We are IntechOpen, the world's leading publisher of Open Access books Built by scientists, for scientists

17

Image-based Subspace Analysis for Face Recognition

Vo Dinh Minh Nhat and SungYoung Lee
Kyung Hee University
Korea

In classical statistical pattern recognition tasks, we usually represent data samples with *n*-dimensional vectors, i.e. data is vectorized to form data vectors before applying any technique. However in many real applications, the dimension of those 1D data vectors is very high, leading to the "curse of dimensionality". The curse of dimensionality is a significant obstacle in pattern recognition and machine learning problems that involve learning from few data samples in a high-dimensional feature space. In face recognition, Principal component analysis (PCA) and Linear discriminant analysis (LDA) are the most popular subspace analysis approaches to learn the low-dimensional structure of high dimensional data. But PCA and LDA are based on 1D vectors transformed from image matrices, leading to lose structure information and make the evaluation of the covariance matrices high cost. In this chapter, straightforward image projection techniques are introduced for image feature extraction. As opposed to conventional PCA and LDA, the matrix-based subspace analysis is based on 2D matrices rather than 1D vectors. That is, the image matrix does not need to be previously transformed into a vector. Instead, an image covariance matrix can be constructed directly using the original image matrices. We use the terms "matrix-based" and "image-based" subspace analysis interchangeably in this chapter. In contrast to the covariance matrix of PCA and LDA, the size of the image covariance matrix using image-based approaches is much smaller. As a result, it has two important advantages over traditional PCA and LDA. First, it is easier to evaluate the covariance matrix accurately. Second, less time is required to determine the corresponding eigenvectors (Jian Yang et al., 2004). A brief of history of image-based subspace analysis can be summarized as follow. Based on PCA, some image-based subspace analysis approaches have been developed such as 2DPCA (Jian Yang et al., 2004), GLRAM (Jieping Ye, 2004), Non-iterative GLRAM (Jun Liu & Songcan Chen 2006; Zhizheng Liang et al., 2007), MatPCA (Songcan Chen, et al. 2005), 2DSVD (Chris Ding & Jieping Ye 2005), Concurrent subspace analysis (D.Xu, et al. 2005) and so on. Based on LDA, 2DLDA (Ming Li & Baozong Yuan 2004), MatFLDA (Songcan Chen, et al. 2005), Iterative 2DLDA (Jieping Ye, et al. 2004), Non-iterative 2DLDA (Inoue, K. & Urahama, K. 2006) have been developed until date. The main purpose of this chapter is to give you a generalized overview of those matrix-based approaches with detailed mathematical theory behind that. All algorithms presented here are up-to-date till Jan. 2007.

Source: Face Recognition, Book edited by: Kresimir Delac and Mislav Grgic, ISBN 978-3-902613-03-5, pp.558, I-Tech, Vienna, Austria, June 2007

1. Introduction

A facial recognition system is a computer-driven application for automatically identifying a person from a digital image. It does that by comparing selected facial features in the live image and a facial database. With the rapidly increasing demand on face recognition technology, it is not surprising to see an overwhelming amount of research publications on this topic in recent years. In this chapter we briefly review on linear subspace analysis (LSA), which is one of the fastest growing areas in face recognition research and present in detail recently developed image-based approaches.

Method	Reference	Section
PCA	(M. Turk & A. Pentland 1991)	2.1
LDA	(Belhumeur P.N., et al., 1997)	2.2
2DPCA	(Jian Yang et al., 2004) MatPCA (Songcan Chen, et al. 2005)	3.1
2DLDA	(Ming Li & Baozong Yuan 2004) MatFLDA (Songcan Chen, et al. 2005)	3.2
GLRAM	(Jieping Ye, 2004) Concurrent subspace analysis (D.Xu, et al. 2005) 2DSVD (Chris Ding & Jieping Ye 2005)	4.1
Non-iterative GLRAM	(Zhizheng Liang et al., 2007)	4.2
Iterative 2DLDA	(Jieping Ye, et al. 2004)	4.3
Non-iterative 2DLDA	(Inoue, K. & Urahama, K. 2006)	4.4

Table 1. Summary of these algorithms presented in this chapter

LSA has gained much attention in a wide range of problems arising in image processing, computer vision and especially pattern recognition. In LSA, the singular value decomposition (SVD) is usually the basic mathematical tool. The most popular LSA methods used in Face Recognition (FR) are Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). PCA (M. Turk & A. Pentland 1991) is a subspace projection technique widely used for face recognition. It finds a set of representative projection vectors such that the projected samples retain most information about original samples. The most representative vectors are the eigenvectors corresponding to the largest eigenvalues of the covariance matrix. Unlike PCA, LDA (Belhumeur P.N., et al., 1997) finds a set of vectors that maximizes Fisher Discriminant Criterion. It simultaneously maximizes the between-class scatter while minimizing the within-class scatter in the projective feature vector space. While PCA can be called unsupervised learning techniques, LDA is supervised learning technique because it needs class information for each image in the training process. In above approaches, the image data first needs to be transformed into vectors before any further processing. Recently, two-dimensional PCA (2DPCA) and two-dimensional LDA (2DLDA) have been proposed in which image covariance matrices can be constructed directly using original image matrices. In contrast to the covariance matrices of traditional approaches (PCA and LDA), the size of the image covariance matrices using 2D approaches (2DPCA and 2DLDA) are much smaller. As a result, it is easier to evaluate the covariance matrices accurately, computation cost is reduced and the performance is also improved (Jian Yang et al., 2004). We categorize the existing techniques in image-based subspace analysis into two main categories. One category can be considered as a one-sided low-rank approximation which includes 2DPCA (Jian Yang et al., 2004), MatPCA (Songcan Chen, et al. 2005), 2DLDA (Ming Li & Baozong Yuan 2004), and MatLDA (Songcan Chen, et al. 2005). The other is classified as two-sided low-rank approximation such as GLRAM (Jieping Ye, 2004), Non-iterative GLRAM (Jun Liu & Songcan Chen 2006; Zhizheng Liang et al., 2007), 2DSVD (Chris Ding & Jieping Ye 2005), Concurrent subspace analysis (D.Xu, et al. 2005), Iterative 2DLDA (Jieping Ye, et al. 2004), and Non-iterative 2DLDA (Inoue, K. & Urahama, K. 2006). Tabel 1. gives an summary of those algorithms presented. Basis notations used in this chapter are summarized in Table 2.

