Face Recognition using Principle Component Analysis

Kyungnam Kim
Department of Computer Science
University of Maryland, College Park
MD 20742, USA

Summary

This is the summary of the basic idea about PCA and the papers about the face recognition using PCA.

1. Introduction

The Principal Component Analysis (PCA) is one of the most successful techniques that have been used in image recognition and compression. PCA is a statistical method under the broad title of *factor analysis*. The purpose of PCA is to reduce the large dimensionality of the data space (observed variables) to the smaller intrinsic dimensionality of feature space (independent variables), which are needed to describe the data economically. This is the case when there is a strong correlation between observed variables.

The jobs which PCA can do are prediction, redundancy removal, feature extraction, data compression, etc. Because PCA is a classical technique which can do something in the linear domain, applications having linear models are suitable, such as signal processing, image processing, system and control theory, communications, etc.

Face recognition has many applicable areas. Moreover, it can be categorized into face identification, face classification, or sex determination. The most useful applications contain crowd surveillance, video content indexing, personal identification (ex. driver's licence), mug shots matching, entrance security, etc. The main idea of using PCA for face recognition is to express the large 1-D vector of pixels constructed from 2-D facial image into the compact principal components of the feature space. This can be called eigenspace projection. Eigenspace is calculated by identifying the eigenvectors of the covariance matrix derived from a set of facial images(vectors). The details are described in the following section.

Section 2 describes mathematical formulation of PCA. More details about face recognition by PCA are given in Section 3. Implementation and some results are shown in Section 4. Finally, I present critical reviews in Section 5.

2. Mathematics of PCA

A 2-D facial image can be represented as 1-D vector by concatenating each row (or column) into a long thin vector. Let's suppose we have M vectors of size N (= rows of image × columns of image) representing a set of sampled images. p_j's represent the pixel values.

$$x_i = [p_1 \ldots p_N]^T, i = 1, \ldots, M \tag{1}$$

The images are mean centered by subtracting the mean image from each image vector. Let m represent the mean image.

$$m = \frac{1}{M} \sum_{i=1}^{M} x_i \tag{2}$$

And let w_i be defined as mean centered image

$$w_i = x_i - m \tag{3}$$

Our goal is to find a set of e_i's which have the largest possible projection onto each of the w_i's. We wish to find a set of M orthonormal vectors e_i for which the quantity

$$\lambda_i = \frac{1}{M} \sum_{n=1}^{M} (e_i^T w_n)^2 \tag{4}$$

is maximized with the orthonormality constraint

$$e_l^T e_k = \delta_{lk} \tag{5}$$

It has been shown that the e_i's and λ_i's are given by the eigenvectors and eigenvalues of the covariance matrix

$$C = WW^T \tag{6}$$

where W is a matrix composed of the column vectors w_i placed side by side. The size of C is $N \times N$ which could be enormous. For example, images of size 64×64 create the covariance matrix of size 4096×4096. It is not practical to solve for the eigenvectors of C directly. A common theorem in linear algebra states that the vectors e_i and scalars λ_i can be obtained by solving for the eigenvectors and eigenvalues of the $M \times M$ matrix $W^T W$. Let d_i and μ_i be the eigenvectors and eigenvalues of $W^T W$, respectively.

$$W^T W d_i = \mu_i d_i \tag{7}$$

By multiplying left to both sides by W

$$WW^T(Wd_i) = \mu_i(Wd_i) \tag{8}$$

which means that the first $M - 1$ eigenvectors e_i and eigenvalues λ_i of WW^T are given by Wd_i and μ_i, respectively. Wd_i needs to be normalized in order to be equal to e_i. Since we only sum up a finite number of image vectors, M, the rank of the covariance matrix cannot exceed $M - 1$ (The -1 come from the subtraction of the mean vector m).

The eigenvectors corresponding to nonzero eigenvalues of the covariance matrix produce an orthonormal basis for the subspace within which most image data can be represented with a small amount of error. The eigenvectors are sorted from high to low according to their corresponding eigenvalues. The eigenvector associated with the largest eigenvalue is one that reflects the greatest variance in the image. That is, the smallest eigenvalue is associated with the eigenvector that finds the least variance. They decrease in exponential fashion, meaning that the roughly 90% of the total variance is contained in the first 5% to 10% of the dimensions.

A facial image can be projected onto M' ($\ll M$) dimensions by computing

$$\Omega = [v_1 v_2 \dots v_{M'}]^T \tag{9}$$

where $v_i = e_i^T w_i$. v_i is the i^{th} coordinate of the facial image in the new space, which came to be the principal component. The vectors e_i are also images, so called, *eigenimages*, or *eigenfaces* in our case, which was first named by [1]. They can be viewed as images and indeed look like faces. So, Ω describes the contribution of each eigenface in representing the facial image by treating the eigenfaces as a basis set for facial images. The simplest method for determining which face class provides the best description of an input facial image is to find the face class k that minimizes the Euclidean distance

$$\epsilon_k = \|(\Omega - \Omega_k)\| \tag{10}$$

where Ω_k is a vector describing the k^{th} face class. If ϵ_k is less than some predefined threshold θ_ϵ, a face is classified as belonging to the class k.

