinct views: (a) topology and (b) learning. The topology of an ANN is about the existence or nonexistence of a connection. The learning in an ANN is to determine the strengths of the topology connections. One of the most representatives of ANNs is MultiLayer Perceptron. Various methods of setting the strength of connections in an MLP exist. One way is to set the weights explicitly, using a prior knowledge. Another way is to 'train' the MLP, feeding it by teaching patterns and then letting it change its weights according to some learning rule. In this paper the MLP is used as one of the base classifiers.

Decision Tree (DT) is considered as one of the most versatile classifiers in the machine learning field. DT is considered as one of unstable classifiers. It means that it can converge to different solutions in successive trainings on same dataset with same initializations. It uses a tree-like graph or model of decisions. The kind of its knowledge representation is appropriate for experts to understand what it does [11].

Its intrinsic instability can be employed as a source of the diversity which is needed in classifier ensemble. The ensemble of a number of DTs is a well-known algorithm called Random Forest (RF) which is considered as one of the most powerful ensemble algorithms. The algorithm of RF was first developed by Breiman [1].

In a previous work, Parvin et al. have only dealt with the reducing the size of classifier ensemble [9]. They have shown that one can reduce the size of an ensemble of pairwise classifiers. Indeed they propose a method for reducing the ensemble size in the best meaningful manner. Here we inspire from their method, we propose a framework based on that a set of classifier ensembles are produced that its size order is not important. Indeed we propose an ensemble of binary classifier ensembles that has the order of c , where c is number of classes.

This paper proposes a framework to develop combinational classifiers. In this new paradigm, a multiclass classifier in addition to a few ensembles of pairwise classifiers creates a classifier ensemble. At last, to produce final consensus vote, different votes (or outputs) are gathered, after that a heuristic classifier ensemble algorithm is employed to aggregate them.

This paper focuses on Persian handwritten digit recognition (PHDR), especially on Hoda dataset [4]. Although there are well works on PHDR, it is not rational to compare them with each other, because there was no standard dataset in the PHDR field until 2006 [4]. The contribution is only compared with those used the same dataset used in this paper, i.e. Hoda dataset.

2 Artificial Neural Network

A first wave of interest in ANN (also known as 'connectionist models' or 'parallel distributed processing') emerged after the introduction of simplified neurons by McCulloch and Pitts in 1943. These neurons were presented as models of biological neurons and as conceptual components for circuits that could perform computational tasks. Each unit of an ANN performs a relatively simple job: receive input from neighbors or external sources and use this to compute an output signal which is propagated to other units. Apart from this processing, a second task is the adjustment of the weights. The system is inherently parallel in the sense that many units can carry out their computations at the same time. Within neural systems it is useful to distinguish three types of units: input units (indicated by an index *i*) which receive data from outside the ANN, output units (indicated by an index *o*) which send data out of the ANN, and hidden units (indicated by an index *h*) whose input and output signals remain within the ANN. During operation, units can be updated either synchronously or asynchronously. With synchronous updating, all units update their activation simultaneously; with asynchronous updating, each unit has a (usually fixed) probability of updating its activation at a time *t*, and usually only one unit will be able to do this at a time. In some cases the latter model has some advantages.

An ANN has to be configured such that the application of a set of inputs produces the desired set of outputs. Various methods to set the strengths of the connections exist. One way is to set the weights explicitly, using a priori knowledge. Another way is to 'train' the ANN by feeding it teaching patterns and letting it change its weights according to some learning rule. For example, the weights are updated according to the gradient of the error function. For further study the reader must refer to an ANN book such as Haykin's book on theory of ANN [3].

<i>Tid</i>	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Fig. 1. An exemplary raw data

3 Decision Tree Learning

DT as a machine learning tool uses a tree-like graph or model to operate deciding on a specific goal. DT learning is a data mining technique which creates a model to predict the value of the *goal* or *class* based on input variables. Interior nodes are the representative of the input variables and the leaves are the representative of the target

value. By splitting the source set into subsets based on their values, DT can be learned. Learning process is done for each subset by recursive partitioning. This process continues until all remain features in subset has the same value for our goal or until there is no improvement in *Entropy*. Entropy is a measure of the uncertainty associated with a random variable.

