International Journal of Learning Management Systems

http://dx.doi.org/10.12785/ijlms/010204

A Classifier Ensemble of Binary Classifier Ensembles

Hamid Parvin[1], Hamid Alinejad-Rokny[2], Sajad Parvin[1]

[1] *Department of Computer Engine, Nourabad Mamasani Branch, Islamic Azad University, Nourabad, Iran*

[2] *Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran*

Email: hamidparvin@mamasaniiau.ac.ir , alinejad@ualberta.ca , s.parvin@mamasaniiau.ac.ir

Received: 15 Oct. 2012, Revised: 20 Nov. 2012, Accepted: 18 Dec. 2012

Published online: 1 Jul. 2013

Abstract: This paper proposes an innovative combinational algorithm to improve the performance in multiclass classification domains. Because the more accurate classifier the better performance of classification, the researchers in computer communities have been tended to improve the accuracies of classifiers. Although obtaining the more accurate classifier is often aimed, there is an alternative option to reach for it. Indeed one can use many inaccurate classifiers each of which is specialized for a subspace in the problem space and then s/he can consider their consensus vote as the classification. This paper proposes a new ensembles methodology that uses ensemble of binary classifiers as elements of an ensemble. These ensembles of binary classifiers jointly work using majority weighted voting. The results of these ensembles are in weighted manner combined to decide the final vote of the classification. In empirical result, these weights in final classifier are determined with using a series of genetic algorithms. We evaluate the proposed framework on a very large scale Persian digit handwritten dataset and the results show effectiveness of the algorithm.

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

Introduction

In practice, there may be problems that one single classifier can't deliver a satisfactory performance [13]. In such situations, employing ensemble of classifying learners instead of single classifier can lead to a better learning [11]. 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 dataitems 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" [5], [11] and [15]. 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

∗ This work is based on an earlier work of Parvin et al. [21]

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.

Evolutionary computations are considered universal optimizers or problem solvers. It is common to take it as an optimizer in large fields of science. The most well-known of them is considered to be Genetic Algorithm (GA). John Holland first introduced GA [6].

GA like other machine learning algorithms is based loosely on mechanism of biological evolution. It is applied in the wide problem spaces [11] in two ways: their direct usage as classifiers [8], and their usage as optimizing tools for determining parameters of classifiers. In [2], the GA is used to find decision boundaries in feature space. Another application of the GA is optimization of parameters in classification process. Many researchers also use GA in feature subset selection [1], [4], [10], [14] and [16]. Combination of classifiers is another field that GA has a hand as an optimization tool. Indeed, GA has also been used for feature selection in classifier ensemble [9] and [12].

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 [17].

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 [3].

This paper proposes a framework to develop combinational classifiers. In this new paradigm, a multiclass classifier in addition to a few pairwise classifiers creates a classifier ensemble. At last, to produce final consensus vote, different votes (or outputs) are gathered, after that the weighted majority voting algorithm is employed to aggregate them. The weights are determined by universal optimizer problem solvers like genetic algorithm.

This paper focuses on Persian handwritten digit recognition (PHDR), especially Hoda dataset [7]. 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 [7]. The contribution is only compared with those used the same dataset used in this paper, i.e. Hoda dataset.

1.2. 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].

1.2. 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: $(x,Y) = (x_1, x_2, x_3,\ldots, x_n, Y)$. 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 table that 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. First, we split our source by the Refund and then MarSt and finally TaxInc. For making rules from a decision tree, we must go upward from leaves as our antecedent to root as our consequent.

1.3. k-Nearest Neighbor

K-nearest neighbor algorithm (k-NN) is a method for classifying objects based on closest training examples in the feature space. k-NN is a type of instance-based learning, or lazy learning where the function is only approximated locally and all computation is deferred until classification. The k-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor.

As it is obvious, the k-NN classifier is a stable classifier. A stable classifier is the one converge to an identical classifier apart from its training initialization. It means the 2 consecutive trainings of the k-NN algorithm with identical k value, results in two classifiers with the same performance. This is not valid for

the MLP and DT classifiers. We use 3-NN as a base classifier in the paper. It is then inferred that using a k-NN classifier in an ensemble is not a good option.