3. Face recognition

Once the eigenfaces have been computed, several types of decision can be made depending on the application. What we call face recognition is a broad term which may be further specified to one of following tasks:

- *identification* where the labels of individuals must be obtained,
- *recognition* of a person, where it must be decided if the individual has already been seen,
- *categorization* where the face must be assigned to a certain class.

PCA computes the basis of a space which is represented by its training vectors. These basis vectors, actually eigenvectors, computed by PCA are in the direction of the largest variance of the training vectors. As it has been said earlier, we call them eigenfaces. Each eigenface can be viewed a feature. When a particular face is projected onto the face space, its vector into the face space describe the importance of each of those features in the face. The face is expressed in the face space by its eigenface coefficients (or weights). We can handle a large input vector, facial image, only by taking its small weight vector in the face space. This means that we can reconstruct the original face with some error, since the dimensionality of the image space is much larger than that of face space.

In this report, let's consider face identification only. Each face in the training set is transformed into the face space and its components are stored in memory. The face space has to be populated with these known faces. An input face is given to the system, and then it is projected onto the face space. The system computes its distance from all the stored faces.

However, two issues should be carefully considered:

1. What if the image presented to the system is not a face?
2. What if the face presented to the system has not already learned, i.e., not stored as a known face?

The first defect is easily avoided since the first eigenface is a good face filter which can test whether each image is highly correlated with itself. The images with a low correlation can be rejected. Or these two issues are altogether addressed by categorizing following four different regions:

1. Near face space and near stored face $\Longrightarrow$ known faces
2. Near face space but not near a known face $\Longrightarrow$ unknown faces
3. Distant from face space and near a face class $\Longrightarrow$ non-faces
4. Distant from face space and not near a known class $\Longrightarrow$ non-faces

Since a face is well represented by the face space, its reconstruction should be similar to the original, hence the reconstruction error will be small. Non-face images will have a large reconstruction error which is larger than some threshold θ_r. The distance ϵ_k determines whether the input face is near a known face.

4. Implementation and Results

There is a well-known face database which can be downloadable from the *AT&T Laboratories, Cambridge* at http://www.uk.research.att.com/facedatabase.html. It contains ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open/closed eyes, smiling/not smiling) and facial details (glasses/no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement).

An experiment with a subset of the database, which only contains 12 subject's images, has been performed to ensure how well the eigenface system can identify each individual's face. Please note that this experiment is not exhaustive for the database. The system has been implemented by MATLAB. 10 subjects are selected as training set and other 2 subjects are the part of test set, which should be classified as unknown faces. There are 5 additional test images, each of which is the known face. I also appended 2 non-face images to test whether it can detect them correctly.

Figure 3 shows the eigenface images which are originally the eigenvectors e_i of the covariance matrix at Eq.6. The first eigenface account for the maximal variation of the training vectors. The 10 original training images and their reconstructed versions are depicted in Figure 1 and 2. The result was very successful given the test images in Figure 4. Every test image was correctly classified. When two unknown faces in Figure 5 are input to the system, the ϵ's at Eq. 10 are larger than the predefined threshold. In the case of two non-face images in Figure 5, the reconstruction errors were larger than the reconstruction threshold, then they are not considered as face images. The MATLAB source codes are attached to the appendix in the end of this summary .

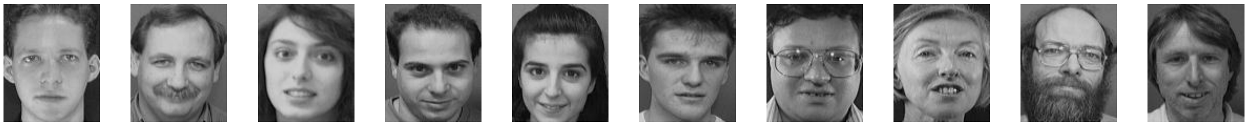

Figure 1: Original training images

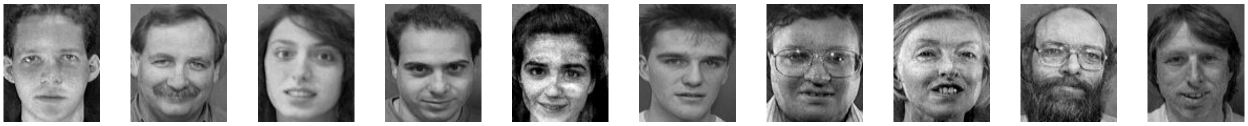

Figure 2: Reconstructed images of training images- they are almost same as their original images

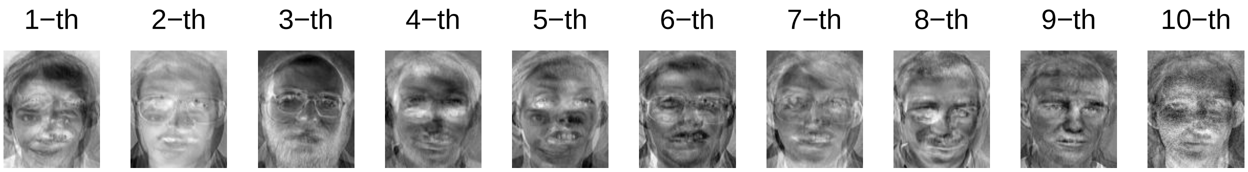

Figure 3: Eigenface - The first eigenface account for the maximal variation of the training vectors, and the second one accounts for the second maximal variation, etc.