Data comes in records of the form: $(x, Y) = (x_1, x_2, x_3, \dots, x_n, Y)$. The dependent variable, Y , is the target variable that we are trying to understand, classify or generalize. The vector x is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task. To clarify that what the DT learning is, consider Fig.1. Fig.1 has 3 *attributes* *Refund*, *Marital Status* and *Taxable Income* and our goal is cheat status. We should recognize if someone cheats by the help of our 3 attributes. To do learn process, attributes split into subsets. Fig.2 shows the process tendency. First, we split our source by the *Refund* and then *MarSt* and *TaxInc*.

For making rules from a decision tree, we must go upward from leaves as our antecedent to root as our consequent. For example consider Fig.2. Rules such as following are apprehensible. We can use these rules such as what we have in Association Rule Mining.

- Refund=Yes \Rightarrow cheat=No
- TaxInc<80, MarSt= (Single or Divorce), Refund=No \Rightarrow cheat=No
- TaxInc>80, MarSt= (Single or Divorce), Refund=No \Rightarrow cheat=Yes
- Refund=No, MarSt=Married \Rightarrow cheat=No

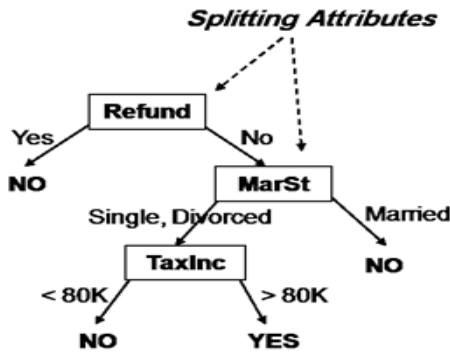


Fig. 2. The process tendency for Fig.1

4 Proposed Algorithm

The main idea behind the proposed method is to use a number of pairwise classifiers to reinforce the main classifier in the error-prone regions of the problem space. Fig.3 depicts the training phase of the proposed method schematically.

In the proposed algorithm, a multiclass classifier is first trained. Its duty is to produce a confusion matrix over the validation set. Note that this classifier is trained over the total train set. At next step, the pair-classes which are mostly confused with each

other and are also mostly error-prone are detected. After that, a number of pairwise classifiers are employed to reinforce the drawbacks of the main classifier in those error-prone regions. A simple heuristic is used to aggregate their outputs.

At the first step, a multiclass classifier is trained on all train data. Then, using the results of this classifier on the validation data, confusion matrix is obtained. This matrix contains important information about the functionalities of classifiers in the dataset localities. The close and Error-Prone Pair-Classes (EPPC) can be detected using this matrix. Indeed, confusion matrix determines the between-class error distributions. Assume that this matrix is denoted by a . Item a_{ij} of this matrix determines how many instances of class c_j have been misclassified as class c_i .

Table 1 shows the confusion matrix obtained from the base multiclass classifier. As you can see, digit 5 (or equivalently class 6) is incorrectly recognized as digit 0 fifteen times (or equivalently class 1), and also digit 0 is incorrectly recognized as digit 5 fourteen times. It means 29 misclassifications have totally occurred in recognition of these two digits (classes). The mostly erroneous pair-classes are respectively (2, 3), (0, 5), (3, 4), (1, 4), (6, 9) and so on according to this matrix. Assume that the i -th mostly EPPC is denoted by $EPPC_i$. So $EPPC_1$ will be (2, 3). Also assume that the number of selected EPPC is denoted by k .

