Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 918105, 9 pages
http://dx.doi.org/10.1155/2015/918105

Research Article

Multiclass AdaBoost ELM and Its Application in LBP Based Face Recognition

Yunliang Jiang,[1,2] Yefeng Shen,[1,3] Yong Liu,[1] and Weicong Liu[1]

[1]*Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China*
[2]*School of Information & Engineering, Huzhou Teachers College, Huzhou 313000, China*
[3]*School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China*

Correspondence should be addressed to Yong Liu; yongliu@iipc.zju.edu.cn

Received 22 August 2014; Revised 11 November 2014; Accepted 18 November 2014

Academic Editor: Jiuwen Cao

Extreme learning machine (ELM) is a competitive machine learning technique, which is simple in theory and fast in implementation; it can identify faults quickly and precisely as compared with traditional identification techniques such as support vector machines (SVM). As verified by the simulation results, ELM tends to have better scalability and can achieve much better generalization performance and much faster learning speed compared with traditional SVM. In this paper, we introduce a multiclass AdaBoost based ELM ensemble method. In our approach, the ELM algorithm is selected as the basic ensemble predictor due to its rapid speed and good performance. Compared with the existing boosting ELM algorithm, our algorithm can be directly used in multiclass classification problem. We also carried out comparable experiments with face recognition datasets. The experimental results show that the proposed algorithm can not only make the predicting result more stable, but also achieve better generalization performance.

1. Introduction

Many research works have been done in feedforward neural networks, which pointed out that the feedforward neural networks are able to not only approximate complex nonlinear mapping, but also provide models for some natural and artificial problems which classic parametric technics are unable to handle.

Recently, Huang et al. [1] proposed a new simple algorithm based on single layer feedforward networks (SLFNs) called extreme learning machine (ELM). For ELM randomly generates parameters of the networks, its learning speed can be thousands of times faster than traditional feedforward network learning algorithms like back-propagation (BP) algorithm, which needs to iterate many times to get optimal parameters.

In addition, Huang [2] also shows that in theory ELMs (with the same kernels) tend to outperform SVM and its variants in both regression and classification applications with much easier implementation. Based on this conclusion, the paper in the literature proposed by Wong et al. [3] explores the superiority of the fault identification time of ELM.

In view of the advantages of the algorithm, Cao et al. put it into some areas, such as landmark recognition [4] and protein sequence classification [5]. Besides, Cao et al. [6] proposed an improved learning algorithm which incorporates the voting method into the popular extreme learning machine in classification applications and outperforms the original ELM algorithm as well as several recent classification algorithms.

AdaBoost [7] is one of the most popular algorithms of classifier ensemble to improve the generalization performance. Wang and Li in [8] proposed an algorithm named dynamic AdaBoost ensemble ELM (named DAEELM in this paper). The proposed algorithm takes the ELM as the basic classifier and applies AdaBoost to solve binary classification problem. Similarly, Tian and Mao in [9] combined the modified AdaBoost.RT [10] with ELM to propose a new hybrid artificial intelligent technique called ensemble ELM.

Ensemble ELM aims to improve ELM's performance in regression problem.

However, until now, not so much works have been done to apply AdaBoost to ELM for multiclass classification problem directly. In Freund and Schapire's work [11], they give two extensions of their boosting algorithm to multiclass prediction problems in which each example belongs to one of several possible classes (rather than just two). Since ELM can directly work for multiclass classification problem, this paper proposes an algorithm named multiclass AdaBoost ELM (MAELM). This new algorithm applies multiclass AdaBoost as an ensemble method to a number of ELMs. In addition, this paper proposes a structure to apply ELM and MAELM to local binary patterns (LBP) [12] based face recognition problem. Experiments in LBP based face recognition will show that the proposed algorithm outperforms the original ELM.

This paper is an extension of our previous work [13]. In this paper, we extend our previous work by proposing a new way to combine ELM with PCA instead of using random weights between the input layer and the hidden layer, as well as the bias of the activation function. Experiments in LBP based face recognition will show the stable and good performance with our extended approach.

The rest of the paper is organized as follows. Section 2 gives a brief review of the ELM and PCA, original and multiclass AdaBoost and LBP. The proposed MAELM is presented in Section 3. The experimental result will be shown in Section 4 and a short discussion about the proposed algorithm will be presented in Section 5. Finally, in Section 6, we conclude the paper.

2. A Review of Related Work

In this section, a review of the original ELM algorithm and PCA and multiclass AdaBoost and the LBP based face recognition is presented.

2.1. ELM. For N arbitrary distinct samples (x_i, t_i), where $x_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T \in R^d$ and $t_i = [t_{i1}, t_{i2}, \ldots, t_{iK}]^T \in R^K$, standard SLFNs with L hidden nodes and activation function $h(x)$ are mathematically modeled as follows:

$$\sum_{i=1}^{L} \beta_i h_i\left(x_j\right) = \sum_{i=1}^{L} \beta_i h_i\left(w_i \cdot x_j + b_i\right) = o_j, \tag{1}$$

where $j = 1, 2, \ldots, N$.

Here, $w_i = [w_{i1}, w_{i2}, \ldots, w_{id}]^T$ is the weight vector connecting the ith hidden node and the input nodes, $\beta_i = [\beta_{i1}, \ldots, \beta_{iK}]^T$ is the weight vector connecting the ith hidden node and the output nodes, and b_i is the threshold of the ith hidden node.

