GA-Driven LDA in KPCA Space for Facial Expression Recognition

Qijun Zhao and Hongtao Lu

Department of Computer Science and Engineering,
Shanghai Jiao Tong University, Shanghai 200030, P.R. China
qijunzhao@hotmail.com
lu-ht@cs.sjtu.edu.cn

Abstract. Automatic facial expression recognition has been studied comprehensively recently, but most existent algorithms for this task perform not well in presence of nonlinear information in facial images. For this sake, we employ KPCA to map the original facial data to a lower dimensional space. Then LDA is applied in that space and we derive the most discriminant vectors using GA. This method has no singularity problem, which often arises in the traditional eigen decomposition-based solutions to LDA. Other work of this paper includes proposing a rather simple but effective preprocessing method and using Mahalanobis distance rather than Euclidean distance as the metric of the nearest neighbor classifier. Experiments on the JAFFE database show promising results.

1 Introduction

Automatic facial expression recognition has many potential applications such as more intelligent human-computer interface and human emotion interpretation. Along with hot researches in this field, many methods have been proposed. Samel and Iyengar gave an overview of the early work on this topic [1]. More recent work can be found in the surveys of Pantic and Rothkrantz [2], Fasel and Luettin [3].

Six basic emotions, i.e. happiness, sadness, surprise, fear, anger and disgust, are agreed on by most scholars in this realm. In order to recognize these expressions, Elkman et al. [4] proposed the facial action coding system (FACS), which uses 44 action units (AUs) to describe facial actions with regard to their locations as well as their intensities. Instead of dividing the whole face into different units, Lyons et al. [5] chose 34 feature points (e.g., corners of the eye and mouth) on a face for the purpose of analyzing its expression. Another method is to base analysis and recognition directly on the whole facial image rather than its segmentations or so-called feature points [6]. Such method is typically fast and simple. For this sake, we take the last method in this paper. But before further processing, we first make all face images have the same inter-ocular distance and an average face shape so that there is more correspondence among the features on them.

L. Wang, K. Chen, and Y.S. Ong (Eds.): ICNC 2005, LNCS 3611, pp. 28–36, 2005.

As a classic statistical method, linear discriminant analysis (LDA) is widely used as a way of both feature extraction and dimensionality reduction in the recognition of facial expressions [5][7]. The basic idea of LDA is to maintain the cluster property of the data after their projection to the discriminant space. Traditionally, the LDA model is solved as a problem of generalized eigenvalue problem [8]. But because the dimensionality of a typical image (i.e. the number of pixels in it) is usually much larger than the number of available samples, the scatter matrices might be singular and the LDA works poor. This is called the singularity problem.

Principal component analysis (PCA) is often used as a preprocessing step to reduce the dimensionality of the original data. LDA is then applied in the lower dimensional space. This relieves or eliminates the singularity problem[9]. However, PCA only uses the second order statistical information in data. As a result, it fails to perform well in nonlinear cases. Recently, with the idea of kernel methods, Scholkopf [10] extended PCA to the nonlinear case, called Kernel Principle Component Analysis (KPCA). Its application to face recognition [11] shows excellent performance. This is due to the nonlinear essence of kernel methods and the substantive nonlinear information in facial images. In this paper, we employ KPCA as the first step for dimensionality reduction.

LDA is essentially an optimization problem. Therefore Genetic Algorithm (GA), an effective optimization algorithm, can be incorporated into LDA. In fact, GA has already been applied to the recognition of faces [11][12]. In this paper, based on GA, we propose a new solution for LDA. The proposed algorithm avoids the singularity problem and the experiments on the JAFFE database prove its effectiveness.

The remainder of this paper is organized as follows. KPCA is reviewed in section 2. After a simple introduction of LDA, section 3 specifically presents the GA-driven LDA algorithm. To complete the facial expression recognition system, we give the preprocessing method and classifier in section 4. Then section 5 shows the results of our experiments on the JAFFE database. Finally, we concludes this paper in section 6, where further research directions are also presented.

