Neurocomputing 135 (2014) 253–261

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Modified minimum squared error algorithm for robust classification and face recognition experiments

Yong Xu [a,b], Xiaozhao Fang [a], Qi Zhu [a], Yan Chen [a,c,*], Jane You [d], Hong Liu [e]

[a] Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China
[b] Key Laboratory of Network Oriented Intelligent Computation, Shenzhen, China
[c] Shenzhen Sunwin Intelligent Corporation, Shenzhen, China
[d] Biometrics Researcher Centre, Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China
[e] Engineering Laboratory on Intelligent Perception for Internet of Things, Shenzhen Graduate School, Peking University, Shenzhen, China

ARTICLE INFO

Article history:
Received 4 June 2013
Received in revised form
25 September 2013
Accepted 25 November 2013
Communicated by D. Tao
Available online 13 February 2014

Keywords:
Minimum squared error (MSE)
Pattern recognition
Face recognition

ABSTRACT

In this paper, we improve the minimum squared error (MSE) algorithm for classification by modifying its classification rule. Differing from the conventional MSE algorithm which first obtains the mapping that can best transform the training sample into its class label and then exploits the obtained mapping to predict the class label of the test sample, the modified minimum squared error classification (MMSEC) algorithm simultaneously predicts the class labels of the test sample and the training samples nearest to it and combines the predicted results to ultimately classify the test sample. Besides this paper, for the first time, proposes the idea to take advantage of the predicted class labels of the training samples for classification of the test sample, it devises a weighted fusion scheme to fuse the predicted class labels of the training sample and test sample. The paper also interprets the rationale of MMSEC. As MMSEC generalizes better than conventional MSE, it can lead to more robust classification decisions. The face recognition experiments show that MMSEC does obtain very promising performance.

1. Introduction

The minimum squared error algorithm has been widely used for pattern classification. The minimum squared error classification (MSEC) takes the sample and its class label as the input and output respectively, and tries to obtain the mapping that can best transform the input into the corresponding output. MSEC first uses the training samples to perform training and then exploits the obtained mapping to predict the class label of the test sample. Finally, MSEC assigns the test sample into the class whose class label is most similar to the predicted class label of the test sample.

MSEC not only can achieve high accuracy but also holds good properties. For example, it has been proven that for two-class classification MSEC is identical to linear discriminant analysis (LDA) under the condition that the number of training samples approximates the infinity [1,2]. LDA and its variants have been widely used [3]. Moreover, if a special class indicator matrix is used, MSEC and LDA are also equivalent for multi-class classification [4]. LDA has also been shown to be equivalent to canonical correlation analysis (CCA) for multi-class classification [5]. As a result, MSEC will perform very similarly as CCA in multi-class classification [6].

Besides MSEC has been extended to multi-class classification, a well-known nonlinear extension of MSEC, kernel MSE (KMSE), has been proposed. KMSE performs very well in the field of pattern recognition too [2,7,8]. Other various improvements to the MSE methodology have also been devised. For example, "Lasso" based MSE (LBMSE) was recently proposed for classification [9–11]. LBMSE tries to obtain good generalization performance by minimizing the l_1 norm of the solution vector and can be viewed as an extension of conventional MSEC. Differing from conventional MSEC, LBMSE takes the training sample and the test sample themselves as the input and the output, respectively. After the mapping between the input and output is constructed, LBMSE also uses a way different from that of MSEC to perform classification. As shown in Refs. [12–14], we can also modify MSEC to a classification algorithm that is similar to LBMSE but subject to the constraint of minimizing the l_2 norm of the solution vector. This algorithm will be computationally more efficient than LBMSE and has comparable classification performance. Linear regression classification (LRC) proposed in Ref. [15] is a typical example of this kind of algorithm. The MSEC algorithms with the constraints of minimizing the l_1 or l_2 norm can also be referred to as

* Corresponding author at: Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen, China. Tel.: +86 755 86169696; fax: +86 755 86169393.
E-mail address: jadechenyan@gmail.com (Y. Chen).

http://dx.doi.org/10.1016/j.neucom.2013.11.025
0925-2312

penalized MSECs [16] or representation-based classification (RBC) algorithms.

Besides the inputs and outputs of the method proposed in our paper are different from those of RBC, it also differs from RBC as follows. The proposed method should solve only one equation and exploit it to predict the class label of all the test samples. However, RBC must solve at least one equation for classifying a test sample. In particular, RBC proposed in Refs. [12–14] should solve and exploit one equation for classifying a test sample. LRC should depend on the solutions of c equations to classify a test sample. c is the number of the classes. As a result, our proposed method is usually computationally more efficient than RBC.

The total least squares (TLSs) [17,18] is another well-known improvement to the MSE. TLS assumes that both the input and output are corrupted and each of them can be expressed as the sum of the corresponding "true data" and "measurement noise". Differing from TLS, conventional MSE methods just assumes that the output is corrupted but the input is not. Based on TLS, researchers also proposed the weighted and structured total least squares (WSTLSs) [17–20]. WSTLSs are usually numerically solved by using local optimization methods [17]. In addition, recursive least-squares methods were proposed as reinforcement learning algorithms [21]. Two-stage least squares (2SLS) was proposed for latent variable models [22]. Bayesian minimum mean-square error was also proposed to explore the theoretical issue in pattern classification such as to estimate the classification error [23–25]. In addition, some means such as the regularized term was also used to improve the numerical stability of MSE [26]. The means of regularization is indeed widely used and Hessian regularization proposed in Ref. [27] obtained very good performance in image annotation. Orthogonal MSE [28] and computationally more efficient MSE algorithm [29–31] were also devised. Besides pattern classification [32], the minimum squared error algorithms have been applied to other fields such as density estimation, clustering, feature extraction, data fitting and regression as well as image coding [7,17,30,31,33–36]. We also note that MSE has been widely used in the field of signal processing for resolving some important problems such as direction estimation, estimation of deterministic parameters with noise covariance uncertainties, optimization of the downlink multiuser MIMO systems and multipath channel estimations [37–39]. The MSE algorithm was also used for other issues such as Kalman filters and probabilistic principal component analysis [40]. The naïve MSE algorithm and its variants have been also widely used in regression [41,42].