The standard SLFNs with L hidden nodes with activation function $h(x)$ can be compactly written as follows:

$$H\beta = T, \tag{2}$$

where

$$H = \begin{bmatrix} h_1\left(w_1 \cdot x_1 + b_1\right) & \cdots & h_L\left(w_L \cdot x_1 + b_L\right) \\ \vdots & & \vdots \\ h_1\left(w_1 \cdot x_N + b_1\right) & \cdots & h_L\left(w_L \cdot x_N + b_L\right) \end{bmatrix},$$
$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}, \qquad T = \begin{bmatrix} t_1^T \\ \vdots \\ t_N^T \end{bmatrix}. \tag{3}$$

Different from the conventional gradient-based solution of SLFNs, ELM simply solves the function by

$$\beta = H^+ T. \tag{4}$$

H^+ is the Moore-Penrose generalized inverse of matrix H. As Huang et al. have pointed out in [14], H^+ can be represented by

$$H^+ = H^T\left(\frac{I}{C} + HH^T\right)^{-1}, \tag{5}$$

where I is an identity matrix, which has the same dimension with HH^T. C is a constant number which can be set by the user. Adding I/C can avoid the situation that HH^T is singular. Huang et al. [1] successfully applied ELM to solve binary classification problem and Huang et al. [14] extended the ELM to directly solve the multiclass classification problem.

Since the original ELM randomly generates the weights between the input layer and the hidden layer, as well as the bias of the activation function, its performance may be not so stable. Instead of that, some other ways like PCA algorithm rewards to try.

2.2. PCA. Principal component analysis (PCA) was invented in 1901 by Pearson [15], as an analogue of the principal axes theorem in mechanics, which was later independently developed (and named) by Hotelling in the 1930s [16]. Now, it is mostly used as a tool in exploratory data analysis and for making predictive models. PCA can be done by eigenvalue decomposition of a data covariance (or correlation) matrix or singular value decomposition of a data matrix, usually after mean centering (and normalizing or using Z-scores) the data matrix for each attribute [17]. The results of a PCA are usually discussed in terms of component scores, sometimes called factor scores (the transformed variable values corresponding to a particular data point) and loadings (the weight by which each standardized original variable should be multiplied to get the component score).

The procedure of PCA is as follows:

$$X = \left(x_{ij}\right)_{n*p} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ \vdots & \vdots & \ldots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{bmatrix}. \tag{6}$$

Step 1. Compute the matrix V which is the covariance matrix of X.

Step 2. Find out the eigenvalue of $|V - \lambda E| = 0$, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_p$.

Step 3. Compute the standardization feature vector of $(V - \lambda E)\beta = 0$ $\beta_1, \beta_2, \ldots, \beta_p$.

Step 4. Yield the principal components $Y_r = \beta_r' X$ $(r = 1, 2, \ldots, p)$.

E is an identity matrix, which has the same dimension with V. The matrix Y consists of n row vectors, where each vector is the projection of the corresponding data vector from matrix X.

PCA is a statistical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance, and each succeeding component in turn has the highest variance possible under the constraint that it is orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent if the dataset is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.

2.3. Original AdaBoost and Multiclass AdaBoost. AdaBoost has been very successfully applied in binary classification problem. Original AdaBoost is proposed in [7]. Before proposing the AdaBoost algorithm, the function $I(x)$ is predefined as

$$I(x) = \begin{cases} 1, & \text{if } x = \text{true} \\ 0, & \text{if } x = \text{false.} \end{cases} \tag{7}$$

AdaBoost algorithm is summarized as follows.

Given the training data $\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}$, where $x_i \in R^d$ denotes the ith input feature vector with d dimensions, y_i denotes the label of the ith input feature vector, where $y_i \in \{-1, +1\}$. Use $T_j(x)$ to denote the jth weak classifier and suppose M weak classifiers will be combined.

(1) Initialize the observation weights $\omega_i = 1/N$, $i = 1, 2, \ldots, N$.

(2) For $m = 1 : M$,

(a) fit a classifier $T_m(x)$ to the training data using weights ω_i;

(b) compute the weighted error

$$\text{err}_m = \frac{\sum_{i=1}^{N} \omega_i I\left(y_i \neq T_m\left(x_i\right)\right)}{\sum_{i=1}^{N} \omega_i}; \tag{8}$$

(c) compute the weight of the mth classifier

$$\alpha_m = \log \frac{1 - \text{err}_m}{\text{err}_m}; \tag{9}$$

(d) update the weights of sample data, for all $i = 1, 2, \ldots, N$

$$\omega_i = \omega_i \cdot \exp\left(\alpha_m \cdot I\left(y_i \neq T_m\left(x_i\right)\right)\right); \tag{10}$$

(e) renormalize ω_i, for all $i = 1, 2, \ldots, N$.

(3) Output

$$C(x) = \arg\max_k \sum_{m=1}^{M} \alpha_m \cdot I\left(T_m = k\right). \tag{11}$$

Here, k is $+1$ or -1. In binary classification, any classifier whose generalization performance is better than $1/2$ is a weak classifier. For the original AdaBoost, we have the following.

(1) For the ith and the jth classifiers, if $\text{err}_i < \text{err}_j < 1/2$, we have $\alpha_i > \alpha_j > 0$, which means the final ensemble classifier values more of the ith classifier's result. Specifically, if $\text{err}_j = 1/2$, $\alpha_j = 0$, which means the final ensemble classifier just ignores the classifier since its effect is the same as random guess.