After determining the mostly erroneous pair-classes, or EPPCs, a set of m ensembles of binary classifiers is to be trained to jointly, as an ensemble of binary classifiers, reinforce the main multiclass classifier in the region of each EPPC. So as it can be inferred, it is necessary to train k ensembles of m binary classifiers. Assume that the ensemble which is to reinforce the main multiclass classifier in the region of $EPPC_i$ is denoted by PWC_i . Each binary classifier contained in PWC_i , is trained over a bag of train data like RF. The bags of train data contain only b percent of the randomly selected of train data. It is worthy to be mentioned that pairwise classifiers which are to participate in PWC_i are trained only on those instances which belongs to $EPPC_i$. Assume that the j -th classifier binary classifier of PWC_i is denoted by $PWC_{i,j}$. Because there exists m classifiers in each of PWC_i and also there exists k EPPC, so there will be $k*m$ binary classifiers totally. For example in the Table 1 the EPPC (2, 3) can be considered as an erroneous pair-class. So a classifier is necessary to be trained for that EPPC using those dataitems of train data that belongs to class 2 or class 3. As mentioned before, this method is flexible, so we can add arbitrary number of PWC_i to the base primary classifiers. It is expected that the performance of the proposed framework outperforms the primary base classifier. It is worthy to note that the accuracies of $PWC_{i,j}$ can easily be approximated using the train set. Because $PWC_{i,j}$ is trained only on b percent of the train set with labels belong to $EPPC_i$, provided that b is very small rate, then the accuracy of $PWC_{i,j}$ on the train set with labels belong to $EPPC_i$ can be considered as its approximated accuracy. Assume that the mentioned approximated accuracy of $PWC_{i,j}$ is denoted by $P_{i,j}$.

It is important to note that each of PWC_i acts as a binary classifier. As it mentioned each PWC_i contains m binary classifiers with an accuracy vector, P_i . It means of these binary ensemble can take a decision with weighed sum algorithm illustrated in [5]. So we can combine their results according to weighs computed by the equation 1.

$$w_{i,j} = \log\left(\frac{P_{i,j}}{1 - P_{i,j}}\right) \tag{1}$$

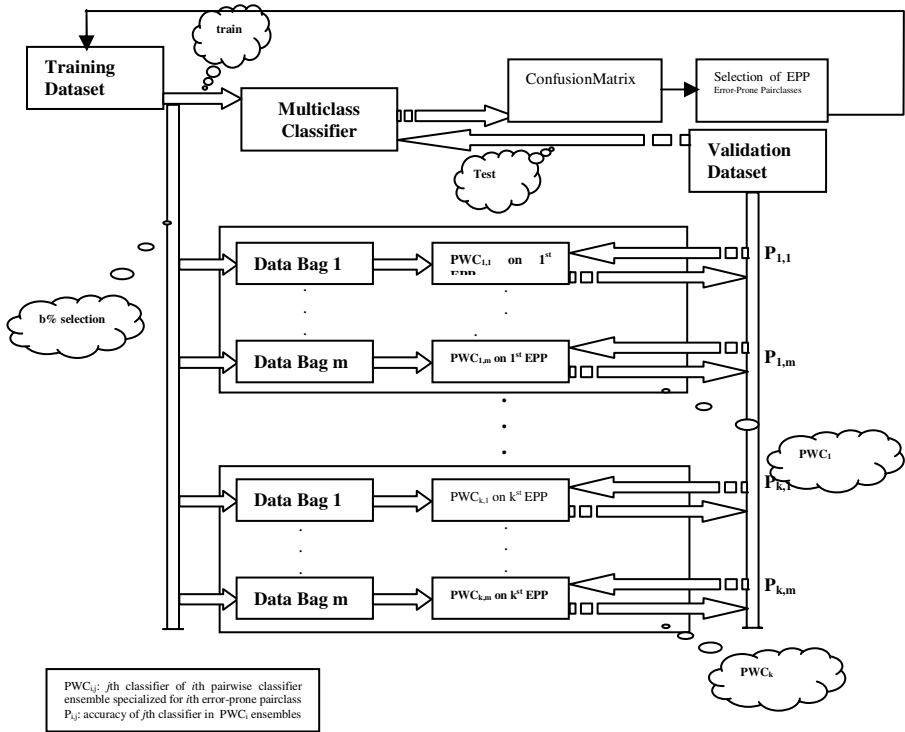


Fig. 3. The first training phase of the proposed method

where $w_{i,j}$ is the accuracy of j -th classifier in the i -th binary ensemble. It is proved that the weights obtained according to the equation 1 are optimal weights in theory. Now the two outputs of each PWC_i are computed as equation 2.

