Pattern Recognition Project-2

9957513
多媒體工程所
宋秉一

Content

PCA vs. LDA

PCA results new orthogonal bases in the low dimension feature subspace with the minimal space vector norm. That is, PCA is commonly used for data compression. Instead of minimizing the projected error, the LDA using labeled data to present the more spreadable bases with training dataset. So the LDA is more suitable for data classification but not guarantee minimization of the projected error.

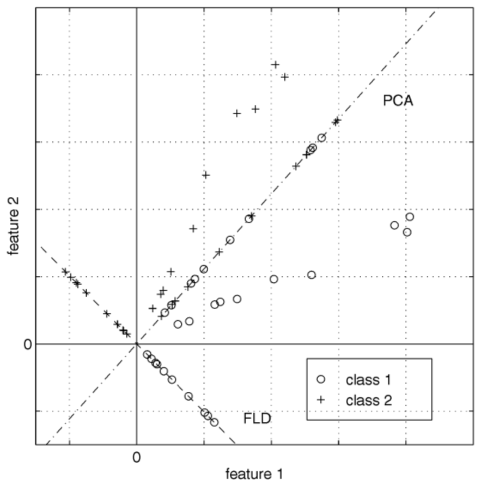


Figure 1.A comparison of PCA and FLD for a two class problem where data for each class lies near a linear subspace [1].

In our project, we will compare the classification results of two methods; PCA and LDA correspond with the Eigen-face and Fisher-face. According to the property of PCA and LDA, we predict that LDA will perform better classification results.

Data Presentation

Given N gray levels face images of c people. Each image is rearranged to a column vector $x_{height \times width}$. Combining all x together we have X

$$X = [x_1 \quad x_2 \quad \cdots \quad x_N]$$

Eigen-face

Mean face

Before PCA applied, we want to remove the features that all people have, which is considered the redundancy for classification. Move the data center to global center. The data center of face images is also called the *mean face*.

$$u = E(X) = \frac{1}{N}\sum_{i=1}^{N} x_i$$

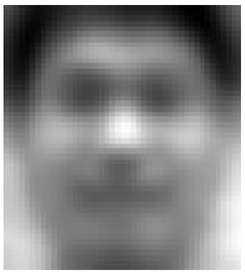


Figure 2.The mean face from training data

Covariance Matrix

Define X subtract the mean face (global mean) and compute the covariance matrix.

$$X = [(x_1 - u) \quad (x_2 - u) \quad \cdots \quad (x_N - u)]$$
$$\Sigma = XX^T$$

Extract the eigen-faces

Apply the eigen-decomposition to the covariance matrix. The eigen-face is the column vector of V (eigenvector). The PCA bases are weighting according the eigenvalues which are the variance of each basis projection.

$$XX^TV = \lambda V$$

Due to the image dimension is larger than the number of training data, the eigen-decomposition can be very inefficient. We can modify the formula as solving the eigen problem of X^TX. Which the eigenvector of XX^T is corresponded with Xv. Note that the columns of Xv must be normalized to length 1 to get unite vectors as the subspace bases.

$$X^TXv = \lambda v$$
$$XX^T(Xv) = \lambda(Xv)$$
$$V = Xv$$

The new bases are orthonormal and span the feature subspace of training data which is known as the eigen-face. Rearrange the column vectors to image size to get the eigen-faces.