(2) If the pth classifier misclassifies the qth sample, the qth sample will have a big weight in the next iteration. As a result, the $(p + 1)$th classifier will pay more attention to it. On the contrary, if the pth classifier classifies the qth sample correctly, the qth sample will have a small weight in the next iteration, which means $(p + 1)$th classifier will pay less attention to it.

However, for a K-class classification problem, we have $y_i \in \{1, 2, \ldots, K\}$ and $K > 2$. If a classifier's generalization performance is better than $1/K$ (maybe much smaller than $1/2$), it can be called a weak classifier. Since original AdaBoost only takes a classifier whose generalization performance is better than $1/2$ as a weak classifier, obviously, it cannot be directly implemented to multiclass conditions that K is bigger than 2. Freund and Schapire [11] extend the original AdaBoost to multiclass condition. The weight of the mth classifier is modified as

$$\alpha_m = \log \frac{1 - \text{err}_m}{\text{err}_m} + \log(K - 1). \tag{12}$$

Similar to the binary condition, for the ith and the jth classifiers, if $\text{err}_i < \text{err}_j < 1 - 1/K$, we have $\alpha_i > \alpha_j > 0$, which means the final ensemble classifier values more of the ith classifier's result. In particular, if $\text{err}_j = 1 - 1/K$, $\alpha_j = 0$.

2.4. LBP Based Face Recognition. The original LBP operator goes through each 3×3 neighborhood in a picture. It takes the center pixel as the threshold value of the neighborhood and considers the result as a decimal number. The LBP operator is shown in Figure 1. Then, the texture of the picture can be represented by the histogram of all the decimal numbers.

To apply LBP operator in face recognition problem, Ahonen et al. [12] divided the face image into several windows and calculated the histogram of each window by LBP operator. The final feature vector is gotten by combining the histograms

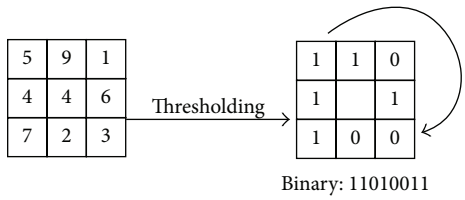

Figure 1: Basic LBP operator.

into a spatially enhanced histogram. The spatial enhanced histogram is provided with three levels of information: the patterns of pixel level; the patterns of regional level; the global patterns of the face image. Experiments in [12] have shown that the LBP description is more robust against variants in pose or illumination than holistic methods. All our experiments in Section 4 are done with the most original LBP operator.

3. MAELM and Face Recognition Structure

In this part, the multiclass AdaBoost ELM (MAELM) algorithm is proposed and a structure of face recognition based on LBP and ELM is also included.

3.1. Proposed MAELM Algorithm. By applying the multiclass AdaBoost to ELM, this paper proposes the multiclass AdaBoost ELM (MAELM) algorithm. The algorithm takes a number of ELM classifiers as the weak classifiers. $\mathrm{ELM}_i(x)$ denotes the ith ELM classifier. The proposed algorithm is put as follows.

(1) Initialize the observation weights $\omega_i = 1/N$, $i = 1, 2 \ldots, N$.

(2) For $m = 1 : M$,

(a) fit a classifier $\mathrm{ELM}_m(x)$ to the training data using weights ω_i;

(b) compute the weighted error

$$\mathrm{err}_m = \frac{\sum_{i=1}^{N} \omega_i I\left(c_i \neq \mathrm{ELM}_m\left(x_i\right)\right)}{\sum_{i=1}^{N} w_i}; \tag{13}$$

(c) compute the weight of the mth classifier

$$\alpha_m = \log \frac{1 - \mathrm{err}_m}{\mathrm{err}_m} + \log\left(K - 1\right); \tag{14}$$

(d) update the weight of sample data, for all $i = 1, 2, \ldots, N$

$$\omega_i = \omega_i \cdot \exp\left(\alpha_m \cdot I\left(c_i \neq \mathrm{ELM}_m\left(x_i\right)\right)\right); \tag{15}$$

(e) renormalize ω_i.

(3) Output

$$C\left(x\right) = \arg\max_k \sum_{m=1}^{M} \alpha_m \cdot I\left(\mathrm{ELM}_m\left(x\right) = k\right). \tag{16}$$

Part (2)(a) of the proposed algorithm should be paid more attention. Both [8, 9] did not give any detail of how to fit the basic classifier $\mathrm{ELM}_m(x)$ with weighted samples, but it is a very important part of AdaBoost. Zong et al. [18] proposed an algorithm named weighted ELM by introducing a diagonal matrix $W \in R^{N \times N}$, whose element $W_{i,i}$ denotes the weight of the ith training sample. In view of some special situations, we introduce the weighted ELM algorithm. Obviously, it boils down to the original one when the weighted matrix is the identity matrix.

The proposed method maintains the advantages from original ELM: (1) it is simple in theory and convenient in implementation; (2) wide types of feature mapping functions or kernels are available for the proposed framework; (3) the proposed method can be applied directly into multiclass classification tasks. In addition, after integrating with the weighting scheme, the weighted ELM is able to deal with data with imbalanced class distribution while maintaining the good performance on well-balanced data as unweighted ELM; by assigning different weights for each example according to the users' needs, the weighted ELM can be generalized to cost sensitive learning.