Notations	Descriptions
$x_i \in \Re^n$	the i^{th} image point in vector form
$X_i \in \Re^{r \times c}$	the i^{th} image point in matrix form
Π_i	the i^{th} class of data points (both in vector and matrix form)
n	dimension of x_i
m	dimension of reduced feature vector y_i
r	number of rows in X_i
c	number of columns in X_i
N	number of data samples
C	number of classes
N_i	number of data samples in class Π_i
L	transformation on the left side
R	transformation on the right side
l_1	number of rows in Y_i
l_2	number of columns in Y_i

Table 2. Notations and Descriptions

2. Linear Subspace Analysis Introduction

In this section we briefly review about LSA which includes PCA and LDA. One approach to cope with the problem of excessive dimensionality of the image space is to reduce the dimensionality by combining features. Linear combinations are particularly attractive because they are simple to compute and analytically tractable. In effect, linear methods project the high-dimensional data onto a lower dimensional subspace. Suppose that we have N sample images $\{x_1, x_2, ..., x_N\}$ taking values in an n-dimensional image space. Let us also consider a linear transformation mapping the original n-dimensional image space into an m-dimensional feature space, where $m < n$. The new feature vectors $y_k \in \Re^m$ are defined by the following linear transformation:

$$y_k = W^T(x_k - \mu) \tag{1}$$

where $k = 1,2,...,N$, $\mu \in \Re^n$ is the mean of all samples, and $W \in \Re^{n \times m}$ is a matrix with orthonormal columns. Atfer the linear transformation, each data point x_k can be represented by a feature vector $y_k \in \Re^m$ which is used for classification.

2.1 Principal Component Analysis - PCA

Different objective functions will yield different algorithms with different properties. PCA aims to extract a subspace in which the variance is maximized. Its objective function is as follows:

$$W_{opt} = [w_1 w_2 ... w_m] = \arg\max_W \left| W^T S_t W \right| \tag{2}$$

with the total scatter matrix is defined as

$$S_t = \frac{1}{N}\sum_{k=1}^{N}(x_k - \mu)(x_k - \mu)^T \tag{3}$$

and $\mu = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the mean of all samples. The optimal projection $W_{opt} = [w_1 w_2 ... w_m]$ is the set of n-dimensional eigenvectors of S_t corresponding to the m largest eigenvalues.

2.2 Linear Discriminant Analysis - LDA

While PCA seeks directions that are efficient for representation, LDA seeks directions that are efficient for discrimination. Assume that each image belongs to one of C classes $\{\Pi_1, \Pi_2, ..., \Pi_C\}$. Let N_i be the number of the samples in class $\Pi_i (i = 1,2,...,C)$, $\mu_i = \frac{1}{N_i}\sum_{x \in \Pi_i} x$ be the mean of the samples in class Π_i. Then the between-class scatter matrix S_b is defined as

$$S_b = \frac{1}{N}\sum_{i=1}^{C} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4}$$

and the within-class scatter matrix S_w is defined as

$$S_w = \frac{1}{N}\sum_{i=1}^{C}\sum_{x_k \in \Pi_i}(x_k - \mu_i)(x_k - \mu_i)^T \tag{5}$$

In LDA, the projection W_{opt} is chosen to maximize the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples, i.e.,

$$W_{opt} = \arg\max_W \frac{\left|W^T S_b W\right|}{\left|W^T S_w W\right|} = [w_1 w_2 ... w_m] \quad (6)$$

where $\{w_i \mid i = 1, 2, ..., m\}$ is the set of generalized eigenvectors of S_b and S_w corresponding to the m largest generalized eigenvalues $\{\lambda_i \mid i = 1, 2, ..., m\}$, i.e.,

$$S_b w_i = \lambda_i S_w w_i \quad i = 1, 2, ..., m \quad (7)$$

3. One-sided Image-based Subspace Analysis

In previous section, we review the linear subspace analysis techniques which are based on 1D vectors. However, recently, (Yang et al., 2004) proposed a novel image representation and recognition technique, two-dimensional PCA (2DPCA). 2DPCA has many advantages over classical PCA. In classical PCA, an image matrix should be mapped into a 1D vector in advance. 2DPCA, however, can directly extract feature matrix from the original image matrix. This leads to that much less time is required for training and feature extraction. Further, the recognition performance of 2DPCA is better than that of classical PCA. Inspired by (Yang et al., 2004), a lot of algorithms have been developed based directly on matrix images. As mentioned, we cagetorize those image-based approaches into two main cagetories which are one-side low-rank approximation and two-sided low-rank approximation. In this section, we present two one-sided low-rank approximations which are 2DPCA and 2DLDA approaches.