2 Kernel Principal Component Analysis

When mapped into a higher dimensional space, a non-linearly separable problem may become linearly separable. This underlies the basic idea of KPCA as well as other kernel methods. Denote such mapping as

$$\Phi : R^n \mapsto F,\ x \mapsto \Phi(x), \tag{1}$$

where F is a higher dimensional space and it could be infinite. Assume $\Phi(x_i), i = 1, 2, \cdots, M$, are centered, i.e. $\sum_{i=1}^{M} \Phi(x_i) = 0$. Then the total scatter matrix of these samples in F is $S_t^{\Phi} = \frac{1}{M} \sum_{i=1}^{M} \Phi(x_i)\Phi(x_i)^T$, where T denotes the transpose operation. PCA is applied in F, i.e. to solve the following eigenvalue equation:

$$S_t^{\Phi} w^{\Phi} = \lambda^{\Phi} w^{\Phi}, \tag{2}$$

where w^{Φ} is a column of the optimal projection matrix W^{Φ} in F. Substituting S_t^{Φ} into (2), we get

$$w^{\Phi} = \frac{1}{\lambda^{\Phi}} \frac{1}{M} \sum_{i=1}^{M} (\Phi(x_i) \cdot w^{\Phi}) \Phi(x_i), \tag{3}$$

where '$\cdot$' denotes the inner product. From (3) we can see that all solutions w^{Φ} with $\lambda^{\Phi} \neq 0$ lie in the span of $\Phi(x_1), \Phi(x_2), \cdots, \Phi(x_M)$. Thus, (2) is equivalent to the set of equations:

$$(\Phi(x_i) \cdot S_t^{\Phi} w^{\Phi}) = \lambda^{\Phi} (\Phi(x_i) \cdot w^{\Phi}), \qquad for\ all\ i = 1, 2, \cdots, M, \tag{4}$$

and there exist coefficients α_i $(i = 1, 2, \cdots, M)$ such that

$$w^{\Phi} = \sum_{i=1}^{M} \alpha_i \Phi(x_i). \tag{5}$$

According to (4) and (5), we have

$$\lambda^{\Phi} \sum_{i=1}^{M} \alpha_i (\Phi(x_k) \cdot \Phi(x_i)) = \frac{1}{M} \sum_{i=1}^{M} \alpha_i (\Phi(x_k) \cdot \sum_{j=1}^{M} \Phi(x_j)(\Phi(x_j) \cdot \Phi(x_i))), \tag{6}$$

for all $k = 1, 2, \cdots, M$. Defining an $M \times M$ matrix K with its element in the $i_{\mathtt{th}}$ row and $j_{\mathtt{th}}$ column as

$$K_{ij} := (\Phi(x_i) \cdot \Phi(x_j)), \tag{7}$$

this reads

$$M\lambda^{\Phi} K\alpha = K^2\alpha, \tag{8}$$

where α denotes the column vector with entries $\alpha_1, \alpha_2, \cdots, \alpha_M$. As shown in [10], all its solutions are given by the eigenvectors of K:

$$M\lambda^{\Phi}\alpha = K\alpha. \tag{9}$$

With these α's and (5), we get the optimal projection matrix for the samples in F. And for a test sample x, we can calculate its $k_{\mathtt{th}}$ principal component in F by

$$(w_k^{\Phi} \cdot \Phi(x)) = \sum_{i=1}^{M} \alpha_i^k (\Phi(x_i) \cdot \Phi(x)), \tag{10}$$

where w_k^{Φ} is the $k_{\mathtt{th}}$ column of W^{Φ} and α^k is the eigenvector of K corresponding to its $k_{\mathtt{th}}$ largest eigenvalue.

Generally, m eigenvectors corresponding to the m largest eigenvalues are chosen to span the KPCA space and in order to normalize them, we take the whitening procedure as following:

$$\alpha^i = \frac{\alpha^i}{\sqrt{\lambda_i^{\Phi}}}, \qquad i = 1, 2, \cdots, m. \tag{11}$$

According to the Mercer condition, the dot product in these equations can be replaced by

$$k(x_i, x_j) = (\Phi(x_i) \cdot \Phi(x_j)), \tag{12}$$

where *k(*,*)* is called a kernel function. The polynomial function

$$k(x, y) = (x \cdot y + 1)^d, \tag{13}$$

where d is any positive integer, and the radial basis function or Gaussian kernel function

$$k(x, y) = \exp(-\frac{\|x - y\|^2}{\sigma^2}), \tag{14}$$

where $\sigma > 0$, are among the commonly used kernel functions [10].