Under the weighted circumstance, the solution of β becomes

$$\beta = H^T \left(\frac{I}{C} + WHH^T\right)^{-1} WT. \tag{17}$$

3.2. Application in LBP Based Face Recognition. This paper combines LBP based feature vectors with ELM to build a face recognition structure. There have been some papers [19, 20] about applying ELM in face recognition problem. However, the existed ELM based face recognition structures are all based on statistical features, for example, PCA [21] and LDA [22].

In order to get better generalization performance, the proposed face recognition structure implements the LBP based method to get the feature vector and ELM as the classifier. It has been proved in [12] that LBP based method is more robust than PCA and LDA when lighting, facial expression, and poses change. At the same time, ELM is very fast in classification and has very good generalization performance. So, it is reasonable to combine LBP method and ELM to build the face recognition structure.

There are two steps of the proposed face recognition structure. The first step is to train the training samples by ELM or MAELM. In this step, the training samples are represented by LBP based feature vectors. Then, the feature vectors are used to train the classifier model by ELM or MAELM; see Figure 2. The second step is to predict the labels of the test samples. The test samples are also represented by the LBP based feature vectors. Then, the classifier model trained in the first step is implemented to predict the labels of the test samples; see Figure 3.

4. Experiments

In this paper, two of the mostly used face recognition datasets Yale and ORL are used to prove the efficiency of the proposed

Figure 2: Training the samples by ELM or MAELM.

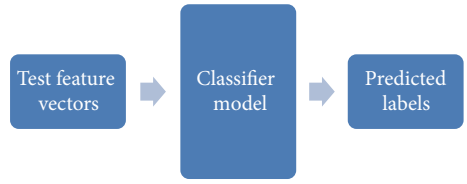

Figure 3: Predicting the labels of test samples.

Table 1: Parameter list.

Parameters	Meaning
M	Number of the basic classifiers
C	Constant value in generalized inverse of H
L	Number of hidden nodes in ELM
t	Number of training images of each person
w	Divide each face image into $w * w$ windows
r	The dimension after reduction

algorithm. To make the results valid, except for Section 4.2, the average testing accuracy is obtained on 20 trials randomly generated training set and test set. This paper chooses the sigmoid function as the activation function for it is the most commonly used one.

The parameters to set and their meanings in the experiments are listed in Table 1. For example, if the experiment sets $M = 10$, $C = 1$, $L = 1000$, $t = 5$, and $w = 5$, it means that selecting 5 images of each person builds the training set and the remaining images build the test set. Each image is divided into 5×5 windows. After building the training and test set, ELM with $C = 1$, $L = 1000$ and MAELM, which combines 10 ELMs with $C = 1$, $L = 1000$, are evaluated in the built sets.

4.1. Performance Changes with C and L. Although ELM is comparatively not that sensitive to the arguments as SVM, its performance still changes with the hidden layer number L and the constant value C.

Suppose we have N training samples; Huang et al. [1] rigorously prove that SLFNs (with N hidden nodes) with random bias and input weights can exactly learn the N distinct observations. If the training error is allowed, the number of hidden nodes can be much smaller than N. At the same time, the constant value C also has some impacts of the solution of H's Moore-Penrose generalized inverse.

In this part, the experiment is conducted in Yale dataset. The experiment sets $M = 20$, $t = 5$, and $w = 3$. In addition, the L is set as $100, 400, 700, \ldots, 1900$ and the C is set as $10^{-5}, 10^{-4}, \ldots, 1, 10^{1}, 10^{2}, \ldots, 10^{5}$. The performance of ELM and MAELM is shown in Figure 4.

It is obvious that both ELM and MAELM are not sensitive to the change of arguments. The difference between ELM and MAELM is mainly in the region where L is very small and C is very large. From Figure 4, one can conclude that ELM performs badly in this region, since its accuracy rate is below 0.6. On the contrary, MAELM is still very stable in this region. Its accuracy rate is bigger than 0.8.

After seeing PCA's good performance in the region of face recognition, we wonder if PCA could have a stable and better performance when it replaces the way we originally construct the matrix H.

The experiment is also conducted in Yale dataset with the same parameters. Besides, the new parameter r, which is the dimension after reduction, could not be set bigger than the number of input nodes. In view of the dimension of dataset and other limitations in the experiment, the parameter r is set as $10, 20, \ldots, 60$. Since it is complex in the picture because of the imbalance with the parameters change, we choose to show them in the table. The performance of ELM (Figure 4(a)) and MAELM (Figure 4(b)) with PCA is listed in Table 2; the best accuracy rate in the table is bold.

It is clear that both ELM and MAELM with PCA are not so sensitive to the change of arguments. The difference between them is mainly in the region where L is very small and C is very large. From Table 2, one can conclude that MAELM with PCA performs better in this region when C is very small, but when C is large and r is small, ELM with PCA performs rather well and stable. Besides, ELM with PCA's performance is almost as well as the other one in the region where C and r are both very large, and its accuracy rate is bigger than 0.85.