2. Proposed Algorithm

The main idea behind the proposed method is to use a number of pairwise classifiers to reinforce the main classifier in error-prone regions of problem space. Figure 1 depicts the training phase of the proposed method schematically.

In the proposed algorithm, a multiclass classifier is first trained. Its duty is to obtain confusion matrix over 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 set of distinct classifiers is used for each class as an ensemble which is to learn that class. Considering the outputs of the main multiclass classifier and ones of the pairwise classifiers totally as a new space, GA is finally used to determine the weight of each classifier to vote in the ensemble. So, GA is run as many as the number of classes. It means GA is utilized as an aggregator in an ensemble detecting a class. Assume that the number of classes is denoted by c. So GA is run c times, each of them is denoted by GA_1, GA_2,..., GA_c. GA_i means the running of GA to detect i-th class or equivalently (i-1)-th digit.

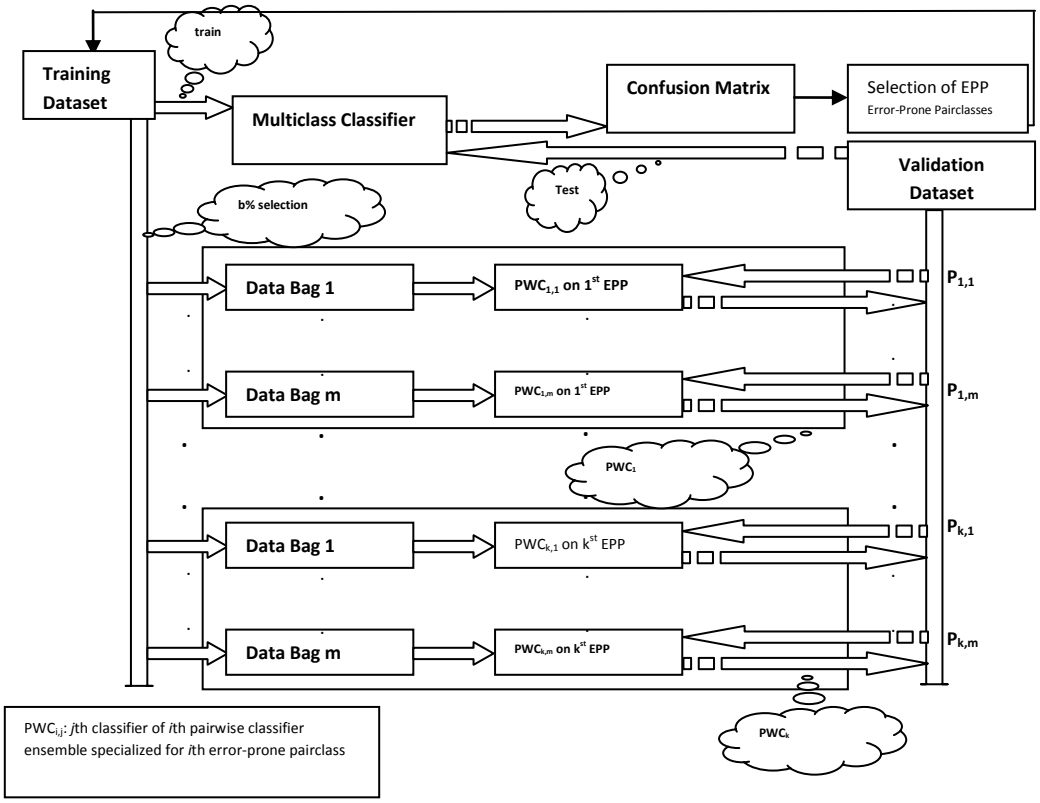

Figure 1: The first training phase of the proposed method.

2.1. Determining erroneous pair-classes

At the first step, a multiclass classifier is trained on all train data. Then, using results of this classifier on the evaluation 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 (EPPS) 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.

Figure 2 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 with a high value (or equivalently class 1), and also digit 0 is incorrectly recognized as digit 5 with a high value. 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.