3 GA-Driven LDA

Suppose there are M samples, $x_1, x_2, \cdots, x_M \in R^n$, which are categorized into L classes with N_j samples of the class $j (j = 1, 2, \cdots, L)$. Let I_j denote the set of indexes of the samples belonging to the class j, c denote the mean of all these samples, and c_j denote the mean of the class j. The within-class scatter matrix $S_w = \frac{1}{L}\sum_{j=1}^{L}\frac{1}{N_j}\sum_{i\in I_j}(x_i - c_j)(x_i - c_j)^T$ and the between-class scatter matrix $S_b = \frac{1}{L}\sum_{j=1}^{L} N_j(c_j - c)(c_j - c)^T$. The quotient of the determinants of the between-class and within-class scatter matrices of the projected samples is a common criterion of LDA and the optimal projection matrix of LDA is to maximize it:

$$W_{LDA} = \arg\max_W \frac{|W^T S_b W|}{|W^T S_w W|}. \tag{15}$$

To apply GA, the optimal discriminant vectors, i.e. the column vectors in W_{LDA}, are viewed as the rotation of the basis of the KPCA space [12]. We start the evolution with the identity basis of the KPCA space and rotate them two by two with randomly selected angles. A random number of vectors are then chosen as discriminant vectors from the rotated vectors.

Definition of Chromosome. Referring to Liu and Wechsler's work [12], binary chromosome, i.e. a bit string, is defined for GA-driven LDA. As we have discussed above, the solutions (discriminant vectors) are derived from the pairwise rotation of the identity basis of the KPCA space: $\epsilon_1, \epsilon_2, \cdots, \epsilon_m$, where $\epsilon_i \in R^m$ with '*1*' as its $i_{\mathtt{th}}$ element and '*0*' the others. Because two of the basis are rotated together once, the total number of rotation angles for m basis vectors is $\frac{m(m-1)}{2}$. Thus the chromosome should depict the following rotation angles: $a_1, a_2, \cdots, a_{\frac{m(m-1)}{2}}$. Each angle a_k $(k = 1, 2, \cdots, \frac{m(m-1)}{2})$ is represented by 10 bits, thus the rotation interval is $\frac{\pi}{2^{11}}$ (the rotation angles are confined to $[0, \frac{\pi}{2}]$ for simplicity). And a_k corresponds to the $i_{\mathtt{th}}$ and $j_{\mathtt{th}}$ vectors (suppose $i < j$), where $k = \sum_{t=1}^{i-1}(m - t) + j - i$. The corresponding rotation matrix Q_k is an $m \times m$ identity matrix but $Q_k(i, i) = \cos(a_k)$, $Q_k(i, j) = -\sin(a_k)$, $Q_k(j, i) = \sin(a_k)$ and $Q_k(j, j) = \cos(a_k)$. And the total rotation matrix is $Q = Q_1 Q_2 \cdots Q_{\frac{m(m-1)}{2}}$. Let $\xi_1, \xi_2, \cdots, \xi_m \in R^m$ be the result vectors after rotation:

$$[\xi_1 \xi_2 \cdots \xi_m] = [\epsilon_1 \epsilon_2 \cdots \epsilon_m] \times Q. \tag{16}$$

Not all these vectors are chosen as discriminant vectors. Instead, we randomly select l vectors from them. This leads to another m bits in the chromosome. These bits, $b_1, b_2, \cdots, b_m$, demonstrate which vectors are selected: if b_i=1, then ξ_i is chosen; otherwise, discarded.