4.2. Prediction Stability Analysis. Since the original ELM randomly generates the weights between the input layer and the hidden layer, as well as the bias of the activation function, its performance even for the same training and test set changes each time. This is to say the performance of

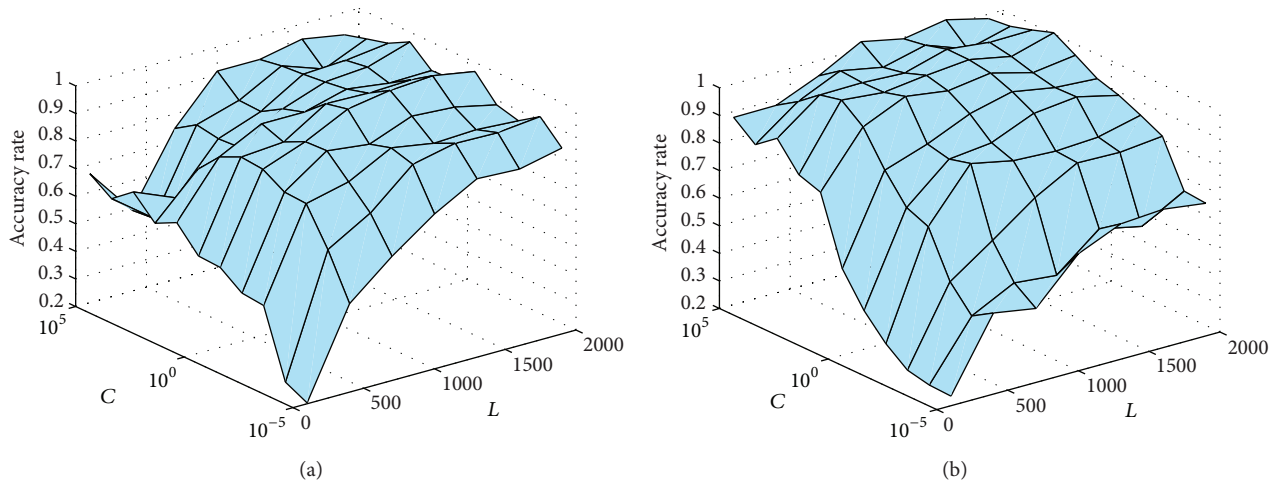

Figure 4: The performance of ELM (a). The performance of MAELM (b).

Table 2: Performance of ELM and MAELM with PCA.

C/r	10	20	30	40	50	60
10^{-5}	0.19/**0.38**	0.29/**0.33**	0.2/**0.32**	0.23/**0.41**	0.09/**0.15**	0.22/**0.30**
10^{-4}	0.19/**0.20**	0.22/**0.40**	**0.21/0.21**	**0.36**/0.35	0.27/**0.28**	0.26/**0.38**
10^{-3}	0.27/**0.39**	**0.38**/0.35	0.24/**0.46**	**0.37**/0.36	**0.33**/0.31	**0.38**/0.30
10^{-2}	**0.43**/0.34	**0.76**/0.32	**0.85**/0.32	**0.86**/0.43	**0.86**/0.37	**0.86**/0.35
10^{-1}	**0.79**/0.36	**0.84**/0.43	**0.88**/0.40	**0.88**/0.36	**0.90**/0.42	**0.91**/0.53
10^{0}	**0.84**/0.58	**0.95**/0.76	**0.86**/0.81	**0.90**/0.85	**0.87**/0.82	**0.91**/0.87
10^{1}	**0.77**/0.66	**0.90**/0.87	**0.89**/0.88	**0.91/0.91**	**0.98**/0.97	**0.91**/0.88
10^{2}	**0.77**/0.73	0.85/**0.87**	**0.91/0.91**	0.90/**0.92**	**0.94**/0.92	**0.92/0.92**
10^{3}	**0.77**/0.74	**0.90**/0.89	0.92/**0.93**	**0.94/0.94**	0.93/**0.95**	0.90/**0.92**
10^{4}	**0.74**/0.55	**0.91**/0.88	**0.93/0.93**	**0.88/0.88**	**0.91/0.91**	0.85/**0.86**
10^{5}	**0.79**/0.71	**0.87/0.87**	**0.87/0.87**	**0.97/0.97**	**0.96/0.96**	**0.91/0.91**

original ELM may not be so stable. The proposed algorithm successfully reduces the instability.

From Figure 4, one is able to conclude that ELM tends to get better performance when $C = 1$, while $C = 10^3$ is better for MAELM. Let $M = 10$, $C = 10^3$, $L = 1000$, $t = 5$, and $w = 3$ for MAELM and $C = 1$, $L = 1000$, $t = 5$, and $w = 3$ for ELM. Besides, ELM and MAELM ($r = 20$) with PCA are also included under the corresponding situations because of the considerate performance above. Experiments are done in Yale datasets. In order to prove that the proposed algorithm is more stable than the original ELM, experiments are done in the same training set and test set (randomly generated) for 20 times. The result is shown in Figure 5.

In Figure 5, it is obvious that the performance of MAELM is much more stable than the original ELM. Although ELM or MAELM with PCA performs far more stable than the original ELM and MAELM (since they take the algorithm of PCA into consideration instead of the random weights between the input layer and the hidden layer and the bias of the activation function), the accuracy rates of them, which are always in the middle from Figure 5, are still not so good as the original MAELM. We conclude the result of Figure 5 in Table 3. Please notice that although the generalization performance of MAELM seems to be much better than ELM in the table, it is improper to conclude that MAELM performs better. The reason is that the training set and test set are fixed. One cannot exclude the possibility that MAELM performs better than ELM only under this dataset. Some other experiments will be done in the following parts to show MAELM's better generalization performance.

Table 3: Performance of ELM and MAELM under the same training set and test set.