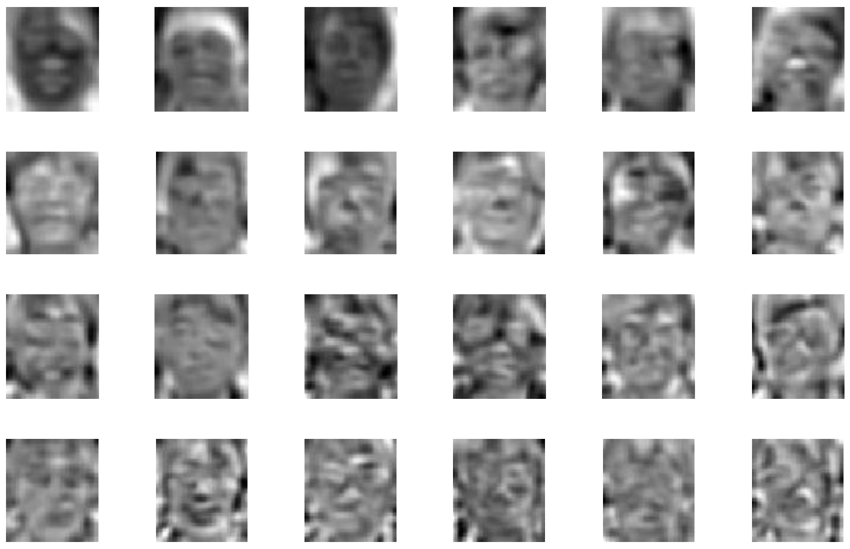


Figure 3.The eigen-faces descending sort with eigenvalue.

The eigen-face corresponded to larger eigenvalue is smoother and lower frequency which implies the major component (left-top image). And each eigen-face is independent with others.

Control the dimension of feature space

To get a lower feature dimension and loss the least data information, we can sort the eigenvalue and select the first K biggest corresponded eigenvectors as the bases. By applying a threshold T, we can control the selected number of basis vectors; the bigger T implies the less projection error.

$$\arg\max_{K}\left\{T \le \frac{\sum_{i=1}^{K}\lambda_K}{trace(\lambda)}, T \in [0,1]\right\}$$

$$V \leftarrow V_{N\times K} = [V_1 \quad V_2 \quad \cdots \quad V_K]$$

Feature space projection

Query a new face image Q and get the feature bases coefficients C_{pca} which are the positions of images in the feature space. The projected coefficients to a set of orthonormal bases can be derived from multiplying an transposed bases matrix.

$$Q \leftarrow Q - u$$

$$Q \cong VC_{pca}$$

$$C_{pca} \cong V^{-1}Q = V^TQ$$

Fisher-face

Scatter matrix

The FLD method use labeled training data to make a more separable feature subspace for classification. Instead of using the covariance matrix, we compute the scatter matrices with labeled data to estimate the dimension relation. The main idea is to maximize the ratio of between-class scatter matrix S_B and within-class scatter matrix S_W.

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - u_i)(x_k - u_i)^T$$

$$S_B = \sum_{i=1}^{c} N_i (u_i - u)(u_i - u)^T$$

To maximize the ratio, we project the scatter matrix into a feature subspace and try to find the bases. This is a generalized eigen-decomposition problem. The w_k is the eigenvector and λ_k is the corresponded eigenvalue.

$$\mathrm{W}_{opt} = \arg\max_W \frac{|W^T S_B W|}{|W^T S_W W|}$$

$$S_B \mathrm{w}_k = \lambda_k S_W \mathrm{w}_k$$

$$S_W^{-1} S_B \mathrm{w}_k = \lambda_k \mathrm{w}_k$$

PCA

In general, the number of image is much smaller than the image dimension (width*height). The rank of S_W is at most $N - c$ and is always singular. To make it invertible, project them into a lower feature dimension using PCA. Note that W_{pca} is performed with orthonormal matrix which is as well as the eigen-face bases we have shown.

$$\mathrm{W}_{pca} = \arg\max_W |W^T S_T W| = V$$

$$\mathrm{W}_{fld} = \arg\max_W \frac{|W^T V^T S_B V W|}{|W^T V^T S_W V W|}$$

$$(V^T S_W V)^{-1} V^T S_B V \mathrm{w}_k = \lambda_k \mathrm{w}_k$$

$$\mathrm{W}_{opt} = V \mathrm{W}_{fld}$$

Where S_T is equivalent to the covariance matrix.

$$S_T = E([X - u][X - u]^T) = S_W + S_B$$

The W_{opt} we solved is known as the Fisher-face as well as the feature subspace bases. The dimension of W_{opt} is reduced to $c - 1$.