3.1 Two-dimensional PCA (2DPCA)

As mentioned above, in 2D approach, the image matrix does not need to be previously transformed into a vector, so a set of N sample images is represented as $\{X_1, X_2, ..., X_N\}$ with $X_i \in \Re^{r \times c}$, which is a matrix space of size $r \times c$. The total scatter matrix is defined as

$$G_t = \frac{1}{N} \sum_{i=1}^{N} (X_i - M)^T (X_i - M) \quad (8)$$

with $M = \frac{1}{N} \sum_{i=1}^{N} X_i \in \Re^{r \times c}$ is the mean image of all samples. $G_t \in \Re^{r \times r}$ is also called image covariance (scatter) matrix. A linear transformation mapping the original $r \times c$ image space into an $r \times m$ feature space, where $m < c$. The new feature matrices $Y_i \in \Re^{r \times m}$ are defined by the following linear transformation:

$$Y_i = (X_i - M)W \in \Re^{r \times m} \quad (9)$$

where $i = 1, 2, ..., N$ and $W \in \Re^{r \times m}$ is a matrix with orthogonal columns. In 2DPCA, the projection W_{opt} is chosen to maximize $tr(W^T G_t W)$. The optimal projection $W_{opt} = [w_1 w_2 ... w_m]$

with $\{w_i | i = 1, 2, ..., m\}$ is the set of c-dimensional eigenvectors of G_t corresponding to the m largest eigenvalues.

3.2 Two-dimensional LDA (2DLDA)

In 2DLDA, the between-class scatter matrix S_b is re-defined as

$$G_b = \frac{1}{N}\sum_{i=1}^{C} N_i (M_i - M)^T (M_i - M) \tag{10}$$

and the within-class scatter matrix S_w is re-defined as

$$G_w = \frac{1}{N}\sum_{i=1}^{C} \sum_{X_k \in C_i} (X_k - M_i)^T (X_k - M_i) \tag{11}$$

with $M = \frac{1}{N}\sum_{i=1}^{N} X_i \in \Re^{r \times c}$ is the mean image of all samples and $M_i = \frac{1}{N_i}\sum_{X_k \in \Pi_i} X_k \in \Re^{r \times c}$ be the mean of the samples in class $\Pi_i (i = 1..C)$. Similarly, a linear transformation mapping the original $r \times c$ image space into an $r \times m$ feature space, where $m < c$. The new feature matrices $Y_i \in \Re^{r \times m}$ are defined by the following linear transformation :

$$Y_i = (X_i - M)W \in \Re^{r \times m} \tag{12}$$

where $i = 1, 2, ..., N$ and $W \in \Re^{c \times m}$ is a matrix with orthogonal columns. And the projection W_{opt} is chosen with the criterion same as that in (6). While the classical LDA must face to the singularity problem, we can see that 2DLDA overcomes this problem. We need to prove that G_w^{-1} exists, i.e. $rank(G_w) = c$. We have,

$$\begin{aligned} rank(G_w) &= rank\left(\frac{1}{N}\sum_{i=1}^{C} \sum_{X_k \in C_i} (X_k - M_i)^T (X_k - M_i)\right) \\ &\le (N - C) * \min(r, c) \end{aligned} \tag{13}$$

The inequality in (13) holds because $rank(X_i) = \min(r, c)$. So, in 2DLDA, G_w is nonsingular when

$$\begin{aligned} & c \le (N - C) * \min(r, c) \\ \Leftrightarrow\ & N \ge C + \frac{c}{\min(r, c)} \end{aligned} \tag{14}$$

In real situation, (14) is always true, so G_w is always nonsingular.

3.3 Classifier for 2DPCA and 2DLDA

After a transformation by 2DPCA or 2DLDA, a feature matrix is obtained for each image. Then, a nearest neighbor classifier is used for classification. Here, the distance between two arbitrary feature matrices Y_i and Y_j is defined by using Euclidean distance as follows:

$$d(Y_i, Y_j) = \sqrt{\sum_{u=1}^{k}\sum_{v=1}^{s}(Y_i(u,v) - Y_j(u,v))^2} \tag{15}$$

Given a test sample Y_t, if $d(Y_t, Y_c) = \min_j d(Y_t, Y_j)$, then the resulting decision is Y_t belongs to the same class as Y_c.