$$PWC_i(x|h) = \sum_{j=1}^m w_{i,j} * PWC_{i,j}(x|h) \quad , \quad h \in EPPC_i \quad 2$$

where x is a test data.

The last step of the proposed framework is to combine the results of the main multiclass classifier and those of PWC_i . It is worthy to note that there are $2*k$ outputs from the binary ensembles plus c outputs of the main multiclass classifier. So the problem is to map a $2*k+c$ intermediate space to a c space each of which corresponds to a class. The results of all these classifiers are fed as inputs in the aggregators. The Output i of aggregator is the final joint output for class i . Here, the aggregation is done using a special heuristic method. This process is done using a heuristic based ensemble which is illustrated in the Fig.4. As the Fig.4 shows, after producing the intermediate space, the outputs of i -th ensemble of binary classifier are multiplied in a q_i number. This q_i number is equal to the sum of the main multiclass classifier's confidences for the classes belong to $EPPC_i$. Assume that the results of the multiplication of q_i by the outputs of PWC_i are denoted by $MPWC_i$. It is important to note that $MPWC_i$ is a vector of two confidences; the confidences of the classifier framework to the classes belonging to PWC_i .

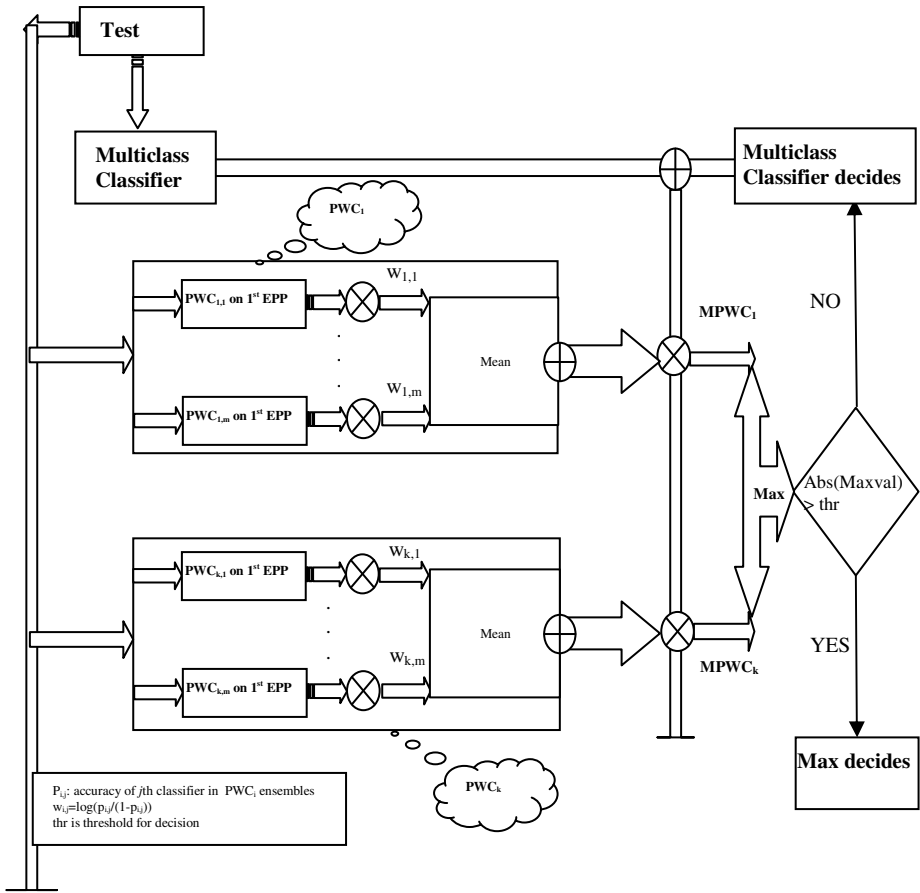


Fig. 4. Heuristic test phase of the proposed method test

After calculating the $MPWC_i$, the max value is selected between all of them. If the framework's confidence for the most confident class is satisfactory for a test data, then it is selected for final decision of framework, else the main multiclass classifier decides for the data. It means that the final decision is taken by equation 3.

$$Decision(x) = \begin{cases} MaxDecision(x) & \max_{h \in EPPC_{sc}} (MPWC_{sc}(h|x)) > thr \\ \max_{h \in \{1, \dots, c\}} (MCC(h|x)) & otherwise \end{cases} \tag{3}$$

where $MCC(h|x)$ is the confidence of the main multiclass classifier for the class h given a test data x . $MPWC_{sc}(h|x)$ is the confidence of the sc -th ensemble of binary classifiers for the class h given a test data x . $MaxDecision$ is calculated according to equation 4.

$$MaxDecision(x) = \arg \max_{h \in EPPC_{sc}} (MPWC_{sc}(h|x)) \tag{4}$$

