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

3,800 Open access books available
116,000 International authors and editors
120M Downloads

154 Countries delivered to
TOP 1% Our authors are among the most cited scientists
12.2% Contributors from top 500 universities

Interested in publishing with us?
Contact book.department@intechopen.com

Numbers displayed above are based on latest data collected. For more information visit www.intechopen.com
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 2DLD (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.
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.

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

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). Table 1 gives an summary of those algorithms presented. Basis notations used in this chapter are summarized in Table 2.

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

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 $y_i \in \mathbb{R}^m$ are defined by the following linear transformation:
where \( k = 1, 2, \ldots, N \), \( \mu \in \mathbb{R}^n \) is the mean of all samples, and \( W \in \mathbb{R}^{n \times m} \) is a matrix with orthonormal columns. After the linear transformation, each data point \( x_k \) can be represented by a feature vector \( y_k \in \mathbb{R}^m \) 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:

\[
W_{opt} = \arg \max_W \| W^T S_W W \|
\]

with the total scatter matrix is defined as

\[
S = \frac{1}{N} \sum_{i=1}^{N} (x_i - \mu)(x_i - \mu)^T
\]

and \( \mu = \frac{1}{N} \sum_{i=1}^{N} x_i \) is the mean of all samples. The optimal projection \( W_{opt} = [w_1, w_2, \ldots, w_m] \) is the set of \( n \)-dimensional eigenvectors of \( S \) 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 \( \{\Pi_1, \Pi_2, \ldots, \Pi_C\} \). Let \( N_i \) be the number of the samples in class \( \Pi_i \) \((i = 1, 2, \ldots, C)\), \( \mu_i = \frac{1}{N_i} \sum_{x \in \Pi_i} x \) be the mean of the samples in class \( \Pi_i \). Then the between-class scatter matrix \( S_b \) is defined as

\[
S_b = \frac{1}{N} \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T
\]

and the within-class scatter matrix \( S_w \) is defined as

\[
S_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{x \in \Pi_i} (x_i - \mu_i)(x_i - \mu_i)^T
\]

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.,
$W_{opt} = \arg \max_{W} \frac{W^T S_{W} W}{W^T S_{b} W} = [w_1 w_2 ... w_m]$ \hfill (6)

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

$$S_{w_i} w_i = \lambda_i S_{w} w_i \quad i = 1, 2, ..., m$$ \hfill (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 categorize those image-based approaches into two main categories 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 $X_i \in \mathbb{R}^{r \times c}$, which is a matrix space of size $r \times c$ . The total scatter matrix is defined as

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

with $M = \frac{1}{N} \sum_{i=1}^{N} X_i \in \mathbb{R}^{r \times c}$ is the mean image of all samples. $G_i \in \mathbb{R}^{r \times c}$ 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 $Y_i \in \mathbb{R}^{r \times m}$ are defined by the following linear transformation:

$$Y_i = (X_i - M)W \in \mathbb{R}^{r \times m}$$ \hfill (9)

where $i = 1, 2, ..., N$ and $W \in \mathbb{R}^{r \times m}$ is a matrix with orthogonal columns. In 2DPCA, the projection $W_{opt}$ is chosen to maximize $tr(W^T G_i W)$ . The optimal projection $W_{opt} = [w_1 w_2 ... w_m]$
with \( \{w_i\} = 1, 2, \ldots, m \) is the set of \( c \)-dimensional eigenvectors of \( G \), 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

\[
S_b = \frac{1}{N} \sum_{i=1}^{c} N_i (M_i - M)^T (M_i - M)
\]