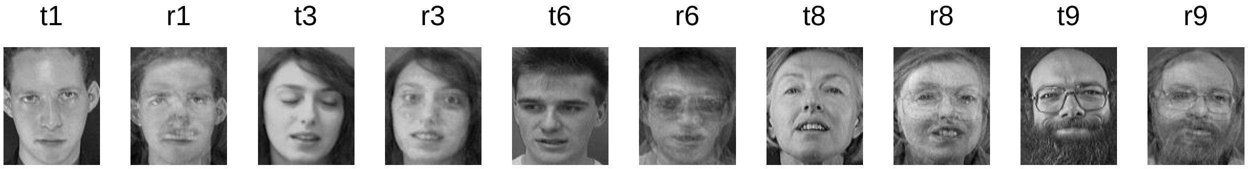

Figure 4: Test images - the number corresponds the order of the set of original training images in Figure 1. r* means the reconstructed image

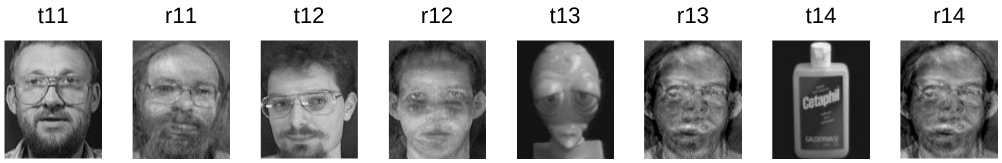

Figure 5: Unknown face and Non-face images - t11 and t12 are unknown faces. t13 and t14 are non-faces. r** means the reconstructed image

5. Critical Reviews

In order to examine more carefully how well the idea works, I did much extensive and exhaustive experiments with different sets of training images. These are done by some criteria not considered by the authors. For the accuracy of the experiments, the training set should not overlapped the test set. The size of the training set was increased, i.e., 1st configuration - 10 subjects each of whom has 8 images (totally 80 images), 2nd configuration - 20 subject each of whom has 8 images (totally 160 images), ... up to 40 subjects which is the maximum number of subject in the given database. The results showed me that the system gave correct identification for the known faces which is trained by the system. The noticeable fact is that the bigger the size of training set is, the less the reconstruction errors are. This is because there are more basis vectors (eigenvectors) to express the given data (faces + non-faces) in the feature space. But this makes the non-faces have less reconstruction error - this is undesirable case. Differentiating whether it is a known face or unknown is kind of hard problem. Deciding appropriate threshold θ_ϵ in Eq.10 is not easy because there is no noticeable difference between the minimum of ϵ of known faces and those of unknown faces. The details are given in the additional paper "Evaluation and Analysis".

As mentioned in Section 2, A facial image can be projected onto much smaller M' ($\ll M$) dimensions. The experiment told me that the identification of known faces is quite good, which is comparable to the case of larger dimensions of M. That it, we can express the face image data in much smaller dimensions with the help of some prominent basis vectors, which in fact reflect maximum variability. Please see "Evaluation and Analysis" for the results.

I would like to enumerate some limitations of the algorithm, which I found from the experiments and careful considerations.

1. The face image should be normalized and frontal-view
2. The system is an auto-associative memory (p.153 in [2]). It is harmful to be overfitted.
3. Training is very computationally intensive.
4. It is hard to decide suitable thresholds - It is kind of Art !
5. The suggested methods to deal with unknown faces and non-faces are not good enough to differentiate them from known faces

Here are some significant questions raised by the articles and also by myself.

- How many eigenvectors (eigenfaces) are required to recover an original face by some percent of the total?
- Are the images of the same subject projected onto similar (near) points in the feature space?
- Is there any idea to decide the threshold automatically?
- Is there another tool to do similar things as PCA do?

I discussed how I might approach some of above limitations and questions in the additional paper "My suggestion".

The eigenface system to perform face recognition gave me the basic idea of PCA and more extensive study of the course material. Although the face recognition results were acceptable, the system only using eigenfaces might not be applicable as a real system. It need to be more robust and to have other discriminant features.

References

[1] M.A. Turk and A.P. Pentland, "Face Recognition Using Eigenfaces", *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 586-591, 1991.

[2] K. I. Diamantaras and S. Y. Kung, "Principal Component Neural Networks: Theory and Applications", *John Wiley & Sons,Inc.*, 1996.

[3] Alex Pentland, Baback Moghaddam, and Thad Starner, "View-Based and Modular Eigenspaces for Face Recognition", *IEEE Conf. on Computer Vision and Pattern Recognition*, MIT Media Laboratory Tech. Report No. 245 1994