where sc is:

$$sc(x) = \arg \max_i (\max_{h \in EPPC_i} (MPWC_i(h|x))) \tag{5}$$

Because of the reinforcement of the main classifier by some ensembles in erroneous regions, it is expected that the accuracy of this method outperforms a simple MLP or unweighted ensemble. Fig.3 along with Fig.4 stands as the structure of the ensemble framework.

5 Why the Proposed Method Works Results

As we presume in the paper, it is aimed to add as many as pairwise classifiers to compensate a predefined error rate, $PDER * EF(MCL, DValidation)$, where $PDER$ is a predefined error rate and $EF(MCL, DValidation)$ is error frequency of multiclass classifier, MCL , over the validation data, $DValidation$. Assume we add $|EPS|$ pairwise classifiers to the main MLC . It is as in the equation below.

$$\sum_{i=1}^{|eps|} (p(\hat{w} = EPPC_i.x | w = EPPC_i.y, x) + p(\hat{w} = EPPC_i.y | w = EPPC_i.x, x)) = PDER * EF(MCL, DValidation, DTrain) \tag{6}$$

Now assume that a data instance x which belongs really to class q is to be classified by the proposed algorithm; it has the error rate which can be obtain by equation 12. First assume p^p_{max} is probability for the proposed classifier ensemble to take decision by one of its binary classifiers that is able to distinguish two classes: q and p . Also assume p^{pr}_{max} is probability for the proposed classifier ensemble to take decision by one of its binary classifiers that is able to distinguish two classes: r and p . They can be is obtained by equation 7 and 8 respectively.

$$p^{pr}_{max}(EPPC = (p, r) | x \in q) = (MCC(p|x) + MCC(r|x)) * \max(PWC(p|x), PWC(r|x)) \tag{7}$$

$$p^p_{max}(EPPC = (p, q) | x \in q) = (MCC(p|x) + MCC(q|x)) * \max(PWC(p|x), PWC(q|x)) \tag{8}$$

We can assume equation 9 without losing generality.

$$\forall r \neq q | \max(PWC(p|x \in q), PWC(r|x \in q)) \cong \mu \ll \max(PWC(p|x \in q), PWC(q|x \in q)) = \lambda \tag{9}$$

where μ is a fixed value and then we have:

$$p^{pr}_{max}(EPPC = (p, r) | x \in q) \cong (MCC(p|x) + MCC(r|x)) \times \mu \propto (b_{p,q} + b_{r,q}) \times \mu \tag{10}$$

$$p^p_{max}(EPPC = (p, q) | x \in q) = (MCC(p|x) + MCC(q|x)) \times \lambda = (b_{p,q} + b_{q,q}) \times \lambda \tag{11}$$