Genetic Operators. Two genetic operators, crossover and mutation, are employed. Both of them are conducted with a given probability. If crossover is taken on two individuals, a random position is selected. Then the bits before this position in one individual and those after this position in the other individual are combined to form a new individual. So are the rest of them. As for mutation, if one bit of an individual is supposed to be mutated, then it is converted from '*0*' to '*1*' or from '*1*' to '*0*'; otherwise, keep it unchanged.

Fitness. The fitness in GA-driven LDA is based on the criterion of LDA. Take an individual $D = (a_1 a_2 \cdots a_{\frac{m(m-1)}{2}}; b_1 b_2 \cdots b_m)$ as an example. Assume the vectors after rotation are $\xi_1, \xi_2, \cdots, \xi_m$ and l vectors $\eta_1, \eta_2, \cdots, \eta_l$ are selected from them as the discriminant vectors. Then the sample $x_{KPCA} \in R^m$ in the KPCA space will be mapped to $x_{GA-LDA} \in R^l$ in the GA-LDA space:

$$x_{GA-LDA} = [\eta_1 \eta_2 \cdots \eta_l]^T \cdot x_{KPCA}, \tag{17}$$

where x_{KPCA} can be obtained from the original sample $x \in R^n$ according to (10).

Project all samples into the GA-LDA space by (10) and (17) and calculate the within-class scatter matrix S_w^G and the between-class scatter matrix S_b^G for the projected samples in the GA-LDA space. Then the fitness of the individual D is defined as

$$\zeta(D) = tr(S_b^G)/tr(S_w^G). \tag{18}$$

Once the fitness of all individuals in the population has been worked out, choose those individuals with larger fitness and form a new generation. Unless the stopping criterion is met, for example, the maximum number of trials is reached, GA is run on the new generation again. When GA stops, the individual with the largest fitness in the last generation gives the result, i.e. l optimal discriminant vectors: $W_{GA-LDA} = [\eta_1^* \eta_2^* \cdots \eta_l^*]$.

4 Facial Expression Recognition

4.1 Preprocessing Facial Images

Firstly, we manually mark the centers of eyes on each face with two points and the face region with a rectangle. Let's denote the centers of left and right eyes as *El* and *Er*. And the left-most, the right-most, the upper-most and the bottom-most points of the face are denoted by L, R, T and B respectively. Secondly, calculate the average Y coordinate $\bar{E}_y$ of eyes, the average X coordinates $\bar{El}_x$ and $\bar{Er}_x$ of the left and right eyes, and the average Y coordinates $\bar{T}_y$ and $\bar{B}_y$ of the top-most and bottom-most points. Hereafter, the subscripts 'x' and 'y' represent the X and Y coordinates. Thirdly, calibrate the eyes to make them

have the same average inter-ocular distance $\bar{d} = \bar{Er_x} - \bar{El_x}$. This is implemented through scaling in three intervals along the horizontal direction on the face: $[L_x, El_x]$, $[El_x, Er_x]$ and $[Er_x, R_x]$. These intervals are scaled, respectively, to the average ones, $[0, (w - \bar{d})/2]$, $[(w - \bar{d})/2, \bar{d} + (w - \bar{d})/2]$ and $[\bar{d} + (w - \bar{d})/2, w]$, where $w = R_x - L_x$ is the width of the face. Fourthly, calibration along the vertical direction is conducted on each facial image, i.e., vertically divide the face into two intervals, $[T_y, E_y]$ and $[E_y, B_y]$, and scale them to two average intervals, $[0, h * (\bar{E_y} - \bar{T_y})/(\bar{B_y} - \bar{E_y})]$ and $[h * (\bar{E_y} - \bar{T_y})/(\bar{B_y} - \bar{E_y}), h]$, where $h = B_y - T_y$ is the height of the face. Lastly, the face is scaled to the standard width W and height H.

After the calibration and standardization, we work out the average images of each expression and average these average images to get another average image, which we call the BLANK one. Then it is subtracted from all facial images so that we get the difference images as the final input data of the facial expression recognition system. Here the philosophy lies in the belief that information really helpful for the recognition task is included in the difference, thus with these difference images we can perform the recognition task well and the redundant information is reduced. Fig. 1 shows this procedure.