Algorithm	Mean accuracy rate	Standard derivation
ELM	0.8972	0.0213
MAELM	0.9361	0.0157
ELM_PCA	0.9222	0
MAELM_PCA	0.9222	0

4.3. Performance Changes with M. In order to evaluate the changes of performance when M changes, the experiment in this part lets $C = 1$, $t = 5$, $w = 4$, $L = 1000$ for the original MAELM, $r = 20$ for MAELM with PCA, and $M = 2, 4, 6, \ldots, 50$. The average test accuracy is obtained on

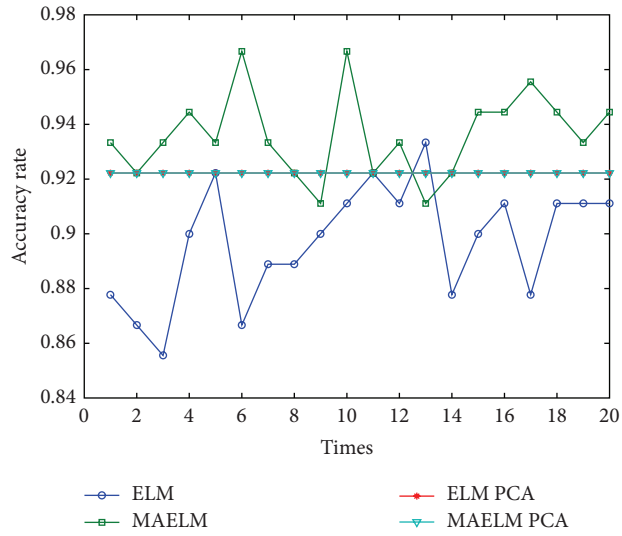

Figure 5: Performance of ELM and MAELM under the same training set and test set.

20 trials randomly generated training set and test set. Yale dataset is used for the experiment. The result is presented in Figures 6 and 7.

From Figure 6, it is obvious that as the M increases, the generalization performance also becomes better. However, the trend becomes slower as M increases. From Figure 7, one can conclude that as the M increases when M is small, the performance decreases a little, while M becomes larger after 25; the performance also becomes better, although the trend is not so stable as the original MAELM. This situation indicates that in real-world applications, M does not need to be very big. Good generalization performance can be obtained by setting M less than 30 in the algorithm of original MAELM, which achieves better than MAELM with PCA under the same situation.

4.4. Better Generalization Performance Than ELM. In this part, experiments are done both in Yale and ORL datasets. The experiments set the parameters of those algorithms as follows: $C = 1$, $L = 1000$, $t = 5$, $M = 20$ (MAELM), and $r = 20$ (PCA). The experiments take 3×3, 4×4, 5×5, 6×6, and 7×7 windows into consideration, which means setting $w = 3, \ldots, 7$. The average testing accuracy is obtained on 20 trials randomly generated training set and test set.

The experiment indicates that MAELM has better generalization performance both in Yale and ORL datasets under different window sizes. See Figure 8 for details, while in Figure 9, it is obvious that ELM with PCA has much better performance both in Yale and ORL datasets under different window sizes. In addition this algorithm keeps more stable than any other algorithms both in Yale and ORL datasets.

4.5. The Performance in PCA. After seeing all these experiments, we can conclude that although MAELM with PCA performs not so well as the original one, ELM with PCA performs much better than before, especially in the experiment in Section 4.2. It is obvious that the performance of the experiments with PCA is just between the original ELM and MAELM.

What is more, since the original ELM randomly generates the weights between the input layer and the hidden layer, as well as the bias of the activation function, its performance is not so stable. The proposed algorithm with PCA successfully reduces the instability which is very important in the real world.

Although PCA improves the performance of ELM in a certain degree, it still could not reach the ability of MAELM with random weights and bias. Finally, it comes to the result that the proposed algorithm named MAELM performs much better in solving the multiclass classification problem.

5. Discussion

5.1. Complexity Comparison. Very similar to MAELM, the DAEELM [8] also considers taking the ELM as the weak classifier and implements AdaBoost as the ensemble method. The difference is that MAELM implements multiclass AdaBoost which can be directly used in multiclass classification problem, while DAEELM implements dynamic ensemble AdaBoost [23], which aims to solve the binary classification problem.

Many methods have been developed to apply binary classifier to multilabel problem. One-against-all (OAA) [24] and one-against-one (OAO) [25] are mostly used. For a K-class classification problem, under OAA condition, K classifiers have to be trained. Each of them separates a single class from all the remaining classes. Under the OAO condition, $K(K-1)/2$ classifiers have to be trained. Each of them separates a pair of classes.

Suppose that both MAELM and DAEELM have M iterations. For a K-class classification problem, MAELM only needs to train M ELMs, while DAEELM needs to train $M \times K$ and $(M \times K \times (K-1))/2$ classifiers for OAA and OAO condition, respectively. Although DAEELM may stop the iteration earlier, it is obvious that, in theory, MAELM's computation complexity is much lower than DAEELM for K-class classification problem, especially when K is a very big number.

The authors of DAEELM have not published its codes and DAEELM has its own arguments which MAELM does not have. DAEELM also does not provide details of how it trains weighted data with ELM, so it will be unfair to compare the performance of MAELM and DAEELM. However, the conclusion that MAELM is much faster than DAEELM in multiclass classification problem can be drawn from the complexity analysis above.

5.2. Train ELM with Weighted Data. Section 3.1 has mentioned that training ELM with weighted data is a key problem when applying AdaBoost. However, [8, 9] did not mention the key point at all.