As it is inferred from the algorithm in the same condition, its error can be formulated as follow.

$$\begin{aligned}
 error(x | w = q) = & \sum_{EPPC=(p,q)} p_{max}^p (EPPC | x) * p_{pair}(p | x) + \\
 & \sum_{EPPC=(p,r)} p_{max}^{pr} (EPPC | x) + (1 - p_{max}^p - p_{max}^{pr})(1 - b_{q,q})
 \end{aligned}
 \tag{12}$$

where p_{pair} is probability of taking correct decision by binary classifier and $b_{j,q}$ is defined as follow.

$$b_{j,p} = \frac{confusion_{j,p}}{\sum_{i=1}^c confusion_{i,p}}
 \tag{13}$$

So we can reformulate equation 12 as follow:

$$\begin{aligned}
 error(x | w = q) = & \sum_{EPPC=(p,q)} p_{max}^p (EPPC | x) * p_{pair}(p | x) + \sum_{EPPC=(p,r)} p_{max}^{pr} (EPPC | x) \\
 & + (1 - p_{max}^p - p_{max}^{pr})(1 - b_{q,q}) \cong \sum_{EPPC=(p,q)} p_{max}^p (EPPC | x) * p_{pair}(p | x) + \\
 & (1 - p_{max}^p - p_{max}^{pr})(1 - b_{q,q})
 \end{aligned}
 \tag{14}$$

Note that in equation 14 if p_{max}^{pr} and p_{max}^r are zero for an exemplary input the error of classification will be still equal to the main multiclass classifier. If they are not zero for an exemplary input the misclassification rate will still be reduced because of reduction in second part of equation 14. Although the first part increases the error in equation 14, but if we assume that the binary classifiers are more accurate than the multiclass classifier, then the increase is nullified by the decrease part.

6 Experimental Results

This section evaluates the results of applying the proposed framework on a Persian handwritten digit dataset named Hoda [4]. This dataset contains 102,364 instances of digits 0-9. Dataset is divided into 3 parts: train, evaluation and test sets. Train set contains 60,000 instances. Evaluation and test datasets are contained 20,000 and 22,364 instances. The 106 features from each of them have been extracted which are described in [4].

In this paper, MLP and DT are used as base primary classifier. We use MLPs with 2 hidden layers including respectively 10 and 5 neurons in the hidden layer 1 and 2, as the base Multiclass classifier and base simple classifiers. Confusion matrix is obtained from its output. Also DT’s measure of decision is taken as Gini measure. The classifiers’ parameters are kept fixed during all of their experiments. It is important to take a note that all classifiers in the algorithm are kept unchanged. It means that all classifiers are considered as MLP in the first experiments. After that the same experiments are taken by substituting all MLPs whit DTs.

Table 1. Unsoft confusion matrix pertaining to the Persian handwritten OCR

	0	1	2	3	4	5	6	7	8	9
0	969	0	0	4	1	14	2	0	0	1
1	4	992	1	0	2	4	1	1	1	15
2	1	1	974	18	9	1	4	4	0	1
3	0	0	13	957	12	0	3	2	0	1
4	5	0	3	17	973	3	2	2	0	3
5	15	0	0	0	0	977	1	0	0	0
6	2	6	2	1	3	0	974	5	1	3
7	3	0	3	1	0	1	1	986	0	0
8	0	1	0	1	0	0	2	0	995	0
9	1	0	4	1	0	0	10	0	3	976

The parameter k is set to 11. So, the number of pairwise ensembles of binary classifiers added equals to 11 in the experiments. The parameter m is also set to 9. So, the number of binary classifiers per each EPPC equals to 9 in the experiments. It means that 99 binary classifiers are trained for the pair-classes that have considerable error rates. Assume that the error number of each pair-class is available. For choosing the most erroneous pair-classes, it is sufficient to sort error numbers of pair-classes. Then we can select an arbitrary number of them. This arbitrary number can be determined by try and error which it is set to 11 in the experiments.