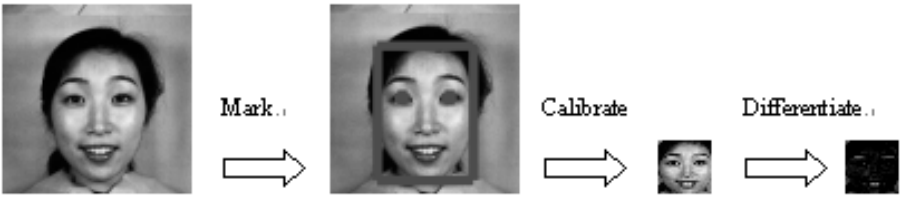

Fig. 1. Preprocessing facial images: *H=24* and *W=24*

4.2 Facial Expression Classifier

Generally, the nearest neighbor classifier [8] is taken when the features of facial images are extracted. And the common metric is Euclidean distance. However, in the view of probability and statistics, the Euclidean distance does not make use of the deviation or covariance of samples. In this paper, we take the Mahalanobis distance as the metric of the nearest neighbor classifier. Take a sample X from a cluster, whose mean is μ and covariance matrix is Σ, as an example, the distance between X and the cluster center μ is defined as

$$d_m^2 = (X - \mu)^T \times \Sigma^{-1} \times (X - \mu), \tag{19}$$

where the superscript '-1' denotes the inverse matrix.

5 Experimental Results

We test the proposed algorithm with the Japanese Female Facial Expression (JAFFE) database. The database contains 213 images of 7 facial expressions

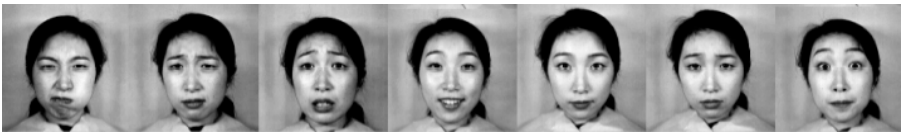

Fig. 2. Samples from the JAFFE database

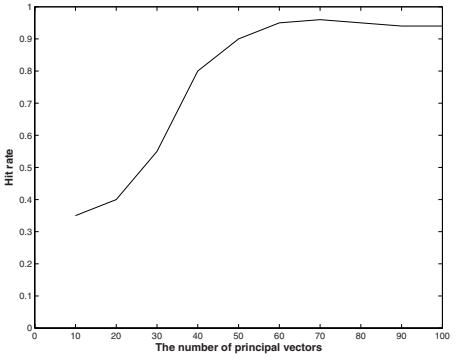

Fig. 3. The hit rate in relation to the number of principal vectors

(angry, depressed, fear, happy, normal, sad, surprised) posed by 10 Japanese female models. Fig. 2 shows some samples of one model.

The experiments are conducted in two ways: leave-one-image-out and leave-one-subject-out. In the 'leave-one-image-out' test, one image is randomly selected out from each expression category. As a result, the whole image set is divided into two sets. One has seven images as the testing set and the other consists of the rest ones as the training set. The whole training and testing process is conducted 10 times and every time a new testing set is selected. Finally we average the results to get the hit rate, or recognition rate, of the proposed scheme. In the 'leave-one-subject-out' test, all images of one subject are selected out to form the testing set with the rest ones as the training set. We also conduct the process 10 times and take the average result as the final hit rate over identity.

We first take the Gaussian kernel function with its $\sigma = 10$ to perform the 'leave-one-subject-out' test. In Fig. 3 we give the hit rate over identity in relation to the number of principal vectors (PVs). From the diagram we can see the best hit rate is 96%, which is apparently better than that of Lyons, 92% [5]. We also perform the tests using the polynomial kernel function. Table 1 shows the average hit rates in the 'leave-one-image-out' and the 'leave-one-subject-out' tests. And the result of 'leave-one-image-out' test, 80%, is also better than Lyons' result, 75% [5].