Researchers have also paid much attention to improve the generalization performance of the classification algorithm. For MSEC, a conventional and important way to improve the generalization performance is to impose the constraint of minimizing the norm especially the l_2 norm of the solution vector on it. Of course, this way is very useful for avoiding the case where the predicted class label of the test sample corrupted by little noise greatly deviates from its true class label. However, the above way still cannot perform well in the case where the test sample is corrupted by great noise. For example, in real-world face recognition applications the test sample might be very different from the training sample from the same subject owing to varying expression, pose and illumination [43–45]. Consequently, the predicted class label of the test sample might have large deviation from its true class label. However, we see that the predicted class label of the training sample is always very close to its true class label. This somewhat means that the MSEC algorithm has great confidence in predicting the class label of the training sample but has less confidence in predicting the class label of the test sample. As a result, if a training sample is very near to the test sample, it is reasonable to integrate the predicted class labels of this training sample and the test sample to classify the test sample.

In this paper, in order to obtain more robust MSEC algorithm, we improve the MSEC algorithm by modifying its classification rule. We establish the same equation as that of the conventional MSEC and also solve it in the same way. Then we exploit the obtained solution to simultaneously predict the class labels of the test sample and the training samples nearest to it and combine the predicted results to ultimately classify the test sample. We use a weighted fusion scheme to combine the predicted class labels of the test sample and the training samples. The weight of the test sample is assigned a larger value in comparison with those of the training samples. When more than one training sample are exploited, we also assign a larger coefficient to the training sample that is closer to the test sample. The experiments also show that MMSEC does obtain much higher classification accuracy than conventional MSEC. This paper has the following noticeable contributions. First, it for the first time proposes the idea to take advantage of the predicted class labels of the training samples to classify the test sample. It also carefully demonstrates the underlying rationale of MMSEC. Second, it devises a weighted fusion scheme to fuse the predicted class labels of the training sample and test sample.

2. The minimum squared error classification (MSEC)

In this section we take the multi-class problem as an example to describe MSEC. Suppose that there are c classes. We assign a class label to each class. If a mapping is able to transform a sample into its class label and we can get this mapping by learning, then we can exploit the learned mapping to predict the class label of each test sample. Let x_i be a p-dimensional row vector and denote the ith training sample, $i = 1, \ldots, N$. N is the total number of the training samples. We use a c-dimensional vector to represent the class label. If a sample is from the first class, we take $g = [1 \quad 0 \quad . \quad . \quad . \quad 0]$ as its class label. If a sample is from the cth class, we take $g = [0 \quad . \quad . \quad . \quad 0 \quad 1]$ as its class label. In other words, if a sample is from the kth class, then the kth element of its class label is one and the other elements are all zeroes. This class label is also referred to as the class label of the kth class.

Assuming that matrix Y can approximately transform each training sample into its class label, MSEC has the following equation:

$$XY = G \tag{1}$$

where

$$X = \begin{bmatrix} x_1 \\ . \\ . \\ . \\ x_N \end{bmatrix}, \; G = \begin{bmatrix} g_1 \\ . \\ . \\ . \\ g_N \end{bmatrix}$$

It is clear that X is an $N \times p$ matrix, G is an $N \times c$ matrix, and Y is a $p \times c$ matrix. We refer to Y as transform matrix. g_i is the class label of the ith training sample.

As Eq. (1) cannot be directly solved, we convert it into the following equation:

$$X^T XY = X^T G \tag{2}$$

We can obtain Y using

$$\overline{Y} = (X^T X + \gamma I)^{-1} X^T G \tag{3}$$

where γ and I denote a small positive constant and the identity matrix, respectively. MSEC classifies a test sample x in the form of row vector as follows: the class label of x is first predicted using $g_x = x\overline{Y}$. Then the distances between g_x and the class labels of all the c classes are calculated. As shown above, the class label of the

jth class is a row vector whose jth element is one and whose other elements are all zeros ($j=1,\ldots,c$). If g_x is the closest to the class label of the kth class, then x will be classified into the kth class.

3. The algorithm of modified minimum squared error classification (MMSEC)

The key of MMSEC is to combine the predicted class labels of the test sample and the training sample nearest to it to classify the test sample. Moreover, the training phase of MMSEC is the same as that of MSEC. Let t be a test sample in the form of the column vector. The algorithm of MMSEC includes the following steps.

Step 1. Establish equation $XY=G$ and solve it using Eq. (3).
Step 2. Use $t'=t^T\overline{Y}$ to predict the class label of test sample t. t' denotes the so-called predicted class label. Calculate the Euclidean distance between t' and the class label of each class. Let $d_j=||t'-la_j||$ denote the distance between t' and the jth class. la_j denotes the class label of the jth class, which is defined in Section 2.
Step 3. Among all training samples the K training samples that are nearest to the test sample in terms of the Euclidean distance are first chosen. Let $q_1,\ldots,q_K$ denote these K training samples. Let $cl_1,\ldots,cl_K$ be the predicted class labels of $q_1,\ldots,q_K$, respectively. For q_j, let $s_j^m=||cl_j-la_m||$ stand for the dissimilarity between q_j and the mth class. la_m still denotes the class label of the mth class. Use $S_t^m=\Sigma_{j=1}^K\beta_j s_j^m$ to denote the dissimilarity between these K training samples and the mth class. Coefficients β_j are set to $\beta_j=1-(dis_j/\Sigma_{i=1}^K dis_i)$, $j=1,\ldots,K$. dis_j stands for the Euclidean distance between q_j and the test sample. If $K=1$, then there is only one weight β_1 and we set it as $\beta_1=1$.
Step 4. Let $e_j=w_1 d_j+w_2 S_t^j$ stand for the "distance" between the jth class and the test sample. $w_1+w_2=1$ and w_1,w_2 are the weights of d_j, S_t^j, respectively. If $r=\arg\min_j e_j$, then test sample t is assigned into the rth class.

Among the above steps of MMSEC, Step 1 indeed completes the training phase and the other steps are devised for classifying the test sample. From Step 3, we know that when more than one training sample are exploited, a larger coefficient will be assigned to the training sample that is closer to the test sample. This is an easily understood strategy owing to the fact that the closer to the test sample the training sample is, the more possible from the same class as the test sample the training sample is usually. In order to realize this strategy, Step 3 sets β_j using $\beta_j=1-(dis_j/\Sigma_{i=1}^K dis_i)$, $j=1,\ldots,K$. Another rationale of this formulation is that it enables S_t^m to be weighted average of s_j^m and the sum of all the weight coefficients equal to 1, i.e. $\beta_1+\cdots+\beta_K=1$. Moreover, this will make S_t^m not greatly deviate from all s_j^m and become a proper and reasonable "mean" of s_j^m.