Figure 4.Fisher-faces

Experiment Results

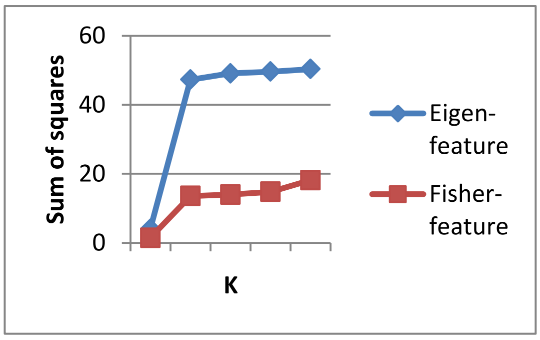
Work flow

We use the K-nearest neighbors to recognize the input face image.

1. Subtract the mean face.
2. Project to the feature space.
3. Find the K-nearest neighbors with training data using Euclidean distance.
4. The input face image is recognized as the biggest vote class (label).

Results

We found that the distance to K is rapidly increased at $\mathrm{K} = 2$. In order to get the correct classification result, we set $\mathrm{K} = 1$ and do the 1-nearest neighbor in the following validation experiments.



K	Sum of squares	
	Eigen	Fisher
1	4.02	1.41
2	47.27	13.54
3	49.08	13.98
4	49.56	14.77
5	50.34	18.15

Figure 5.The Euclidean distances in the feature subspace of different K.

We do the 5-fold cross validation for the experiment. Each round includes 28 training images and 7 testing images.

Round	Eigen-face error rate	Fisher-face error rate
1	0	0
2	0	0
3	0	0
4	0.29	0.29
5	0.29	0.14
Average	0.29	0.22

Table 1.$T = 1$, 28 bases are selected.

Round	Eigen-face error rate	Fisher-face error rate
1	0	0
2	0	0
3	0	0
4	0.43	0.29
5	0.14	0.14
Average	0.29	0.22

Table 2.$T = 0.9$, 14 bases are selected.

The results show that the classification error in Fisher-face is smaller than eigen-face just like we predicted. The following images show that the classification error results with different thresholds used to determine the selection of the bases. The testing image is shown at left-top, and nearest training image at right-top, with its feature image (without adding mean) at the bottom side. The label of each image is titled at image top which red color is presented the error classification.