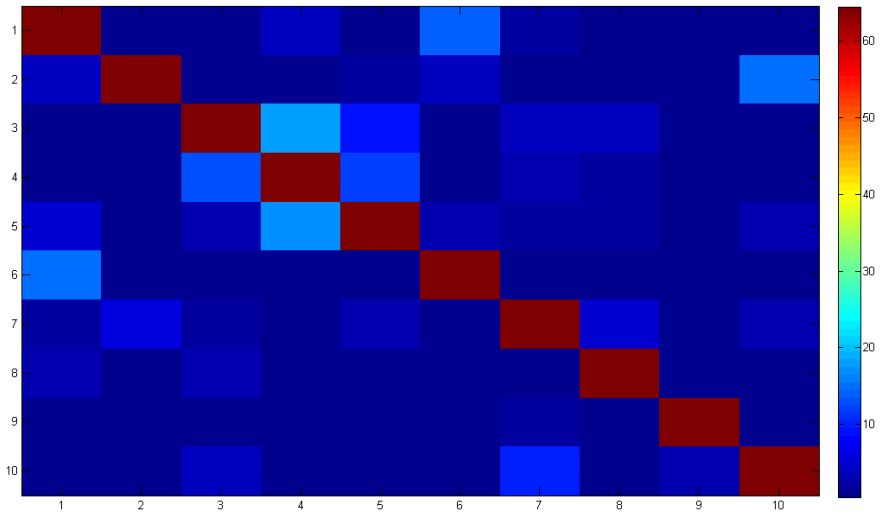

Figure 2: Unsoft confusion matrix pertaining to the Persian handwritten OCR.

2.2. Training of pairwise classifiers

After determining the mostly erroneous pair-classes, or EPPCs, a set of m 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 percept 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 Figure 2 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 percept 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 [9]. So we can combine their results according to weighs computed by the below equation.

$$w_{i,j} = \log(\frac{p_{i,j}}{1-p_{i,j}}) \quad (1)$$

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.

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

where x is a test data.

2.3. Fusion of pairwise classifiers

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 for the aggregators. Note that there are c aggregators, one per each class. The Output of aggregator i is the final joint output for class i. Here, the aggregation is done using a special weighting method. The problem here is how one can optimally determine these weights. In this paper, GA is employed to find these weights.

Because of the capability of the GA in passing local optimums, it is expected that the accuracy of this method outperforms a simple MLP or unweighted ensemble. Figure 1 along with Figure 3 and Figure 4 depicts the structure of the ensemble framework.

As it is shown in Figure 3, in the proposed framework, the number of times that GA is invoked is equal to c, which is the number of digits (classes). This GA-based algorithm is overall illustrated by Figure 3.

In fact, each GA creates an ensemble to detect one digit (class), by considering the 2*k+c intermediate space obtained by the multiclass classifier plus the binary classifier ensembles as new feature space. Each GA uses one hyper-line in this new intermediate feature space, by assigning a weight to each dimension. The chromosome representation of GAi is a vector of real numbers. The function of GAi is calculated as equation 3.

$$Fitness(W_i, ValSet) = \sum_{x \in ValSet} f(x, W_i) \quad (3)$$

where the function $f(x, W_i)$ is computed as equation 4.

$$f(x, W_i) = sign(x, i) * \left(BinOuts(x, W_i) + MultiOuts(x, W_i)\right) \quad (4)$$