Toh in [26] first applied ELM to classify imbalanced data with two classes. ELM tries to minimize the training error of

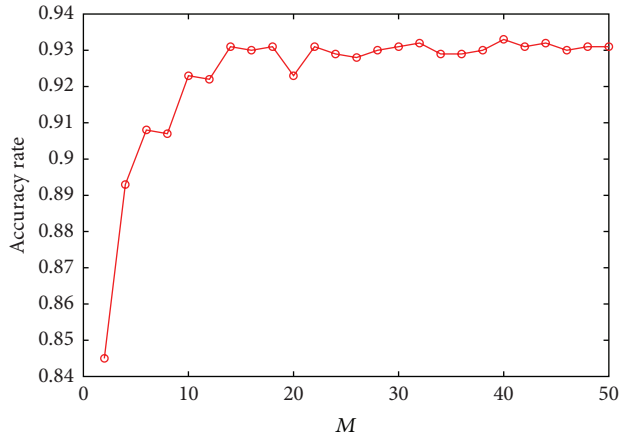

Figure 6: MAELM's performance.

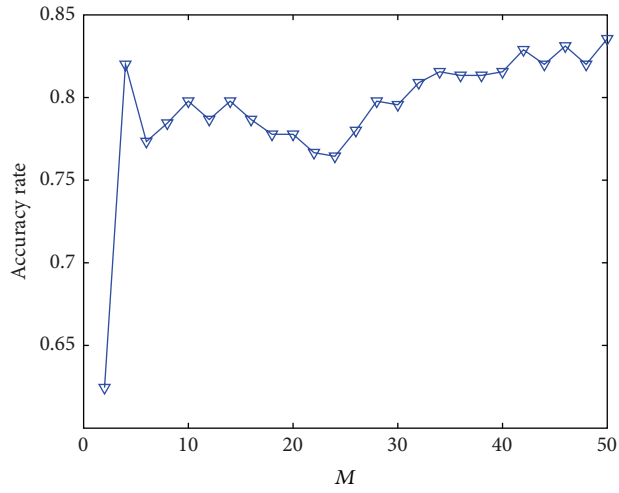

Figure 7: MAELM with PCA's performance.

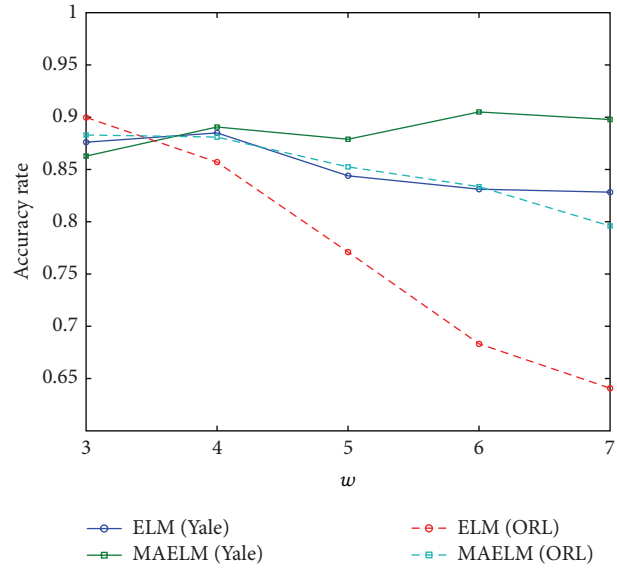

Figure 8: Performances in Yale and ORL.

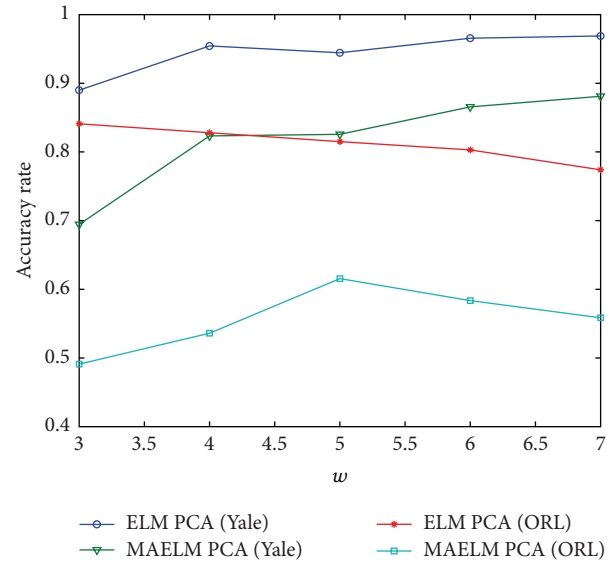

Figure 9: Performances in Yale and ORL.

the data while the proposed algorithm tends to minimize the total error rate (TER), which takes the weights of the positive and negative data into consideration.

In Section 3.1, the weighted ELM is applied in MAELM. Actually, the weighted ELM is inspired and in a way that is very similar to regularized ELM proposed by Deng et al. in [27]. The regularized ELM aims to minimize the weighted training error of the weighted data.

6. Conclusion

This paper proposes a new boosting ELM named MAELM, which applies the multiclass AdaBoost in ELM ensemble to directly solve multiclass classification problem. A face recognition structure combined LBP based method and ELM is also presented in the paper. What is more, this paper proposes the way in which ELM combined with PCA instead of using random weights between the input layer and the hidden layer, as well as the bias of the activation function.

Experiments in LBP based face recognition will show the stable and good performance in a certain degree. Although PCA improves the performance of ELM, it still could not be better than MAELM with random weights and bias. Experiments show that in LBP based face recognition problem, the recognition result of MAELM is more stable than the original ELM and better than any other algorithms listed in the paper.