When implementing MMSEC, we suggest that w_1 is set to a larger value than w_2. The underlying reason is that the experimental analysis shows that to solely exploit the predicted class label of the test sample to classify it usually obtains higher accuracy than to solely exploit the predicted class labels of the training samples nearest to the test sample to classify it.

4. Analysis of the proposed method

4.1. Difference between MMSEC and MSEC

MSEC and the proposed method i.e. MMSEC have the following difference. MSEC seems to be optimal for all the training samples from various classes. In other words, when MSEC simultaneously maps the data of all the training samples into their own class labels, it indeed tries to minimize the sum of the deviation between the obtained class labels and the true class labels. Thus we say that MSEC is able to well convert every training sample into the true class label. However, this does not imply that MSEC can also very well convert every test sample into its true class label. As the test sample data may be viewed as the sum of its true observation and the noise and the noise is disadvantageous for correctly predicting the class label of the test sample, MSEC might erroneously classify the test sample in some cases especially in the case where the noise is very large.

We use Figs. 1–3 to show the predicted errors of the training samples and test samples obtained using MSEC on the face database. For sample q in the form of row vector, the predicted error is defined as $||true_q-q\overline{Y}||$. $true_q$ stands for the true class label of q. From these figures, we see that the training samples always have smaller predicted errors than the test samples. The large predicted error of the test sample somewhat implies that MSEC somewhat has a low confidence in predicting the class label of the test sample! Because the training samples usually have much smaller predicted errors, it is very reasonable to combine the predicted class labels of the test sample and the training samples nearest to it to perform classification.

4.2. Insight into the rationale of MMSEC

In this subsection we will in-depth analyze the rationale of MMSEC. Since MMSEC simultaneously exploits the test sample and nearest training samples to predict the class label of the test

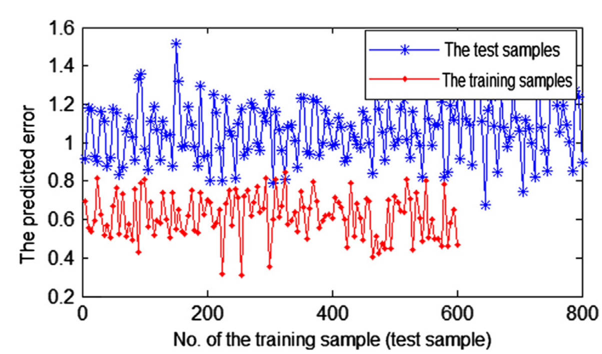

Fig. 1. The predicted errors of the training samples and test samples obtained using MSEC on the subset of the FERET face database shown in Section 5. The first four face images of each subject and the remaining images were used as the test samples and training samples, respectively. The horizontal and vertical axes show the Nos. of the training samples (test samples) and the predicted errors, respectively.

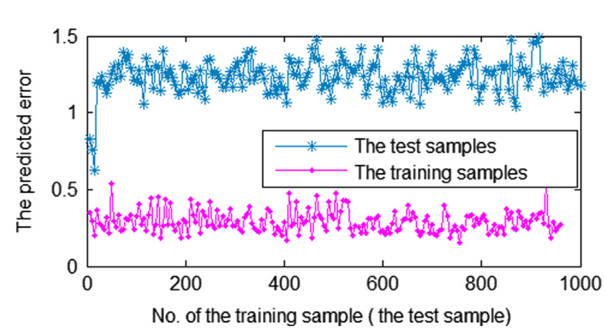

Fig. 2. The predicted errors of the training samples and test samples obtained using MSEC on the AR face database. The first 18 face images of each subject and the remaining images were used as the test samples and training samples, respectively. The horizontal and vertical axes show the Nos. of the training samples (test samples) and the predicted errors, respectively. The predicted errors of all the training samples and the first 1000 test samples are shown.

sample, the predict result seems to be more robust than that of MSEC. In particular, it is easy to know that in the case where the used "nearest" training samples are really from the same class as the test sample, MMSEC must more accurately classify the test sample than MSEC. The underlying reason is that the predicted class labels of the nearest training samples will be almost same as the true class label of the test sample. This will greatly increase the probability that the test sample is assigned to the correct class.

The rationale to exploit the predicted class label of the training sample nearest to the test sample for classification can be formally presented as follows: if the ith training sample x_i is nearest to test sample t and has the same true class label as t, we assume that $t=x_i+\Delta x$. Δx stands for the deviation between t and x_i. Suppose that only the predicted class label of x_i is combined with that of t to ultimately classify t. Let $F(.)$ stand for the mapping to transform the sample into its class label, then $F(t)$ and $F(x_i)$ are the predicted class labels of test sample t and the ith training sample, respectively. It is clear that $F(.)$ is a linear mapping. Thus the predicted class label of t obtained using MSEC can be written as $F(t)=F(x_i)+F(\Delta x)$. As we know, the predicted class label of the training sample is usually extremely close to its true class label. It is clear that if $F(\Delta x)$ has a relatively small value, then $F(x_i)$ will well approximate the predicted class label of t i.e. $F(x_i)\approx F(t)$. However, if $F(\Delta x)$ is great enough, the predicted class label of test sample t obtained using MSEC will be very different from that of the training sample x_i nearest to t.

Now we show that Step 4 of our proposed method is able to alleviate the influence of deviation Δx. The predicted class label of t obtained using MMSEC is $w_1F(t)+w_2F(x_i)=(w_1+w_2)F(x_i)+w_1F(\Delta x)=F(x_i)+w_1F(\Delta x)$. Because $0<w_1<1$, it is clear that $F(x_i)+w_1F(\Delta x)$ is nearer to the true class label of test sample than $F(x_i)+F(\Delta x)$. As $w_1<1$ and $F(x_i)+w_1F(\Delta x)$ and $F(x_i)+F(\Delta x)$ are respectively the predicted class labels of t obtained using MMSEC and MSEC, we know that MMSEC can more correctly classify t than MSEC under the condition that x_i has the same true class label as test sample t. Actually, because of $F(x_i)+w_1F(\Delta x)-true_t\approx w_1F(\Delta x)$, $F(x_i)+\ F(\Delta x)-true_t\approx F(\Delta x)$ and $w_1||F(\Delta x)||<||F(\Delta x)||$, we can conclude that if test sample t has the same true class label as training sample x_i and x_i is nearest to t, then to combine the predicted class labels of the training sample and test sample will be very beneficial to correctly classify the test sample.