where the function sing(x, i) is computed as equation **5.**

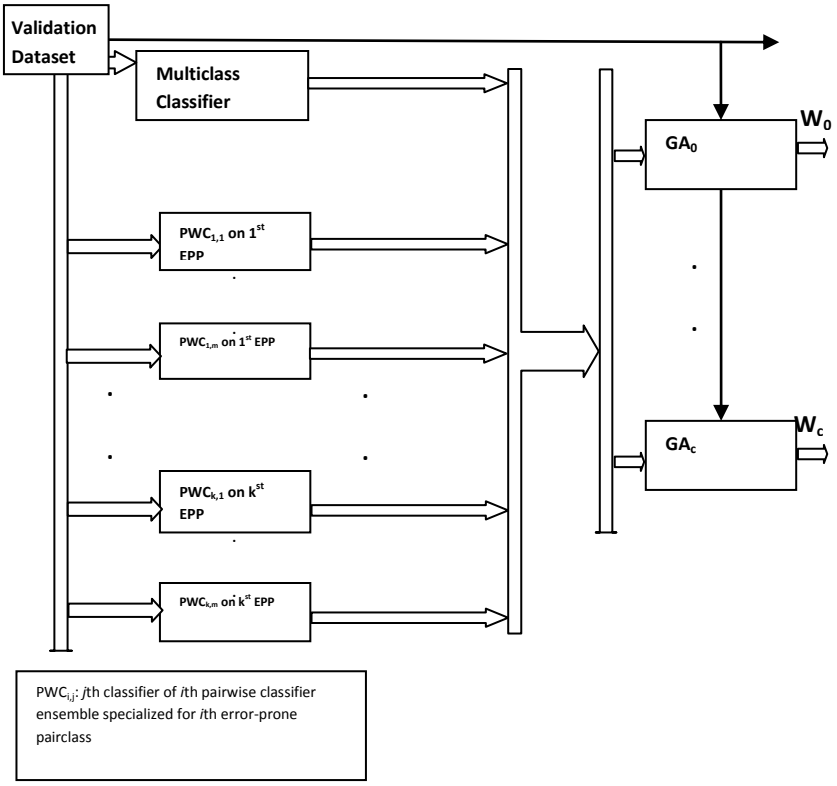

Figure 3: The second training phase of the proposed method based on GA.

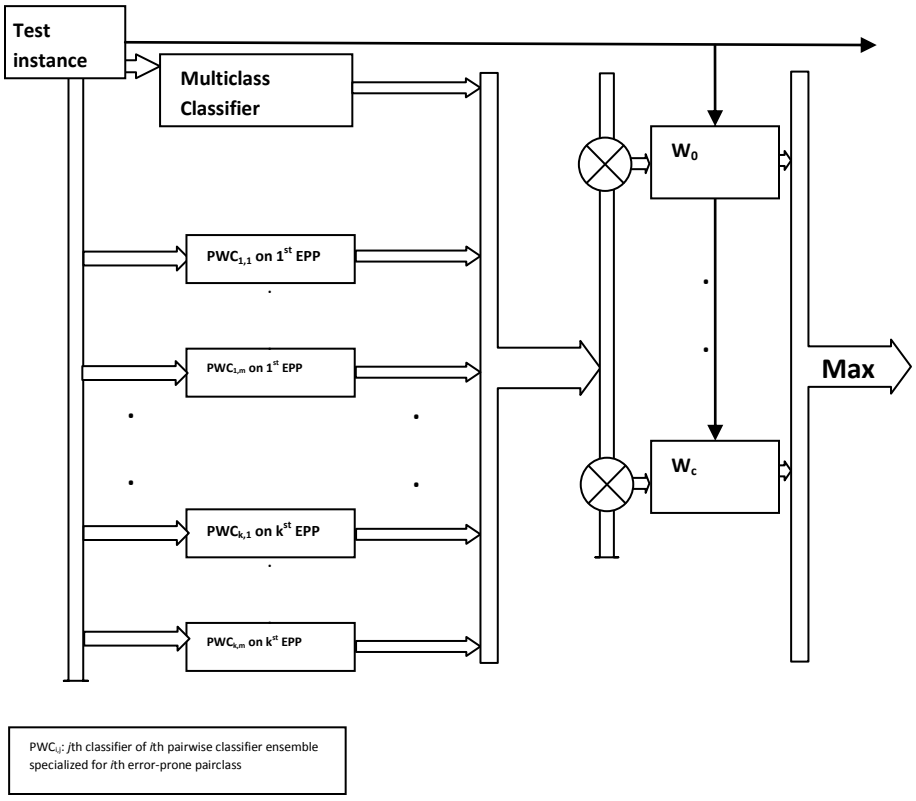

Figure 4: Test phase of the proposed method based on GA.

$$sign(x,i)=\begin{cases}1 & label(x)=i\\ -1 & label(x)\neq i\end{cases} \tag{5}$$

And ValSet in equation 4 is validation set. In the equation 4, BinOuts is the weighted sum of the outputs of the binary ensembles, given an input sample x, which is computed as equation 6, and MultiOuts is the weighted sum of the outputs of the main multiclass classifier, given an input sample x.