As mentioned $9*11=99$ pairwise classifiers are added to main multiclass classifier. As the parameter b is selected 20, so each of these classifiers is trained on only b precepts of corresponding train data. It means each of them is trained over 20 percent of the train set with the corresponding classes. The cardinality of this set is calculated by equation 15.

$$Car = \|train\| * 2 * b / c = 60000 * 2 * 0.2 / 10 = 2400 \quad 15$$

It means that each binary classifier is trained on 2400 datapoints with 2 class labels. Table 2 shows the experimental results comparatively. As it is inferred the framework is outperforms the previous works and the simple classifiers in the case of employing decision tree as the base classifier.

Table 2. The accuracies of different settings of the proposed framework

Methods	DT	ANN
A simple multiclass classifier	96.57	97.83
Parvin et al. [9]	97.93	98.89
Weighed fusion	99.01	98.46

7 Conclusion

In this paper, a new method is proposed to improve the performance of multiclass classification system. We also propose a framework based on that a set of classifier ensembles are produced that its size order is not important. Indeed we propose an ensemble of binary classifier ensembles that has the order of c , where c is number of classes. So first an arbitrary number of binary classifier ensembles are added to main classifier. Then results of all these classifier are given to a set of a heuristic based ensemble. Usage of confusion matrix make proposed method a flexible one. The number of all possible pairwise classifiers is $c*(c-1)/2$ that it is $O(c^2)$. Using this method without giving up a considerable accuracy, we decrease its order to $O(1)$. This feature of our proposed method makes it applicable for problems with a large number of classes. The experiments show the effectiveness of this method. Also we reached to very good results in Persian handwritten digit recognition which is a very large dataset.

References

1. Breiman, L.: Bagging Predictors. *Journal of Machine Learning* 24(2), 123–140 (1996)
2. Gunter, S., Bunke, H.: Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. In: *IWFHR 2002* (January 15, 2002)
3. Haykin, S.: *Neural Networks, a comprehensive foundation*, 2nd edn. Prentice Hall International, Inc. (1999) ISBN: 0-13-908385-5
4. 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* 28(10), 1133–1141 (2007)
5. Kuncheva, L.I.: *Combining Pattern Classifiers, Methods and Algorithms*. Wiley, New York (2005)
6. Minaei-Bidgoli, B., Punch, W.F.: Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. In: *GECCO* (2003)
7. Parvin, H., Alizadeh, H., Minaei-Bidgoli, B.: A New Approach to Improve the Vote-Based Classifier Selection. In: *International Conference on Networked Computing and Advanced Information Management (NCM 2008)*, Korea (2008)
8. Parvin, H., Alizadeh, H., Fathi, M., Minaei-Bidgoli, B.: Improved Face Detection Using Spatial Histogram Features. In: *The 2008 Int. Conf. on Image Processing, Computer Vision, and Pattern Recognition (ICCV 2008)*, Las Vegas, Nevada, USA (July 14-17, 2008)
9. Parvin, H., Alizadeh, H., Minaei-Bidgoli, B., Analoui, M.: An Scalable Method for Improving the Performance of Classifiers in Multiclass Applications by Pairwise Classifiers and GA. In: *International Conference on Networked Computing and advanced Information Management (NCM 2008)*, Korea (2008)
10. Saberi, A., Vahidi, M., Minaei-Bidgoli, B.: Learn to Detect Phishing Scams Using Learning and Ensemble Methods. In: *IEEE/WIC/ACM International Conference on Intelligent Agent Technology, Workshops (IAT 2007)*, Silicon Valley, USA, November 2-5, pp. 311–314 (2007)
11. Yang, T.: Computational Verb Decision Trees. *International Journal of Computational Cognition*, 34–46 (2006)