MMSEC partially owns the advantages of conventional MSE and nearest neighbor classifier. MMSEC is somewhat equivalent to a procedure that slightly modifies the classification results of conventional MSE by using the nearest neighbor classifier. In particular, MMSEC exploits $w_1F(t)+w_2F(x_i)$ to obtain the class label of the test sample. $F(t)$ is the result of conventional MSE and $F(x_i)$ can be partially viewed as the result of the nearest neighbor classifier. The face recognition experiment will demonstrate that MMSEC can obtain better performance than conventional MSE.

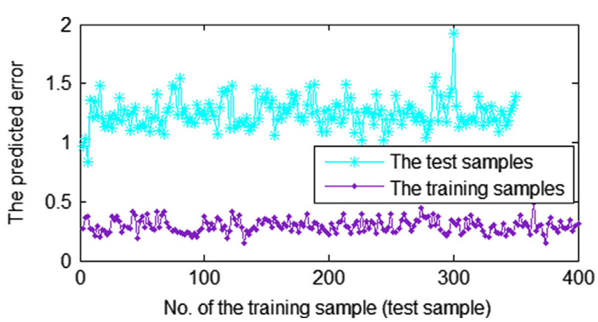

Fig. 3. The predicted errors of the training samples and test samples obtained using MSEC on the GT face database. The first eight face images of each subject and the remaining images were used as the training samples and test samples, respectively. The horizontal and vertical axes show the Nos. of the training samples (test samples) and the predicted errors, respectively.

5. Experimental results

We use three face databases to test our method and MSEC. We also test two other MSE methods, the collaborative representation classification (CRC) proposed in Ref. [12] and the relaxed collaborative representation (RCR) proposed in Ref. [46]. CRC and RCR have shown good performance in face recognition. For simplicity of presentation, we will show only the experimental results of our method with $w_1=0.75$, $w_2=0.25$ and $w_1=0.8$, $w_2=0.2$, respectively. The experimental results will illustrate that our method outperforms MSEC and the other methods. Figs. 4–6 show some of the test samples which were correctly and erroneously classified by MMSEC and MSEC on these face databases, respectively.

5.1. Experiments on the Georgia Tech face database

In this subsection we use the Georgia Tech face database [47] to test our method. Georgia Tech face database (GTFB) was built at Georgia Institute of Technology. GTFB contains images of 50 people taken in two or three sessions. All people in the database were represented by 15 color JPEG images with cluttered background taken at the resolution of 640×480 pixels. The pictures show frontal and/or tilted faces with different facial expressions, lighting conditions and scale. Each image was manually labeled to determine the position of the face in the image. We use the face images

Fig. 4. Some of the test samples which are from the subset of the FERET database and were correctly and erroneously classified by MMSEC with $K=3$ and MSEC, respectively. The first row shows these test samples. The second row shows a training sample of the subject to which the test sample was erroneously assigned by MSEC. The first three face images of each subject and the remaining images are used as test samples and training samples, respectively.

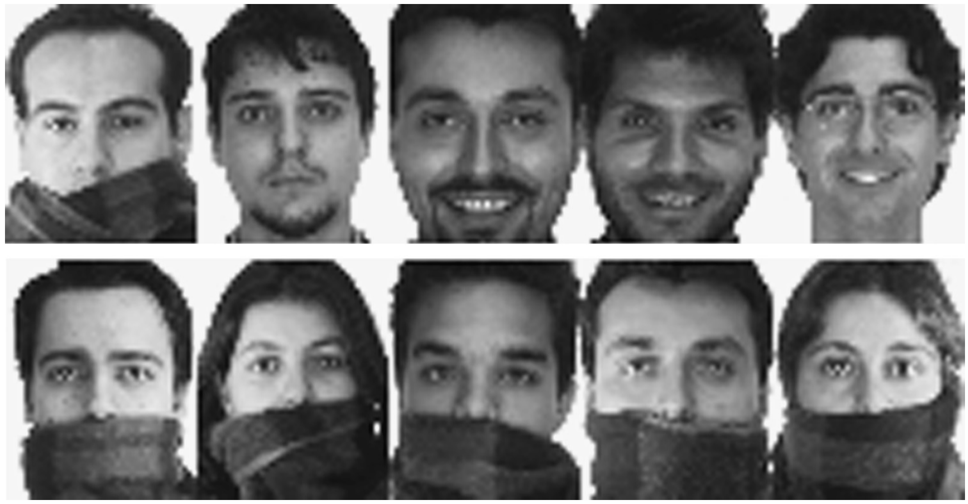

Fig. 5. Some of the test samples which are from the subset of the AR database and were correctly and erroneously classified by MMSEC with $K=1$ and MSEC, respectively. The first row shows these test samples. The second row shows a training sample of the subject to which the test sample was erroneously assigned by MSEC. The first five face images of each subject and the remaining images are used as test samples and training samples, respectively.

Fig. 6. Some of the test samples which are from the GTFB database and were correctly and erroneously classified by MMSEC with $K=3$ and MSEC, respectively. The first row shows these test samples. The second row shows a training sample of the subject to which the test sample was erroneously assigned by MSEC. The first eight face images of each subject and the remaining images are used as training samples and test samples, respectively.

with the background removed and each of these face images has the resolution of 40×30 pixels. They are all converted into gray images in advance. The first three, four, …, or 12 face images of each subject are used as training samples and the remaining images are taken as test samples. Table 1 shows the experimental results. From this table, we see that our proposed method, MMSEC, obtains a much lower rate of classification errors than MSEC, CRC and RCR. For example, when the first eight face images of each subject and the remaining images are used as the training samples and test samples respectively, MMSEC with $w_1=0.75$ and $K=1$ obtains a rate of classification errors of 32.29%. However, the rates of classification errors of RCR, CRC and MSEC are 48.57%, 46.00% and 41.14%, respectively.