4. Two-sided Image-based Subspace Analysis

4.1 Generalized Low Rank Approximations of Matrices (GLRAM)

In paper (Jieping Ye, 2004), Jieping considered the problem of computing low rank approximations of matrices which are based on a collection of matrices. By solving an optimization problem, which aims to minimize the reconstruction (approximation) error, they derive an iterative algorithm, namely GLRAM, which stands for the Generalized Low Rank Approximations of Matrices. GLRAM reduces the reconstruction error sequentially, and the resulting approximation is thus improved during successive iterations. Formally, they consider the following optimization problem

$$\begin{aligned}&\min_{L,R,Y_i}\sum_{i=1}^{N}\left\|X_i - LY_iR^T\right\|_F^2\\ &s.t.\quad L^TL = I_1, R^TR = I_2\end{aligned} \tag{16}$$

where $L \in \Re^{r\times l_1}$, $R \in \Re^{c\times l_2}$, $Y_i \in \Re^{l_1\times l_2}$ for $i = 1..N$, $I_1 \in \Re^{l_1\times l_1}$ and $I_2 \in \Re^{l_2\times l_2}$ are identity matrices, where $l_1 \le r$ and $l_2 \le c$. Before showing how to solve above optimization problem, we briefly review some theorems that support the final iterative algorithm.

Theorem 1. Let L, R and $\{Y_i\}_{i=1}^N$ be the optimal solution to the minimization problem in Eq. (16). Then $Y_i = L^T X_i R$ for every i.

Proof: By the property of the trace of matrices,

$$\begin{aligned}\sum_{i=1}^{N}\left\|X_i - LY_iR^T\right\|_F^2 &= \sum_{i=1}^{N} tr\left((X_i - LY_iR^T)(X_i - LY_iR^T)^T\right)\\ &= \sum_{i=1}^{N} tr\left(X_iX_i^T\right) + \sum_{i=1}^{N} tr\left(Y_iY_i^T\right) - 2\sum_{i=1}^{N} tr\left(LY_iR^TX_i^T\right)\end{aligned} \tag{17}$$

Because $\sum_{i=1}^{N} tr\left(X_iX_i^T\right)$ is a constant, the minimization in Eq. (16) is equivalent to minimizing

$$E = \sum_{i=1}^{N} tr\left(Y_iY_i^T\right) - 2\sum_{i=1}^{N} tr\left(LY_iR^TX_i^T\right) \tag{18}$$

By taking derivatives of (18) , and force it equal to zero

$$\frac{\partial E}{\partial Y_i} = 2Y_i^T - 2R^T X_i^T L = 0 \tag{19}$$

we obtain $Y_i = L^T X_i R$. This completes the proof of the theorem.

Theorem 2. Let L, R and $\{Y_i\}_{i=1}^N$ be the optimal solution to the minimization problem in Eq. (16). Then L, R solve the following optimization problem:

$$\max_{L,R,Y_i} \sum_{i=1}^N \left\| L^T X_i R \right\|_F^2 \quad s.t. \quad L^T L = I_1, R^T R = I_2 \tag{20}$$

Proof: From Theorem 1., $Y_i = L^T X_i R$ for every i, we obtain

$$\begin{aligned} &\sum_{i=1}^N tr\left(Y_i Y_i^T\right) - 2\sum_{i=1}^N tr\left(L Y_i R^T X_i^T\right) \\ &= \sum_{i=1}^N tr\left(L^T X_i R R^T X_i^T L\right) - 2\sum_{i=1}^N tr\left(L L^T X_i R R^T X_i^T\right) \\ &= -\sum_{i=1}^N tr\left(L^T X_i R R^T X_i^T L\right) = -\sum_{i=1}^N \left\| L^T X_i R \right\|_F^2 \end{aligned} \tag{21}$$

Hence the minimization problem in Eq. (16) is equivalent to the maximization of

$$\max_{L,R,Y_i} \sum_{i=1}^N \left\| L^T X_i R \right\|_F^2 \quad s.t. \quad L^T L = I_1, R^T R = I_2 \tag{22}$$

To the best of our knowledge, there is no closed form solution for the maximization in Eq. (22). A key observation, which leads to an iterative algorith for the computation of L, R, is stated in the following theorem:

Theorem 3. Let L, R and $\{Y_i\}_{i=1}^N$ be the optimal solution to the minimization problem in Eq. (). Then,

(1) For a given R , L consists of the l_1 eigenvectors of the matrix

$$S_L = \sum_{i=1}^N X_i R R^T X_i^T \tag{23}$$

corresponding to the largest l_1 eigenvalues.

(2) For a given L , R consists of the l_2 eigenvectors of the matrix

$$S_R = \sum_{i=1}^N X_i^T L L^T X_i \tag{24}$$

corresponding to the largest l_2 eigenvalues.

Proof: From the Theorem 2., the objective function in (22) can be re-written as

$$\begin{aligned}\sum_{i=1}^{N}\left\|L^T X_i R\right\|_F^2 &= \sum_{i=1}^{N} tr\left(L^T X_i R R^T X_i^T L\right)\\ &= tr\left(L^T\left(\sum_{i=1}^{N} X_i R R^T X_i^T\right)L\right) = tr\left(L^T S_L L\right)\end{aligned} \tag{25}$$

where $S_L = \sum_{i=1}^{N} X_i R R^T X_i^T$. Hence for a given R, $L \in \Re^{r\times l_1}$ consists of the l_1 eigenvectors of the matrix S_L corresponding to the largest l_1 eigenvalues. Similarly, For a given L, $R \in \Re^{c\times l_2}$ consists of the l_2 eigenvectors of the matrix $S_R = \sum_{i=1}^{N} X_i^T L L^T X_i$ corresponding to the largest l_2 eigenvalues. This completes the proof of the thereom. An iterative procedure for computing L and R can be presented as follow

Algorithm – GLRAM

Step 0

Initialize $L = L^{(0)} = [I_1, 0]^T$, and set $k = 0$.