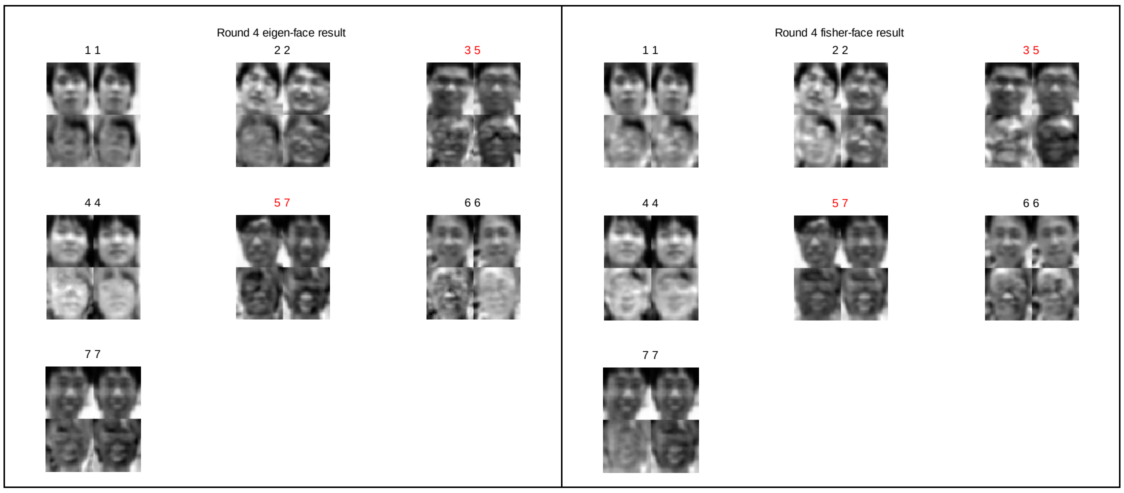


Figure 6.Classification result of round 4 with T=1

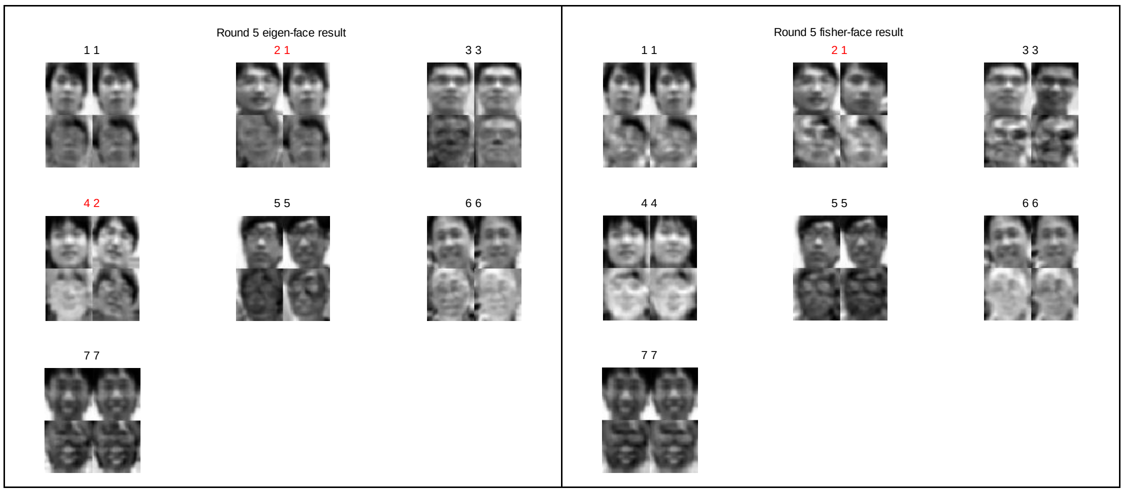


Figure 7.Classification result of round 5 with T=1

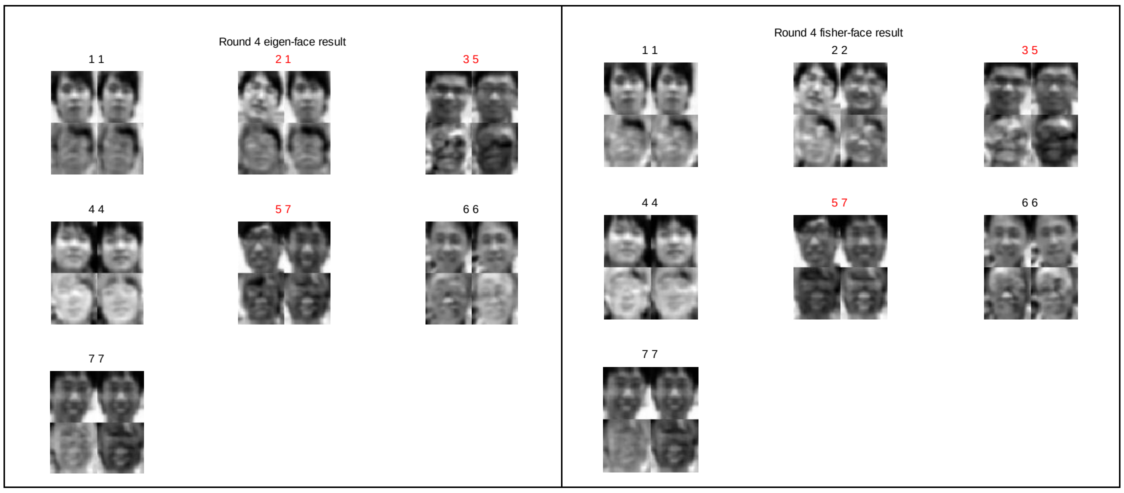


Figure 8.Classification result of round 4 with T=0.9

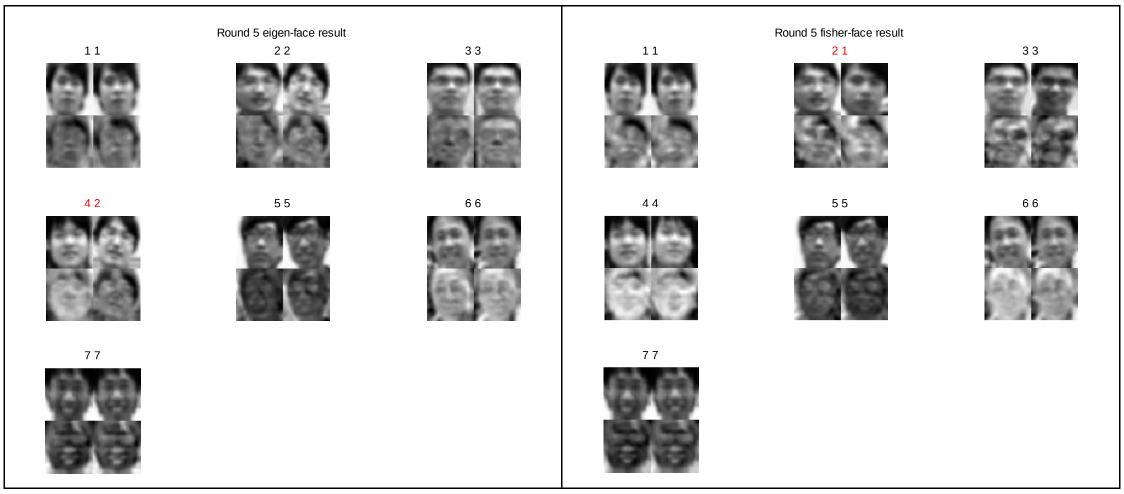


Figure 9.Classification result of round 5 with T=0.9

Discussion

The two methods are depending on the face alignment of image before projecting into feature subspace. The face detector should be robust enough to get an accurate face position and scale in image. And normalize to compare with database.

Both of these feature bases methods are solving the eigen problem with 1D column vectors which are sensitive to the image size. The fisher-face we have used is the 1D basis vector. In order to manage the large scale of 2D images, we could consider the 2D LDA method which is included the *tensor product* conception. Use the hyper plane as the subspace basis and perform better usage of memory and computation time.

Source Code

We use MATLAB to implement the two classifiers.

How to use

The main function includes one input arguments.

T: threshold of eigenvalue ratio [0, 1].

```
>> Main(T)
```

Example

```
>> Main(1)
Round 1
confusionMatrixEigen
     1     0     0     0     0     0     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
eigen-face error rate: 0.00
confusionMatrixFisher
     1     0     0     0     0     0     0
     0     1     0     0     0     0     0
     0     0     1     0     0     0     0
     0     0     0     1     0     0     0
     0     0     0     0     1     0     0
     0     0     0     0     0     1     0
     0     0     0     0     0     0     1
fisher-face error rate: 0.00
```

References

[1]. Peter N. Belhumeur, João P. Hespanha, David J. Kriegman , **Eigenfaces vs. Fisherfaces: Recognition Using Class Specific Linear Projection**, IEEE Transactions on Pattern Analysis and Machine Intelligence - PAMI , vol. 19, no. 7, pp. 711-720, 1997

[2]. Machine Learning for Graphics, Vision and Multimedia http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/

[3]. Scatter matrix http://en.wikipedia.org/wiki/Scatter_matrix