As dimension reduction is a widely used preprocessing method for high-dimensional data [48–53] such as face image, in this subsection we also conduct experiments based on dimension reduction. We first use principal component analysis (PCA) to reduce the dimension of the sample and then apply MMSEC, MSEC, CRC and RCR to the obtained low-dimensional sample. PCA is used to extract N dimensional features from every original sample. N still stands for the total number of training samples. The experimental results shown in Table 2 also illustrate that MMSEC outperforms the other methods.

5.2. Experiments on the FERET face database

We also use a subset of the FERET face database [54] to test our method. This subset is composed of 1400 images from 200 individuals with each subject providing seven images. This subset includes the face images whose names contain two-character strings: “ba”, “bj”, “bk”, “be”, “bf”, “bd”, and “bg”. The images in this subset have pose variations of $\pm15^\circ$, $\pm25^\circ$, and also the variations of the illumination and expression. We take the first two, three and four images of each subject as test samples and take the remaining images as training samples. As a result, in these experiments the number of the training samples per subject is 3, 4 and 5. We use the down-sampling algorithm to resize each image into a 40×40 image before the experiment is performed. Table 3 shows that our proposed method usually classifies more accurately than MSEC, CRC and RCR.

5.3. Experiments on the AR face database

We also use the AR face database [55] to test our method. There are 3120 gray images from 120 subjects. Every subject provides 26 frontal view face images with different facial expressions, conditions of illumination, and occlusions (sun glasses and scarf). These images were taken in two sessions, separated by intervals of 2

Table 1
Rate of classification errors (%) of different methods on the GT face database.

Number of the original training samples per class	3	4	5	6	7
RCR	63.33	61.64	59.20	55.33	49.75
CRC	58.00	58.73	55.80	50.44	48.25
MSEC	55.17	53.82	51.40	36.44	35.25
MMSEC ($w_1=0.75$, $K=1$)	51.00	48.00	45.60	36.44	35.25
MMSEC ($w_1=0.75$, $K=2$)	49.83	47.82	44.80	37.11	33.75
MMSEC ($w_1=0.75$, $K=3$)	50.50	49.27	44.20	38.44	33.25
MMSEC ($w_1=0.80$, $K=1$)	52.00	48.00	44.80	37.78	36.00
MMSEC ($w_1=0.80$, $K=2$)	51.00	48.00	44.60	38.22	35.00
MMSEC ($w_1=0.80$, $K=3$)	51.50	48.91	44.80	39.78	34.50
Number of the original training samples per class	8	9	10	11	12
RCR	48.57	47.67	44.80	42.00	40.67
CRC	46.00	47.33	47.60	43.50	42.00
MSEC	41.14	39.33	37.20	32.50	32.67
MMSEC ($w_1=0.75$, $K=1$)	32.29	30.67	29.20	27.50	26.67
MMSEC ($w_1=0.75$, $K=2$)	32.86	30.33	28.40	24.50	24.00
MMSEC ($w_1=0.75$, $K=3$)	32.57	29.67	28.40	26.00	28.00
MMSEC ($w_1=0.80$, $K=1$)	34.86	32.33	31.20	28.50	28.67
MMSEC ($w_1=0.80$, $K=2$)	35.14	31.00	29.20	26.50	26.00
MMSEC ($w_1=0.80$, $K=3$)	34.29	30.00	29.60	26.00	28.00

Table 2
Rate of classification errors (%) of the integration of PCA and different methods on the GT face database.

Number of the original training samples per class	3	4	5	6	7
RCR	59.17	54.91	54.80	44.22	43.50
CRC	57.83	56.55	56.00	52.00	47.75
MSEC	54.67	53.45	51.20	45.11	41.50
MMSEC ($w_1=0.75$, $K=1$)	51.17	47.64	45.60	36.44	35.25
MMSEC ($w_1=0.75$, $K=2$)	49.67	47.82	44.60	37.11	33.75
MMSEC ($w_1=0.75$, $K=3$)	50.17	48.91	44.60	38.44	33.25
MMSEC ($w_1=0.80$, $K=1$)	51.00	48.00	44.80	37.33	35.75
MMSEC ($w_1=0.80$, $K=2$)	50.00	47.82	44.20	38.00	34.75
MMSEC ($w_1=0.80$, $K=3$)	51.17	48.55	44.80	39.33	34.50
Number of the original training samples per class	8	9	10	11	12
RCR	38.86	38.33	30.80	31.00	26.67
CRC	46.29	45.67	46.80	43.50	43.33
MSEC	41.14	39.33	37.60	33.00	33.33
MMSEC ($w_1=0.75$, $K=1$)	32.29	30.33	29.20	27.50	26.67
MMSEC ($w_1=0.75$, $K=2$)	32.86	30.33	28.40	24.50	24.00
MMSEC ($w_1=0.75$, $K=3$)	32.57	30.00	28.40	26.00	28.67
MMSEC ($w_1=0.80$, $K=1$)	34.86	32.33	30.80	28.50	28.67
MMSEC ($w_1=0.80$, $K=2$)	35.14	31.00	29.20	26.50	26.00
MMSEC ($w_1=0.80$, $K=3$)	34.29	30.33	29.20	26.00	28.00

Table 3
Rate of classification errors (%) of different methods on the FERET database.

Number of the original training samples per class	3	4	5
RCR	53.37	57.50	29.75
CRC	55.63	54.67	31.50
MSEC	52.75	60.67	29.25
MMSEC ($w_1=0.75$, $K=1$)	44.62	53.83	22.25
MMSEC ($w_1=0.75$, $K=2$)	46.12	54.00	23.25
MMSEC ($w_1=0.75$, $K=3$)	45.62	53.83	23.00
MMSEC ($w_1=0.80$, $K=1$)	46.38	56.00	24.00
MMSEC ($w_1=0.80$, $K=2$)	47.38	55.50	25.50
MMSEC ($w_1=0.80$, $K=3$)	47.13	55.00	25.00

weeks. We take the first 18, 19, 20, 21 and 22 face images of each subject as test samples respectively, and take the remaining images as training samples. As a result, in these experiments 4, 5, 6, 7 and 8 face images of each subject are used as training samples respectively. Each cropped and used face image has a size of 50×40. Table 4 also shows that our proposed method outperforms MSEC, CRC and RCR.