Finally, it comes to the result that the proposed algorithm named MAELM, which applies the multiclass AdaBoost in ELM and combines with LBP method, performs much better in solving the multiclass classification problem.

Also, MAELM is compared with DAEELM in multiclass classification problem in theory, which indicates that MAELM has much lower computation complexity than DAEELM. Moreover, this paper makes the problem how to train weighted data by ELM clear.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research is based on work supported in part by the National Natural Science Foundation of China (61370173, 61173123) and the Natural Science Foundation Project of Zhejiang Province under Project LR13F030003.

References

[1] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1–3, pp. 489–501, 2006.

[2] G.-B. Huang, "An insight into extreme learning machines: random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.

[3] P. K. Wong, Z. Yang, C. M. Vong, and J. Zhong, "Real-time fault diagnosis for gas turbine generator systems using extreme learning machine," *Neurocomputing*, vol. 128, pp. 249–257, 2014.

[4] J. W. Cao, T. Chen, and J. Fan, "Fast online learning algorithm for landmark recognition based on BoW framework," in *Proceedings of the 9th IEEE Conference on Industrial Electronic and Application*, pp. 1163–1168, 2014.

[5] J. Cao and L. Xiong, "Protein sequence classification with improved extreme learning machine algorithms," *BioMed Research International*, vol. 2014, Article ID 103054, 12 pages, 2014.

[6] J. Cao, Z. Lin, G.-B. Huang, and N. Liu, "Voting based extreme learning machine," *Information Sciences*, vol. 185, no. 1, pp. 66–77, 2012.

[7] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: a statistical view of boosting," *The Annals of Statistics*, vol. 28, no. 2, pp. 337–407, 2000.

[8] G. Wang and P. Li, "Dynamic Adaboost ensemble extreme learning machine," in *Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE '10)*, pp. V3-54–V3-58, IEEE, Chengdu, China, August 2010.

[9] H.-X. Tian and Z.-Z. Mao, "An ensemble ELM based on modified AdaBoost.RT algorithm for predicting the temperature of molten steel in ladle furnace," *IEEE Transactions on Automation Science and Engineering*, vol. 7, no. 1, pp. 73–80, 2010.

[10] D. P. Solomatine and D. L. Shrestha, "AdaBoost.RT: a boosting algorithm for regression problems," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 2, pp. 1163–1168, 2004.

[11] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[12] T. Ahonen, A. Hadid, and M. Pietikäinen, "Face recognition with local binary patterns," in *Computer Vision-ECCV 2004*, vol. 3021 of *Lecture Notes in Computer Science*, pp. 469–481, Springer, Berlin, Germany, 2004.

[13] Y. Shen, Y. L. Jiang, W. Liu, and Y. Liu, "Multi-class AdaBoost ELM," in *Proceedings of the International Conference on Extreme Learning Machines (ELM '14)*, Singapore, December 2014.

[14] G. B. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 42, no. 2, pp. 513–529, 2012.

[15] K. Pearson, "On lines and planes of closest fit to systems of points in space," *Philosophical Magazine*, vol. 11, no. 2, pp. 559–572, 1901.

[16] H. Hotelling, "Analysis of a complex of statistical variables into principal components," *Journal of Educational Psychology*, vol. 24, no. 7, pp. 498–520, 1933.

[17] H. Abdi and L. J. Williams, "Principal component analysis," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 4, pp. 433–459, 2010.

[18] W. Zong, G.-B. Huang, and Y. Chen, "Weighted extreme learning machine for imbalance learning," *Neurocomputing*, vol. 101, pp. 229–242, 2013.

[19] W. Zong and G.-B. Huang, "Face recognition based on extreme learning machine," *Neurocomputing*, vol. 74, no. 16, pp. 2541–2551, 2011.

[20] A. A. Mohammed, R. Minhas, Q. M. Jonathan Wu, and M. A. Sid-Ahmed, "Human face recognition based on multidimensional PCA and extreme learning machine," *Pattern Recognition*, vol. 44, no. 10-11, pp. 2588–2597, 2011.

[21] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.

[22] K. Etemad and R. Chellappa, "Discriminant analysis for recognition of human face images," *Journal of the Optical Society of America A*, vol. 14, no. 8, pp. 1724–1733, 1997.

[23] R. Li, J. Lu, Y. Zhang, and T. Zhao, "Dynamic Adaboost learning with feature selection based on parallel genetic algorithm for image annotation," *Knowledge-Based Systems*, vol. 23, no. 3, pp. 195–201, 2010.

[24] B. Heisele, P. Ho, J. Wu, and T. Poggio, "Face recognition: component-based versus global approaches," *Computer Vision and Image Understanding*, vol. 91, no. 1-2, pp. 6–21, 2003.

[25] E. L. Allwein, R. E. Schapire, and Y. Singer, "Reducing multiclass to binary: a unifying approach for margin classifiers," *The Journal of Machine Learning Research*, vol. 1, no. 2, pp. 113–141, 2001.

[26] K.-A. Toh, "Deterministic neural classification," *Neural Computation*, vol. 20, no. 6, pp. 1565–1595, 2008.

[27] W. Deng, Q. Zheng, and L. Chen, "Regularized extreme learning machine," in *Proceedings of the IEEE Symposium on Computational Intelligence and Data Mining (CIDM '09)*, pp. 389–395, Nashville, Tenn, USA, April 2009.