Step 1

Compute l_2 eigenvectors $\{\Phi_i^{R(k+1)}\}_{i=1}^{l_2}$ of the matrix $S_R = \sum_{i=1}^{N} X_i^T L^{(k)} L^{(k)T} X_i$ corresponding to the largest l_2 eigenvalues and form $R^{(k+1)} = [\Phi_1^{R(k+1)}..\Phi_{l_2}^{R(k+1)}]$.

Step 2

Compute l_1 eigenvectors $\{\Phi_i^{L(k+1)}\}_{i=1}^{l_1}$ of the matrix $S_L = \sum_{i=1}^{N} X_i R^{(k+1)} R^{(k+1)T} X_i^T$ corresponding to the largest l_1 eigenvalues and form $L^{(k+1)} = [\Phi_1^{L(k+1)}..\Phi_{l_1}^{L(k+1)}]$.

Step 3

If $L^{(k+1)}$, $R^{(k+1)}$ are not convergent then set increase k by 1 and go to Step 1, otherwise proceed to Step 4.

Step 4

Let $L^* = L^{(k+1)}$, $R^* = R^{(k+1)}$ and compute $Y_i^* = L^{*T} X_i R^*$ for $i = 1..N$.

4.2 Non-iterative GLRAM

By further analyzing GLRAM, it is of interest to note that the objective function in Eq. (16) (Zhizheng Liang et al., 2007) has the lower and upper bound in terms of the covariance matrix. They also derive an effective solution for GLRAM which is a non-iterative solution. In the following, we first provide a lemma which is very useful for developing non-iterative GLRAM algorithm.

Lemma 1. Let B be an $m\times m$ symmetric matrix and H be an $m\times h$ which satisties $H^T H = I \in \Re^{h\times h}$. Then, for $i = 1..h$, we have

$$\lambda_{m-h+i}(B) \le \lambda_i(H^T B H) \le \lambda_i(B) \tag{26}$$

where $\lambda_i(B)$ denotes the i^{th} largest eigenvalue of the matrix B.
Proof of this lemma can be referenced in (Zhizheng Liang et al., 2007). From Lemma 1., the following corollary can be obtained
Corollary 1. Let w_i be the eigenvectors corresponding to the i^{th} largest eigenvalue λ_i of B and H be an $m\times h$ which satisties $H^T H = I \in \Re^{h\times h}$. Then,

$$\lambda_{m-h+i} + .. + \lambda_m \le tr(H^T BH) \le \lambda_1 + .. + \lambda_h \tag{27}$$

and the second equality holds if $H = WQ$ where $W = [w_1..w_h]$ and Q is any $h\times h$ orthogonal matrix.
Some following matrices are defined (Zhizheng Liang et al., 2007)

$$G_1 = \sum_{i=1}^{N} X_i^T X_i \tag{28}$$

$$G_2 = \sum_{i=1}^{N} X_i X_i^T \tag{29}$$

Let F_1 consists of the eigenvectors of G_2 corresponding to the first l_2 largest eigenvalues and F_2 consists of the eigenvectors of G_1 corresponding to the first l_1 largest eigenvalues. Next, we define

$$H_{L1} = \sum_{i=1}^{N} X_i F_1 F_1^T X_i^T \tag{30}$$

$$H_{R1} = \sum_{i=1}^{N} X_i^T F_2 F_2^T X_i \tag{31}$$

Let K_1 consists of the eigenvectors of H_{L1} corresponding to the first l_1 largest eigenvalues and K_2 consists of the eigenvectors of H_{R1} corresponding to the first l_2 largest eigenvalues. Applying Corollary 1., we can obtain the following theorem
Theorem 4. Let d_1 be the sum of the first l_1 largest eigenvalues of H_{L1} and d_2 be the sum of the first l_2 largest eigenvalues of H_{R1}. In such a case, the value of Eq. (22) is equal to $\max\{d_1, d_2\}$
Proof : (a) Eq. (22) can be represented as

$$\begin{aligned}\sum_{i=1}^{N} \left\| L^T X_i R \right\|_F^2 &= \sum_{i=1}^{N} tr\left(L^T X_i R R^T X_i^T L\right) \\ &= tr\left(L^T \left(\sum_{i=1}^{N} X_i R R^T X_i^T\right) L\right) = tr\left(L^T S_L L\right)\end{aligned} \tag{32}$$

Applying Corollary 1. we have

$$tr\left(L^T S_L L\right) \le tr\left(S_L\right)_{l_1} \tag{33}$$

Since

$$tr\left(S_L\right)=tr\left(\sum_{i=1}^{N}X_iRR^TX_i^T\right)=tr(R^TG_1R)\le tr(G_1)_{l_2} \tag{34}$$

From Eq. (33) and Eq. (34), we can obtain

$$tr\left(L^TS_LL\right)\le tr\left(G_1\right)_{l_2} \tag{35}$$

Then it is not difficult to obtain $R=F_1Q^2_{l_2\times l_2}$ where $Q^2_{l_2\times l_2}$ is any orthogonal matrix. Substitute $R=F_1Q^2_{l_2\times l_2}$ into S_L and obtrain H_{L1}, we can have $L=K_1Q^1_{l_1\times l_1}$. Futhermore, it is straightforward to verify that the value of Eq. (22) is equal to d_1.