$$BinOuts(x,W_i)=\sum_{j=1}^{k}\sum_{h=1}^{2}W_i(s)*PWC_{i,j}(x\mid EPPC^{h}_{\lceil j/2\rceil}) \tag{6}$$

where s is computed as equation 7.

$$s=(j-1)*2 \tag{7}$$

and MultiOuts is also computed as equation 8.

$$MultiOuts(x,W_i)=\sum_{j=1}^{c}W_i(2*k+j)*MCC(x\mid j) \tag{8}$$

Indeed GA_i try to better discriminate the class i from other classes. Finally, the most voted class is selected as final decision of the framework as depicted in the Figure 4. This is simply done using a max function as it is obvious from the Figure 4. It means that the final decision is taken by equation 9.

$$FinalDecision(x)=\arg\max_i f(x,W_i) \tag{9}$$

3. Experimental Results

This section evaluates the results of applying the proposed framework on a Persian handwritten digit dataset named Hoda [7]. 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 [7]. Some instances of this dataset are depicted in Figure 5.

3.1. Parameter Setting

In this paper, MLP and DT are used as base primary classifier. We use an MLPs with 2 hidden layers including respectively 10 and 5 neurons in the hidden layer 1 and 2, as the base Multiclass classifier. 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.

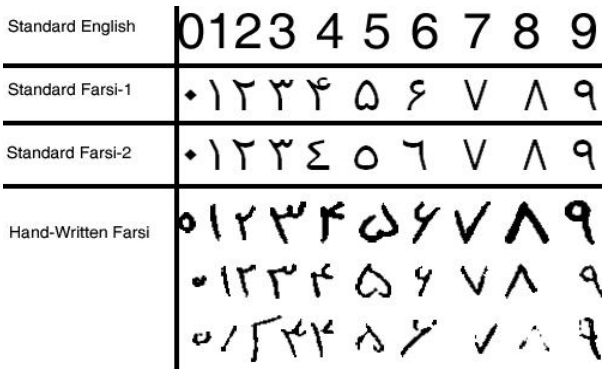

Figure 5: Some instances of Persian OCR data set, with different qualities.

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=110 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 percept of the train set with the corresponding classes. The cardinality of this set is calculated by equation 10.

$$Car = \|train\| * 2 * b / c = 60000 * 2 * 0.2 / 10 = 2400 \tag{10}$$

It means that each binary classifier is trained on 2400 datapoints with 2 class labels.

The results of primary multiclass classifier and those of pairwise binary classifier ensembles are given to 10 GAs as inputs. Therefore, each chromosome contains 32 (22 for outputs of pairwise classifiers ensemble and a more 10 for outputs of the primary multiclass classifier) genes per each class. The number of GAs equals to the number of labels. Gaussian and Scattered operators are respectively used for mutation and recombination. Also, population size is 500. P_{mut} is set to 0.01 and the mutation is considered bitwise. $P_{crossover}$ is set to 0.8.

Termination condition is passing of 200 generations. Fitness function for GA is as mentioned in equation 3. The output of each GA is certainty of GA to select its corresponding class. Max function selects the most certain decision as final joint decision. Table 1 shows the accuracies of the different methods.

It is inferred from the Table 1 that the proposed framework affects significantly in improving the classification precision specially when employing DT and MLP as base classifier. It is also obvious that using DT classifier as base classifier has the most impact in improving the recognition ratio. It is may due to its inherent instability.

As it is expected using a stable classifier like k-NN in an ensemble is not a good option and unstable classifiers like DT and MLP are better options.

4. Conclusion

In this paper, a new method is proposed to improve the performance of multiclass classification system. An arbitrary number of binary classifiers are added to main classifier to increase its accuracy. Then results of all these classifier are given to a set of GAs. The final results can competently obtain by certain weighting approach.

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.

Table 1: Summary of experimental results employing histogram equalization.

Methods	Base Classifier		
	DT	MLP	3-NN
A simple multiclass classifier	95.57	95.7	96.66
Method Proposed in [18]	-	98.89	-
Method Proposed in [19]	-	98.27	-
Method Proposed in [20]	97.20	96.70	96.86
Weighed fusion with GA	98.99	99.04	97.14

References

[1] Bala, J., De Jong, K., Huang, J., Vafaie, H. & Wechsler, H. (1997). Using learning to facilitate the evolution of features for recognizing visual concepts. *Evolutionary Computation*, 4(3), 297-311.