6 Conclusions and Further Considerations

In this paper, an effective algorithm is proposed for facial expression recognition. The proposed facial image preprocessing procedure improves the performance of

Table 1. Average Hit Rates

Kernel Function	Leave-one-image-out	Leave-one-subject-out
Polynomial, d=1	72%	85%
Polynomial, d=2	75%	90%
Gaussian, σ=10	80%	96%
Gaussian, σ=1000	78%	92%

the following KPCA and LDA. This algorithm avoids the singularity problem of LDA through applying KPCA before LDA and acquiring the optimal discriminant vectors by GA iterations rather than by solving the generalized eigenvalue problem. Moreover, the nearest neighbor classifier using Mahalanobis distance also performs well in recognizing facial expressions. Experiments on JAFFE database testify the effectiveness of this GA-driven LDA scheme in KPCA space for the task of facial expression recognition. A possible explanation to the excellence of this proposed algorithm is that it deals well with the nonlinear properties in facial images by using KPCA and taking into consideration the variances of different facial expressions.

However, the proposed preprocessing method is conducted by hand. This makes it laborious. What we are considering now is how to accomplish it automatically. Actually, both the segmentation of faces and the location of facial features, eyes and mouths for instance, are other hot but difficult topics in the literature of face and facial expression recognition. The complexion-based method [13] works well for segmenting faces in color images. Its basic idea is to distinguish human faces from other objects in images by their different optical properties, for example colors, and shape information. This method as well as other methods used in face detection are in our consideration. As for marking eyes in human faces, we intend to make use of such characteristics of eyes as the apparent difference in gray level between the area of eyes and its surroundings. Our primary goal is either to develop new algorithms or to enhance the accuracy of existent automatical face segmentation and face feature location methods.

Apart from the automatical marking of faces and eyes, we are also interested in the effect of the number of retained principal components as well as discriminant vectors on the final recognition rate. How to choose a proper number of such components and vectors is another direction for further researches.

References

1. A. Samel, P. A. Iyengar: Automatic recognition and analysis of human faces and facial expression: a survey. Pattern Recognition, Vol. 25, No. 1 (1992) 65-77
2. Maja Pantic, Leon J. M. Rothkrantz: Automatic analysis of facial expression: the state of the art. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 12 (2000) 1424-1445
3. B. Fasel, Juergen Luettin: Automatic facial expression analysis: a survey. Pattern Recognition, Vol. 36, No. 1 (2003) 259-275

4. P. Elkman, W. V. Friesen: Facial action coding system (FACS), Manual. Palo Alto: Consulting Psychologists Press (1978)
5. M. J. Lyons, S. Akamatsu, M. Kamachi, J. Gyoba: Coding facial expressions with Gabor wavelets. Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition, IEEE Computer Society (1998) 200-205
6. C. Padgett, G. Cottrell: Identifying emotion in static face images. Paper presented at the proceedings of the 2nd Joint Symposium on Neural Computation, San Diego, CA: University of California (1995)
7. Xuewen Chen, Thomas Huang, Facial expression recognition: a clustering-based approach. Pattern Recognition Letters, Vol. 24, No. 9-10 (2003) 1295-1302
8. Keinosuke Fukunaga: Introduction to statistical pattern recognition (second edition). Academic Press, Inc. (1990)
9. A. J. Calder, A. M. Burton, Paul Miller, Andrew W. Young, Shigeru Akamatsu: A principal component analysis of facial expressions. Vision Research, Vol. 41, No. 9 (2001) 1179-1208
10. B. Scholkopf, Alexander Smola, KlausRobert Muller: Nonlinear component analysis as a kernel eigenvalue problem. Neural Computation, Vol. 10, No. 5 (1998) 1299-1319
11. Y. Zhang, C. Liu: Face recognition using kernel principal component analysis and genetic algorithms. Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing (2002) 337-343
12. C. Liu, H. Wechsler: Evolutionary pursuit and its application to face recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, Vol. 22, No. 6 (2000) 570-582
13. Yoo T. W., et al.: A fast algorithm for tracking human faces based on chromatic histograms. Pattern Recognition Letters, Vol. 20, No. 10 (1999) 967-978