Table 4
Rate of classification errors (%) of different methods on the AR database.

Number of the original training samples per class	4	5	6	7	8
RCR	32.69	29.44	29.08	27.63	27.92
CRC	32.46	30.40	29.17	29.74	30.05
MSEC	27.92	24.88	25.87	25.48	25.93
MMSEC ($w_1=0.75$, $K=1$)	27.42	23.69	23.67	22.94	23.52
MMSEC ($w_1=0.75$, $K=2$)	26.48	23.13	24.04	23.07	23.66
MMSEC ($w_1=0.75$, $K=3$)	26.63	22.94	23.96	23.07	23.01
MMSEC ($w_1=0.80$, $K=1$)	26.70	23.37	23.67	22.28	23.10
MMSEC ($w_1=0.80$, $K=2$)	25.72	22.74	23.88	22.15	23.19
MMSEC ($w_1=0.80$, $K=3$)	25.91	22.78	24.04	22.76	22.87

5.4. Experiments on the PIE face database

The CMU PIE face database contains 41,368 face images from 68 subjects. The original face image is cropped to 32×32 pixels gray image [56]. As shown in Ref. [57], for each subject, we adopt only the images with different lighting conditions and fixed pose and expression to conduct the experiment. The first 1, 2 and 3 face images of the adopted images of each subject in this subset are respectively used as training samples and the other face images serve as test samples. Table 5 shows again that MMSEC can obtain lower rate of classification errors than MSEC.

Table 5
Rate of classification errors (%) of different methods on the PIE database.

Number of the original training samples per class	1	2	3
MSEC	14.78	7.59	0.98
MMSEC ($w_1=0.75$, $K=1$)	13.53	8.05	0.65
MMSEC ($w_1=0.75$, $K=2$)	10.88	5.42	0.33
MMSEC ($w_1=0.75$, $K=3$)	10.81	3.79	0.16
MMSEC ($w_1=0.80$, $K=1$)	13.31	6.50	0.65
MMSEC ($w_1=0.80$, $K=2$)	10.66	4.88	0.33
MMSEC ($w_1=0.80$, $K=3$)	10.66	3.64	0.16

5.5. Experiments on noised face images

In this subsection, the GT face database is used and the sets of training samples and test samples are the same as those in Section 5.1. In order to simulate the complex scenario where the test samples are very different from the training samples from the same subject, we use Matlab function "imnoise" to add Gaussian white noise of zero mean and variance of 0.005 to the test samples and make the training samples be the same as the original ones. Fig. 7 shows some of the noised face images. The experimental results shown in Table 6 indicate that our proposed method obtains a much lower rate of classification errors than MSEC, CRC and RCR. For example, when the first eight face images of each subject and the remaining images are respectively used as the training samples and test samples, MMSEC with $w_1=0.75$ and $K=1$ obtains a rate of classification errors of 36.86%. However, the rates of classification errors of RCR, CRC and MSEC are 74.86%, 57.14% and 47.71%, respectively. It is clear that in this case the accuracy of MMSEC is 10% higher than that of MSEC. The main reason why MMSEC can perform much better for noised samples is as follows. MMSEC is trained by training samples and the predicted class label of the nearest training sample might be very

Fig. 7. Some of the noised face images.

Table 6
Rate of classification errors (%) of different methods on the noised test samples from the GT face database.

Number of the original training samples per class	3	4	5	6	7
RCR	75.83	76.00	76.20	75.33	74.50
CRC	63.50	61.45	64.20	62.67	58.25
MSEC	59.67	58.91	57.60	54.00	50.25
MMSEC ($w_1=0.75$, $K=1$)	52.83	49.09	47.60	41.56	40.25
MMSEC ($w_1=0.75$, $K=2$)	52.83	49.45	46.20	40.22	38.00
MMSEC ($w_1=0.75$, $K=3$)	52.33	51.27	47.20	41.33	38.25
MMSEC ($w_1=0.80$, $K=1$)	52.33	51.45	48.40	43.78	43.00
MMSEC ($w_1=0.80$, $K=2$)	53.50	52.18	47.40	42.44	42.00
MMSEC ($w_1=0.80$, $K=3$)	53.33	52.36	48.20	42.89	41.75
Number of the original training samples per class	8	9	10	11	12
RCR	74.86	71.00	70.00	66.50	63.33
CRC	57.14	57.67	58.80	56.00	55.33
MSEC	47.71	45.67	45.60	39.00	40.00
MMSEC ($w_1=0.75$, $K=1$)	36.86	35.33	34.80	32.00	30.00
MMSEC ($w_1=0.75$, $K=2$)	35.43	34.33	34.00	30.50	26.67
MMSEC ($w_1=0.75$, $K=3$)	35.43	33.33	32.00	28.50	28.67
MMSEC ($w_1=0.80$, $K=1$)	38.57	37.33	36.00	32.50	33.33
MMSEC ($w_1=0.80$, $K=2$)	37.14	36.33	36.00	31.00	32.67
MMSEC ($w_1=0.80$, $K=3$)	36.29	36.67	34.40	29.50	33.33

close to the true class label of the test sample, so the use of the nearest training sample enables the MMSEC method to be more robust to the noisy test data.

6. Conclusions

The modified minimum squared error (MMSEC) algorithm proposed in this paper simultaneously exploits the predicted class labels of the test sample and the training samples nearest to it to perform classification. As the training samples nearest to the test sample can provide useful information for classifying it, MMSEC is able to obtain higher classification accuracy than MSEC. MMSEC partially owns the advantages of conventional MSE and nearest neighbor classifier. Various experiments show that MMSEC does obtain higher classification accuracy than conventional MSEC, CRC and RCR.

Acknowledgments

This article is partly supported by NSFC under Grants nos. 61370163, 61233011, 61203376 and 61332011 and 61300032, as well as the Fundamental Research Funds for the Central Universities (HIT.NSRIF. 2009130), Shenzhen Municipal Science and Technology Innovation Council (Nos. JC201005260122A, JCYJ20120613153352732 and CXZZ20120613141657279).