(b) In the same way we can have

$$\begin{aligned}\sum_{i=1}^{N}\left\|L^TX_iR\right\|_F^2&=\sum_{i=1}^{N}tr\left(L^TX_iRR^TX_i^TL\right)=\sum_{i=1}^{N}tr\left(R^TX_i^TLL^TX_iR\right)\\&=tr\left(R^T\left(\sum_{i=1}^{N}X_i^T{}_iLL^TX_i\right)R\right)=tr\left(R^TS_RR\right)\end{aligned} \tag{36}$$

Applying Corollary 1. we have

$$tr\left(R^TS_RR\right)\le tr\left(S_R\right)_{l_2} \tag{37}$$

Since

$$tr\left(S_R\right)=tr\left(\sum_{i=1}^{N}X_i^TLL^TX_i\right)=tr(L^TG_2L)\le tr(G_2)_{l_1} \tag{38}$$

From Eq. (37) and Eq. (38), we can obtain

$$tr\left(R^TS_RR\right)\le tr\left(G_2\right)_{l_1} \tag{39}$$

Then it is not difficult to obtain $L=F_2Q^1_{l_1\times l_1}$ where $Q^1_{l_1\times l_1}$ is any orthogonal matrix. Substitute $L=F_2Q^1_{l_1\times l_1}$ into S_R and obtrain H_{R1}, we can have $R=K_2Q^2_{l_2\times l_2}$. Futhermore, it is straightforward to verify that the value of Eq. (22) is equal to d_2. From (a) and (b), the theorem is proven. From this proof, it is not difficult to derive the non-iterative GLRAM as

Algorithm – Non-iterative GLRAM

Step 1

Compute the matrices G_1 and G_2

Step 2

Compute eigenvectors of the matrices G_1 and G_2, let $R=F_1Q^2_{l_2\times l_2}$ and $L=F_2Q^1_{l_1\times l_1}$

Step 3

Compute eigenvectors of the matrices H_{L1} and H_{R1}, and obtain $L=K_1Q^1_{l_1\times l_1}$

corresponding to R in step 2 and $R = K_2 Q^2_{l_2 \times l_2}$ corresponding to L in step 2 and compute d_1 , d_2

Step 4

Choose R, L corresponding to $\max\{d_1, d_2\}$, and compute $Y_i = L^T X_i R$

4.3 Iterative 2DLDA

In (Jieping Ye, et al. 2004), he proposed a novel LDA algorithm, namely 2DLDA, which stands for 2-Dimensional Linear Discriminant Analysis. However, to distinguish with previous 2DLDA approach, we call this approach Iterative 2DLDA. Iterative 2DLDA aims to find the two-sided optimal transformations (projections L and R) such that the class structure of the original high-dimensional space is preserved in the low-dimensional space. A natural similarity metric between matrices is the Frobenius norm. Under this metric, the (squared) within-class and between-class distances D_w and D_b can be computed as follows:

$$\begin{aligned} D_w &= \sum_{j=1}^{C} \sum_{X_i \in \Pi_j} \left\| X_i - M_j \right\|_F^2 \\ &= tr\left(\sum_{j=1}^{C} \sum_{X_i \in \Pi_j} (X_i - M_j)(X_i - M_j)^T \right) \end{aligned} \quad (40)$$

$$\begin{aligned} D_b &= \sum_{j=1}^{C} N_j \left\| M_j - M \right\|_F^2 \\ &= tr\left(\sum_{j=1}^{C} N_j (M_j - M)(M_j - M)^T \right) \end{aligned} \quad (41)$$

In the low-dimensional space resulting from the linear transformations L and R , the with-in and between-class distances $\tilde{D}_w$ and $\tilde{D}_b$ can be computed as follows:

$$\tilde{D}_w = tr\left(\sum_{j=1}^{C} \sum_{X_i \in \Pi_j} L^T (X_i - M_j) R R^T (X_i - M_j)^T L \right) \quad (42)$$

$$\tilde{D}_b = tr\left(\sum_{j=1}^{C} N_j L^T (M_j - M) R R^T (M_j - M)^T L \right) \quad (43)$$

The optimal transformations L and R would maximize $F(L, R) = \tilde{D}_b / \tilde{D}_w$. Let us define

$$S_w^R = \sum_{X_i \in \Pi_j} (X_i - M_j) R R^T (X_i - M_j)^T \quad (44)$$

$$S_b^R = \sum_{j=1}^{C} N_j (M_j - M) R R^T (M_j - M)^T \quad (45)$$

$$S_w^L = \sum_{X_i \in \Pi_j} (X_i - M_j)^T LL^T (X_i - M_j) \tag{46}$$

$$S_b^L = \sum_{j=1}^{C} N_j (M_j - M)^T LL^T (M_j - M) \tag{47}$$

After defining those matrices we can derive the iterative 2DLDA algorithm as follow

Algorithm – Iterative 2DLDA

Step 0

Initialize $R = R^{(0)} = [I_2, 0]^T$, and set $k = 0$.