[2] Bandyopadhyay, S. & Muthy, C.A. (1995). Pattern Classification Using Genetic Algorithms. *Pattern Recognition Letters*, 16, 801-808.

[3] Breiman, L. (1996). Bagging Predictors. *Journal of Machine Learning*, 24(2), 123-140.

[4] Guerra-Salcedo, C. & Whitley, D. (1999). Feature Selection mechanisms for ensemble creation: a genetic search perspective, In: Freitas AA (Ed.). *Data Mining with Evolutionary Algorithms: Research Directions-Papers from the AAAI Workshop*, 13-17.

[5] Gunter, S. & Bunke, H. (2002). Creation of classifier ensembles for handwritten word recognition using feature selection algorithms. *IWFHR 2002 on January 15*, 183-188.

[6] Holland, J. (1975). *Adaptive in Natural and Artificial Systems*. MIT Press, Cambridge, MA, (1st edition: 1975, The University of Michigan Press, Ann Arbor).

[7] Khosravi, H. & Kabir, E. (2007). 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.

[8] Kuncheva, L.I. & Jain, L.C. (2000). Designing Classifier Fusion Systems by Genetic Algorithms. *IEEE Transaction on Evolutionary Computation*, 33, 351-373.

[9] Kuncheva, L.I. (2005). *Combining Pattern Classifiers, Methods and Algorithms*. New York: Wiley.

[10] Martin-Bautista, M.J. & Vila, M.A. (1999). A survey of genetic feature selection in mining issues. *Proceeding Congress on Evolutionary Computation* (CEC-99), 1314-1321.

[11] Minaei-Bidgoli, B. & Punch, W.F. (2003). Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. *GECCO*.

[12] Minaei-Bidgoli, B., Kortemeyer, G. & Punch, W.F. (2004). Mining Feature Importance: Applying Evolutionary Algorithms within a Web-Based Educational System. *Proc. of the Int. Conf. on Cybernetics and Information Technologies, Systems and Applications.*

[13] Parvin, H., Alizadeh, H., Fathi, M. & Minaei-Bidgoli, B. (2008). Improved Face Detection Using Spatial Histogram Features. *The 2008 Int. Conf. on Image Processing, Computer Vision, and Pattern Recognition (IPCV'08)*, Las Vegas, Nevada, USA, July 14-17.

[14] Punch, W.F., Pei, M., Chia-Shun, L., Goodman, E.D., Hovland, P. & Enbody, R. (1993). Further research on Feature Selection and Classification Using Genetic Algorithms. *In 5th International Conference on Genetic Algorithm*, Champaign IL, 557-564.

[15] Saberi, A., Vahidi, M. & Minaei-Bidgoli, B. (2007). Learn to Detect Phishing Scams Using Learning and Ensemble Methods. *IEEE/WIC/ACM International Conference on Intelligent Agent Technology*, Workshops (IAT 07), 311-314.

[16] Vafaie, H. & De Jong, K. (1993). Robust feature Selection algorithms. *Proceeding 1993 IEEE Int. Conf on Tools with AI*, 356-363.

[17] Yang, T. (2006). Computational Verb Decision Trees. *International Journal of Computational Cognition*, 4(4), 34-46.

[18] Parvin, H., Alizadeh, H., Minaei-Bidgoli, B. & Analoui, M. (2008). An Scalable Method for Improving the Performance of Classifiers in Multiclass Applications by Pairwise Classifiers and GA. *International Conference on Networked Computing and advanced Information Management (NCM 2008),*.

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

[20] Parvin, H., Alizadeh, H., Moshki, M., Minaei-Bidgoli, B., Mozayani, N. (2008). Divide & Conquer Classification and Optimization by Genetic Algorithm. *International Conference on Convergence and hybrid Information Technology (ICCIT08)*, 11-13.

[21] Parvin, H., Minaei-Bidgoli, B., Alizadeh, H. (2011). A New Adaptive Framework for Classifier Ensemble in Multiclass Large Data. *International Conference on Computational Science and Its Applications (ICCSA 2011)*, LNCS, ISSN: 0302-9743, Springer, Heidelberg, 526-536.