References

[1] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, (2nd ed.).
[2] J. Xu, X. Zhang, Y. Li, Kernel MSEC algorithm: a unified framework for KFD, LS-SVM and KRR, in: Proceedings of the International Joint Conference on Neural Networks, 2001, pp. 1486–1491.
[3] D. Tao, X. Li, X. Wu, S.J. Maybank., General tensor discriminant analysis and Gabor features for gait recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (10) (2007) 1700–1715.
[4] J. Ye, Least squares linear discriminant analysis, in: Proceedings of the International Conference on Machine Learning, 2007, pp. 1087–1094.
[5] T. Hastie, A. Buja, R. Tibshirani, Penalized discriminant analysis, Ann. Stat. 23 (1995) 73–102.
[6] L. Sun, S. Ji, J. Ye, Canonical correlation analysis for multilabel classification: a least-squares formulation, extensions, and analysis, IEEE Trans. Pattern Anal. Mach. Intell. 33 (1) (2011) 194–200.
[7] Y. Xu, D. Zhang, Z. Jin, M. Li, J.-Y. Yang, A fast kernel-based nonlinear discriminant analysis for multi-class problems, Pattern Recognit. 39 (6) (2006) 1026–1033.
[8] S.A. Billings, K.L. Lee, Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm, Neural Netw. 15 (1) (2002) 263–270.
[9] A. Belloni, V. Chernozhukov, ℓ1-Penalized Quantile Regression in High-Dimensional Sparse Models, forthcoming Annals of Statistics, 2010b.
[10] P.J. Bickel, Y. Ritov, A.B. Tsybakov, Simultaneous analysis of Lasso and Dantzig selector, Ann. Stat. 37 (2009) 1705–1732.
[11] T. Hastie, M.Y. Park, ℓ1-Regularization path algorithm for generalized linear models, J. R. Stat. Soc. B 69 (2007) 659–677.
[12] L. Zhang, et al., Sparse representation or collaborative representation: which helps face recognition? in: Proceedings of the ICCV, 2011, pp. 471–478.
[13] Q. Shi, A. Eriksson, A. Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: Proceedings of the CVPR, 2011, pp. 553–560.
[14] Y. Xu, D. Zhang, J. Yang, J.-Y. Yang, A two-phase test sample sparse representation method for use with face recognition, IEEE Trans. Circuits Syst. Video Technol. 21 (9) (2011) 1255–1262.
[15] I. Naseem, R. Togneri, M. Bennamoun, Linear regression for face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (11) (2010) 2106–2112.
[16] V. Baramidze, M.-J. Lai, Convergence of discrete and penalized least squares spherical splines, J. Approx. Theory 163 (9) (2011) 1091–1106.
[17] I. Markovsky, S.V. Huffel, Overview of total least-squares methods, Signal Process. 87 (10) (2007) 2283–2302.
[18] Y.C. Eldar, Universal weighted MSE improvement of the least-squares estimator, IEEE Trans. Signal Process. 56 (5) (2008) 1788–1800.
[19] G.M. Schuster, A.K. Katsaggelos, Robust line detection using a weighted MSE estimator, in: Proceedings of International Conference on Image Processing 1 (2003) 293–296.
[20] G.M. Schuster, A.K. Katsaggelos, Robust line detection using a weighted MSE estimator, in: Proceedings of the International Conference on Image Processing, vol. 1, 2003, pp. 293–296.
[21] X. Xu, H. He, D. Hu, Efficient reinforcement learning using recursive least-squares methods, J. Artif. Intell. Res. 16 (2002) 259–292.
[22] G. Yao, R. Ding, Two-stage least squares based iterative identification algorithm for controlled autoregressive moving average (CARMA) systems, Comput. Math. Appl. 63 (5) (2012) 975–984.
[23] L.A. Dalton, E.R. Dougherty, Exact sample conditioned MSE performance of the Bayesian MMSE estimator for classification error – Part I: representation, IEEE Trans. Signal Process. 60 (5) (2012) 2575–2587.
[24] L.A. Dalton, E.R. Dougherty, Optimal mean-square-error calibration of classifier error estimators under Bayesian models, Pattern Recognit. 45 (6) (2012) 2308–2320.
[25] L.A. Dalton, E.R. Dougherty, Bayesian minimum mean-square error estimation for classification error – Part I: definition and the Bayesian MMSE error estimator for discrete classification, IEEE Trans. Signal Process. 59 (1) (2011) 115–129.
[26] H. Kim, B.L. Drake, H. Park, Adaptive nonlinear discriminant analysis by regularized minimum squared errors, IEEE Trans. Knowl. Data Eng. 18 (5) (2006) 603–612.
[27] W. Liu, D. Tao, Multiview Hessian regularization for image annotation, IEEE Trans. Image Process. 22 (7) (2013) 2676–2687.
[28] S. Chen, X. Hong, B.L. Luk, C.J. Harris, Orthogonal-least-squares regression: a unified approach for data modelling, Neurocomputing 72 (10–12) (2009) 2670–2681.
[29] Y. Xu, D. Zhang, Z. Jin, M. Li, J.-Y. Yang, A fast kernel-based nonlinear discriminant analysis for multi-class problems, Pattern Recognit. 39 (6) (2006) 1026–1033.
[30] Q. Zhu, Reformative nonlinear feature extraction using kernel MSE, Neurocomputing 73 (16–18) (2010) 3334–3337.