Step 1

Compute

$$S_w^{R(k)} = \sum_{X_i \in \Pi_j} (X_i - M_j) R^{(k)} R^{(k)T} (X_i - M_j)^T$$

$$S_b^{R(k)} = \sum_{j=1}^{C} N_j (M_j - M) R^{(k)} R^{(k)T} (M_j - M)^T$$

Step 2

Compute l_1 eigenvectors $\{\Phi_i^{L(k)}\}_{i=1}^{l_1}$ of the matrix $(S_w^{R(k)})^{-1} S_b^{R(k)}$ and form $L^{(k)} = [\Phi_1^{L(k)} .. \Phi_{l_1}^{L(k)}]$.

Step 3

Compute

$$S_w^{L(k)} = \sum_{X_i \in \Pi_j} (X_i - M_j)^T L^{(k)} L^{(k)T} (X_i - M_j)$$

$$S_b^{L(k)} = \sum_{j=1}^{C} N_j (M_j - M)^T L^{(k)} L^{(k)T} (M_j - M)$$

Step 4

Compute l_2 eigenvectors $\{\Phi_i^{R(k)}\}_{i=1}^{l_2}$ of the matrix $(S_w^{L(k)})^{-1} S_b^{L(k)}$ and form $R^{(k+1)} = [\Phi_1^{R(k)} .. \Phi_{l_1}^{R(k)}]$.

Step 5

If $L^{(k)}$, $R^{(k+1)}$ are not convergent then set increase k by 1 and go to Step 1, othervise proceed to Step 6.

Step 6

Let $L^* = L^{(k)}$, $R^* = R^{(k+1)}$ and compute $Y_i^* = L^{*T} X_i R^*$ for $i = 1..N$.

4.4 Non-iterative 2DLDA

Iterative 2DLDA computes L and R in turn with the initialization $R = R^{(0)} = [I_2, 0]^T$. Alternatively, we can consider another algorithm that computes and L and R in turn with the initialization $L = L^{(0)} = [I_1, 0]^T$. By unifying them, in this subsection, we can select L and

R which give larger $F(L,R)$ and form the selective algorithm as follow (Inoue, K. & Urahama, K. 2006)

Algorithm – Selective 2DLDA

Step 1

Initialize $R=[I_2,0]^T$, and compute L and R in turn. Let $L^{(1)}$ and $R^{(1)}$ be computed L and R.

Step 2

Initialize $L=[I_1,0]^T$, and compute L and R in turn. Let $L^{(2)}$ and $R^{(2)}$ be computed L and R.

Step 3

If $f(L^{(1)},R^{(1)}) \ge f(L^{(2)},R^{(2)})$ then output $L=L^{(1)}$ and $R=R^{(1)}$, otherwise output $L=L^{(2)}$ and $R=R^{(2)}$

Also in (Inoue, K. & Urahama, K. 2006), they proposed another non-iterative 2DLDA called Parallel 2DLDA which computes L and R independently. Firstly, let us define the row-row within-class and between-class scatter matrix as follows:

$$S_w^r = \sum_{j=1}^{C} \sum_{X_i \in \Pi_j} (X_i - M_j)(X_i - M_j)^T \tag{48}$$

$$S_b^r = \sum_{j=1}^{C} N_j (M_j - M)(M_j - M)^T \tag{49}$$

The optimal left side transformation matrix L would maximize $tr(L^T S_b^r L)/tr(L^T S_w^r L)$. This optimization problem is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\max_{L} \quad tr(L^T S_b^r L) \\ &s.t. \quad L^T S_w^r L = I_1 \end{aligned} \tag{50}$$

Let $S_w^r = U\Lambda U^T$ be the eigen-decomposition of S_w^r, where Λ is a diagonal matrix whose diagonal elements are eigenvalues of S_w^r and U is an orthonormal matrix whose columns are the corresponding eigenvectors. Substitution of $\tilde{L} = \Lambda^{1/2} U^T L$ into (50) gives

$$\begin{aligned} &\max_{\tilde{L}} \quad tr(\tilde{L}^T \Lambda^{-1/2} U^T S_b^r U \Lambda^{-1/2} \tilde{L}) \\ &s.t. \quad \tilde{L}^T \tilde{L} = I_1 \end{aligned} \tag{51}$$

Compute l_1 eigenvectors $\{\tilde{\Phi}_i\}_{i=1}^{l_1}$ of the matrix $\Lambda^{-1/2} U^T S_b^r U \Lambda^{-1/2}$ and form the optimal solution of (50) as $L = U\Lambda^{-1/2}\tilde{L}$ where $\tilde{L} = [\tilde{\Phi}_1..\tilde{\Phi}_{l_1}]$. Alternatively, we define the column-column within-class and between-class scatter matrix as follows:

$$S_w^c = \sum_{j=1}^{C} \sum_{X_i \in \Pi_j} (X_i - M_j)^T (X_i - M_j) \tag{52}$$

$$S_b^c = \sum_{j=1}^{C} N_j (M_j - M)^T (M_j - M) \tag{53}$$

The optimal left side transformation matrix R would maximize $tr(R^T S_b^c R) / tr(R^T S_w^c R)$. This optimization problem is equivalent to the following constrained optimization problem:

$$\begin{aligned} &\max_R \quad tr(R^T S_b^c R) \\ &s.t. \quad R^T S_w^c R = I_2 \end{aligned} \tag{54}$$

Let $S_w^c = V \Lambda V^T$ be the eigen-decomposition of S_w^c , where Λ is a diagonal matrix whose diagonal elements are eigenvalues of S_w^c and V is an orthonormal matrix whose columns are the corresponding eigenvectors. Substitution of $\tilde{R} = \Lambda^{1/2} V^T R$ into (54) gives

$$\begin{aligned} &\max_{\tilde{R}} \quad tr(\tilde{R}^T \Lambda^{-1/2} V^T S_b^c V \Lambda^{-1/2} \tilde{R}) \\ &s.t. \quad \tilde{R}^T \tilde{R} = I_2 \end{aligned} \tag{55}$$

Compute l_2 eigenvectors $\{\tilde{\Psi}_i\}_{i=1}^{l_2}$ of the matrix $\Lambda^{-1/2} V^T S_b^c V \Lambda^{-1/2}$ and form the optimal solution of (54) as $R = V \Lambda^{-1/2} \tilde{R}$ where $\tilde{R} = [\tilde{\Psi}_1 .. \tilde{\Psi}_{l_2}]$. The parallel 2DLDA can be described as follow

Algorithm – Parallel 2DLDA

Step A1

Compute S_w^r and S_b^r

Step A2

Compute eigen-decomposition $S_w^r = U \Lambda U^T$

Step A3

Compute the first l_1 eigenvectors $\{\tilde{\Phi}_i\}_{i=1}^{l_1}$ of the matrix $\Lambda^{-1/2} U^T S_b^r U \Lambda^{-1/2}$ and compute $L = U \Lambda^{-1/2} \tilde{L}$ where $\tilde{L} = [\tilde{\Phi}_1 .. \tilde{\Phi}_{l_1}]$

Step B1

Compute S_w^c and S_b^c

Step B2

Compute eigen-decomposition $S_w^c = V \Lambda V^T$

Step B3

Compute the first l_2 eigenvectors $\{\tilde{\Psi}_i\}_{i=1}^{l_2}$ of the matrix $\Lambda^{-1/2} V^T S_b^c V \Lambda^{-1/2}$ and compute $R = V \Lambda^{-1/2} \tilde{R}$ where $\tilde{R} = [\tilde{\Psi}_1 .. \tilde{\Psi}_{l_2}]$

Since the algorithm computes L and R independently, we can interchange Step A1,A2,A3,A4 and Step B1,B2,B3,B4.

5. Conclusions

In this chapter, we have shown the class of low-rank approximation algorithms based directly on image data. In general, those algorithms are reduced to a couple of eigenvalue problems of row-row and column-column covariance matrices. In contrast to those 1D approaches, the size of the image covariance matrix using image-based approaches is much smaller. As a result, it is easier to evaluate the covariance matrix accurately and less time is required to determine the corresponding eigenvectors. Some future work should be considered such as the relationship between 1D approaches and 2D approaches and an extension of those 2D approaches to higher tensors.

6. References

Belhumeur, P.N.; Hespanha, J.P. & Kriegman, D.J. (1997). Eigenfaces vs. Fisherfaces: recognition using class specific linear projection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* Vol. 19, Issue 7, July 1997, pp. 711 - 720, ISSN 0162-8828

Chris Ding & Jieping Ye (2005). Two-dimensional Singular Value Decomposition (2DSVD) for 2D Maps and Images. *Proc. SIAM Int'l Conf. Data Mining (SDM'05),* pp. 32-43

D. Xu; S. Yan; L. Zhang; H. Zhang; Z. Liu & H. Shum (2005). Concurrent Subspace Analysis. *IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR), 2005*

Inoue, K. & Urahama, K. (2006). Non-Iterative Two-Dimensional Linear Discriminant Analysis. *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on.* Vol. 2, pp. 540- 543

Jian Yang; Zhang, D.; Frangi, A.F. & Jing-yu Yang (2004). Two-dimensional PCA: a new approach to appearance-based face representation and recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* Vol. 26, Issue 1, Jan 2004, pp. 131- 137, ISSN 0162-8828

Jieping Ye (2004). Generalized low rank approximations of matrices. *Proceedings of the twenty-first international conference on Machine learning,* pp. 112, ISSN 1-58113-828-5

Jieping Ye; Ravi Janardan & Qi Li (2004). Two-Dimensional Linear Discriminant Analysis. *Neural Information Processing Systems 2004*

Jun Liu & Songcan Chen (2006). Non-iterative generalized low rank approximation of matrices. *Pattern recognition letters,* Vol. 27, No. 9, 2006, pp 1002-1008

M. Turk & A. Pentland (1991). Face recognition using eigenfaces. *Proc. IEEE Conference on Computer Vision and Pattern Recognition,* pp 586–591

Ming Li & Baozong Yuan (2004). A novel statistical linear discriminant analysis for image matrix: two-dimensional Fisherfaces. *Signal Processing, 2004. Proceedings. ICSP '04. 2004 7th International Conference on*

Songcan Chen; Yulian Zhu; Daoqiang Zhang; Yang Jing-Yu (2005). Feature extraction approaches based on matrix pattern : MatPCA and MatFLDA. *Pattern recognition letters,* Vol. 26, No. 8, 2005, pp. 1157-1167

Zhizheng Liang; David Zhang & Pengfei Shi (2007). The theoretical analysis of GLRAM and its applications. *Pattern recognition,* Vol. 40, Issue 3, 2007, pp 1032-1041