[31] Y.-P. Zhao, Z.-H. Du, Z.-A. Zhang, H.-B. Zhang, A fast method of feature extraction for kernel MSE, Neurocomputing 74 (10) (2011) 1654–1663.
[32] J. Wang, J. You, et al., Extract minimum positive and maximum negative features for imbalanced binary classification, Pattern Recognit. 45 (2012) 1136–1145.
[33] A.M. Bagirov, Modified global k-means algorithm for minimum sum-of-squares clustering problems, Pattern Recognit. 41 (10) (2008) 3192–3199.
[34] S.V. Huffel, I. Markovsky, R.J. Vaccaro, T. Söderström, Total least squares and errors-in-variables modeling, Signal Process. 87 (10) (2007) 2281–2282.
[35] G.-X. Yu, Z. Yu, J. Hua, X. Li, J. You, Sparse representation based spectral regression, ICMLC (2011) 532–537.
[36] S. Chang, L. Carin, A modified SPIHT algorithm for image coding with a joint MSE and classification distortion measure, IEEE Trans. Image Process. 15 (3) (2006) 713–725.
[37] R.G. McKilliam, B.G. Quinn, I.V.L. Clarkson, Direction estimation by minimum squared arc length, IEEE Trans. Signal Process. 60 (5) (2012) 2115–2124.
[38] T.E. Bogale, L. Vandendorpe, Robust sum MSE optimization for downlink multiuser MIMO systems with arbitrary power constraint: generalized duality approach, IEEE Trans. Signal Process. 60 (4) (2012) 1862–1875.
[39] X. Jiang, W.-J. Zeng, E. Cheng, C.-R. Lin, Multipath channel estimation using fast least-squares algorithm, in: Proceedings of 2011 Third International Conference on Communications and Mobile Computing (CMC) (2011) 433–436.
[40] J.-H. Zhao, P.L.H. Yu, J.T. Kwok, Bilinear probabilistic principal component analysis, IEEE Trans. Neural Netw. Learn. Syst. 23 (3) (2012) 492–503.
[41] X. Wang, D. Tao, Z. Li, Entropy controlled Laplacian regularization for least square regression, Signal Process. 90 (6) (2010) 2043–2049.
[42] Y. Su, X. Gao, X. Li, D. Tao, Multivariate multilinear regression, IEEE Trans. Syst. Man Cybern. Part B 42 (6) (2012) 1560–1573.
[43] J. Wang, J. You, et al., Orthogonal discriminant vector for face recognition across pose, Pattern Recognit. 45 (2012) 4069–4079.
[44] W. Yang, C. Sun, L. Zhang, A multi-manifold discriminant analysis method for image feature extraction, Pattern Recognit. 44 (8) (2011) 1649–1657.
[45] W. Yang, X. Yan, L. Zhang, C. Sun, Feature extraction based on fuzzy 2DLDA, Neurocomputing 73 (10–12) (2010) 1556–1561.
[46] M. Yang, L. Zhang, D. Zhang, S. Wang, Relaxed collaborative representation for pattern classification, in: Proceedings of the CVPR (2012) 2224–2231.
[47] ⟨http://www.anefian.com/research/face_reco.htm⟩.
[48] T. Zhou, D. Tao, Double shrinking sparse dimension reduction, IEEE Trans. Image Process. 22 (1) (2013) 244–257.
[49] B. Geng, D. Tao, C. Xu, L. Yang, X.-S. Hua, Ensemble manifold regularization, IEEE Trans. Pattern Anal. Mach. Intell. 34 (6) (2012) 1227–1233.
[50] D. Tao, X. Li, X. Wu, W. Hu, S.J. Maybank, Supervised tensor learning. In: Proceedings of the Fifth IEEE International Conference on Data Mining, 13 (1) , 2007, pp. 1–42.
[51] E.J. Candès, X. Li, Y. Ma, J. Wright, Robust principal component analysis? JACM 58 (3) (2011) 11.
[52] N. Guan, D. Tao, Z. Luo, B. Yuan., NeNMF: an optimal gradient method for nonnegative matrix factorization, IEEE Trans. Signal Process. 60 (6) (2012) 2882–2898.
[53] N. Guan, D. Tao, Z. Luo, B. Yuan, Non-negative patch alignment framework, IEEE Trans. Neural Netw. 22 (8) (2011) 1218–1230.
[54] ⟨http://www.itl.nist.gov/iad/humanid/feret/feret_master.html⟩.
[55] ⟨http://www2.ece.ohio-state.edu/~aleix/ARdatabase.html⟩.
[56] ⟨http://www.cad.zju.edu.cn/home/dengcai/Data/FaceData.html⟩.
[57] X. He, D. Cai, P. Niyogi, Laplacian score for feature selection. NIPS, (2005).

Yong Xu was born in Sichuan, China, in 1972. He received his B.S. degree, M.S. degree in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern Recognition and Intelligence system at NUST (China) in 2005. Now he works at Shenzhen Graduate School, Harbin Institute of Technology. His current interests include pattern recognition, biometric, face recognition, machine learning and image processing.

Xiaozhao Fang received his MS degree in computer science from Guangdong University of Technology, Guangzhou, China, in 2008. He is currently pursuing his Ph.D. degree in computer science and technology at Shenzhen Graduate School, HIT, Shenzhen, China. He has published more than seven journal papers. His current research interests include pattern recognition and machine learning.

Qi Zhu received the M.S. degree in computer science from Harbin Institute of Technology in 2009. He is currently pursuing the Ph.D. degree in computer science and technology at Harbin Institute of Technology. He has published more than 10 journal papers. His current research interests include face recognition and feature extraction.

Yan Chen received her B.E. and M.E. degree in computer science from Northeastern University, China in 1997 and 2000 respectively, and her Ph.D. in 2010 from university of Technology, Sydney(UTS), Australian. Currently, she is a Researcher with Harbin Institute of Technology (HIT) at Shenzhen, China. She is also a R&D member of Shenzhen Sunwin Intelligent Co. Ltd. Her research interests include computer vision and pattern recognition.

Jane You obtained her B.Eng. in Electronic Engineering from Xi'an Jiaotong University in 1986 and Ph.D. in Computer Science from La Trobe University, Australia in 1992. She was a lecturer at the University of South Australia and senior lecturer at Griffith University from 1993 till 2002. Currently she is a professor at the Hong Kong Polytechnic University. Her research interests include image processing, pattern recognition, medical imaging, biometrics computing, multimedia systems and data mining.

Hong Liu received his Ph.D. in Mechanical Electronics and Automation in 1996, and serves as a full professor in School of EE&CS, Peking University. He is also the director of Engineering Lab on Intelligent Perception for Internet of Things. His research fields include computer vision and robotics, image processing and pattern recognition. Prof. Liu has published more than 100 papers and gained Chinese National Aero-space Award, Excellence Teaching Award, and Candidates of Top Ten Outstanding Professors, Peking University. He is an IEEE member, vice chair of Intelligent Robotics Society of Chinese Association for Artificial Intelligent (CAAI), and also the President of National Youth Committee of CAAI. He has served as co-chairs, session chairs or PC members of many important international conferences, such as IEEE/RSJ IROS, IEEE ROBIO and IEEE SMC, and also serves as reviewers for many international journals such as Pattern Recognition, IEEE Trans. on Signal Processing and IEEE Trans. on PAMI.