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

\[
S_w = \frac{1}{N} \sum_{i=1}^{c} \sum_{X \in \Xi_i} (X - M_i)^T (X - M_i)
\]

with \( m \in \mathbb{R}^m \) is the mean image of all samples and \( M = \frac{1}{N} \sum_{i=1}^{N} X_i \in \mathbb{R}^{m \times r} \) be the mean of the samples in class \( \Pi_i \). 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 \( Y_i \in \mathbb{R}^{m \times r} \) are defined by the following linear transformation:

\[
Y_i = (X_i - M)W \in \mathbb{R}^{m \times r}
\]

where \( i = 1, 2, \ldots, N \) and \( W \in \mathbb{R}^{c \times m} \) is a matrix with orthogonal columns. And the projection \( W_{opt} \) is chosen with the criterium 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 \( G_{opt} \) exists, i.e. \( \text{rank}(G_{opt}) = c \). We have,

\[
\text{rank}(G_w) = \text{rank}
\left( \frac{1}{N} \sum_{i=1}^{c} \sum_{X \in \Xi_i} (X_i - M_i)^T (X_i - M_i) \right)
\leq (N - C) \times \min(r, c)
\]

The inequality in (13) holds because \( \text{rank}(X) = \min(r, c) \). So, in 2DLDA, \( G_w \) is nonsingular when

\[
c \leq (N - C) \times \min(r, c)
\]

\[
\Rightarrow N \geq C + \frac{c}{\min(r, c)}
\]

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:

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

(15)

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

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

$$\min_{L,R} \sum_{i=1}^{N} \|Y_i - LY_iR^T\|^2$$

s.t. $L^TL = I, R^TR = I$.

(16)

where $L \in \mathbb{R}^{m_i \times r}, R \in \mathbb{R}^{n_i \times c}$, $Y_i \in \mathbb{R}^{m_i \times n_i}$ for $i = 1, \ldots, N$, $I_r \in \mathbb{R}^{r \times r}$ and $I_c \in \mathbb{R}^{c \times c}$ are identity matrices, where $l_r \leq r$ and $l_c \leq c$. 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 $(Y_i)^\dagger$ be the optimal solution to the minimization problem in Eq. (16). Then $Y_i = L^TX_iR$ for every $i$.

**Proof:** By the property of the trace of matrices,

$$\sum_{i=1}^{N} \text{tr} (Y_i - LY_iR^T)^2 = \sum_{i=1}^{N} \text{tr} ((X_i - LY_iR^T)(X_i - LY_iR^T)^T)$$

$$= \sum_{i=1}^{N} \text{tr} (X_iX_i^T) + \sum_{i=1}^{N} \text{tr} (Y_iY_i^T) - 2 \sum_{i=1}^{N} \text{tr} (LY_iR^TX_i^T)$$

(17)

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

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

(18)
By taking derivatives of (18), and force it equal to zero
\[
\frac{\partial E}{\partial Y} = 2Y' - 2R'X_L'^T L = 0
\]  
(19)
we obtain \( Y_i = U_i'X_i'R \). This completes the proof of the theorem.

**Theorem 2.** Let \( L, R \) and \( Y_1, \ldots, Y_N \) be the optimal solution to the minimization problem in Eq. (16). Then \( L, R \) solve the following optimization problem:
\[
\max \sum_{i=1}^{N} \|U_i'X_i'R\|
\text{s.t.} \quad U_i'L = I, R^T R = I
\]  
(20)

**Proof:** From Theorem 1, \( Y_i = U_i'X_i'R \) for every \( i \), we obtain
\[
\sum_{i=1}^{N} tr(Y_i'Y_i) = 2 \sum_{i=1}^{N} tr(LY_i'R_i'X_i')
\]
\[
= \sum_{i=1}^{N} tr(U_i'X_i'R_i'X_i'L) + 2 \sum_{i=1}^{N} tr(LU_i'X_i'R_i'X_i')
\]
\[
= -\sum_{i=1}^{N} tr(U_i'X_i'R_i'X_i'L) = -\sum_{i=1}^{N} \|U_i'X_i'R\|^2
\]  
(21)

Hence the minimization problem in Eq. (16) is equivalent to the maximization of
\[
\max \sum_{i=1}^{N} \|U_i'X_i'R\|
\text{s.t.} \quad U_i'L = I, R^T R = I
\]  
(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 algorithm for the computation of \( L, R \), is stated in the following theorem:

**Theorem 3.** Let \( L, R \) and \( Y_1, \ldots, Y_N \) be the optimal solution to the minimization problem in Eq. (16). Then,
(1) For a given \( R \), \( L \) consists of the \( l \) eigenvectors of the matrix
\[
S_l = \sum_{i=1}^{N} X_i'R_i'X_i'
\]  
(23)
corresponding to the largest \( l \) eigenvalues.
(2) For a given \( L \), \( R \) consists of the \( l \) eigenvectors of the matrix
\[
S_r = \sum_{i=1}^{N} X_i'L_i'X_i'
\]  
(24)
corresponding to the largest \( l \) eigenvalues.
Proof: From the Theorem 2., the objective function in (22) can be re-written as

\[
\sum_{i=1}^{N} T_i^X R_i^X \left( T_i^X R_i^X \right)^T = \text{tr} \left( A_i^X \right) \text{tr} \left( A_i^X \right)^T \]

where \( S_i = \sum_{j=1}^{N} R_j^X T_j^X \). Hence for a given \( R, L \in \mathbb{R}^{n \times i} \) consists of the \( l_i \) eigenvectors of the matrix \( S_i \) corresponding to the largest \( l_i \) eigenvalues. Similarly, for a given \( L, R \in \mathbb{R}^{n \times i} \) consists of the \( l_i \) eigenvectors of the matrix \( S_k = \sum_{j=1}^{N} L_j^X R_j^X \) corresponding to the largest \( l_i \) eigenvalues. This completes the proof of the theorem. An iterative procedure for computing \( L \) and \( R \) can be presented as follow:

Algorithm – GLRAM

**Step 0**
Initialize \( L = [I, 0]^T \), and set \( k = 0 \).

**Step 1**
Compute \( l_i \) eigenvectors \( \{ \Phi_{l_i}^{(1)}, \ldots, \Phi_{l_i}^{(k)} \} \) of the matrix \( S_k = \sum_{j=1}^{N} X_j^T L_j^T L_j X_j \) corresponding to the largest \( l_i \) eigenvalues and form \( R^{(k+1)} = [\Phi_{l_i}^{(1)}, \ldots, \Phi_{l_i}^{(k)}] \).

**Step 2**
Compute \( l_i \) eigenvectors \( \{ \Phi_{l_i}^{(1)}, \ldots, \Phi_{l_i}^{(k)} \} \) of the matrix \( S_k = \sum_{j=1}^{N} X_j^T R_j^T R_j X_j \) corresponding to the largest \( l_i \) eigenvalues and form \( L^{(k+1)} = [\Phi_{l_i}^{(1)}, \ldots, \Phi_{l_i}^{(k)}] \).

**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 \( Y_i = L_i^T X \) for \( i = 1 \ldots N \).

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 satisfies \( H^T H = I \in \mathbb{R}^{h \times h} \). Then, for \( i = 1, \ldots, m \), we have

\[
\lambda_{i_{\text{min}}}(B) \leq \lambda_i(H^T BH) \leq \lambda_i(B)
\]
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 $\lambda_i$ of $B$ and $H$ be an $m \times h$ which satisfies $H^TH = I \in \mathbb{R}^{h \times h}$. Then,

$$\lambda_{i+1} + \ldots + \lambda_h \leq tr(H^TBH) \leq \lambda_1 + \ldots + \lambda_h$$  \hfill (27)

and the second equality holds if $H = WQ$ where $W = [w_1, \ldots, w_h]$ and $Q$ is any $h \times h$ orthogonal matrix.

Some following matrices are defined (Zhizheng Liang et al., 2007)

$$G_1 = \sum_{i=1}^{N} X_i^T X_i$$  \hfill (28)

$$G_2 = \sum_{j=1}^{N} X_j X_j^T$$  \hfill (29)

Let $F_i$ consists of the eigenvectors of $G_i$ corresponding to the first $l_i$ largest eigenvalues and $F_i$ consists of the eigenvectors of $G_i$ corresponding to the first $l_1$ largest eigenvalues. Next, we define

$$H_{KL} = \sum_{i=1}^{N} X_i F_i F_i^T X_i^T$$  \hfill (30)

$$H_{KL} = \sum_{j=1}^{N} X_j F_j F_j^T X_j$$  \hfill (31)

Let $K_{Li}$ consists of the eigenvectors of $H_{Li}$ corresponding to the first $l_i$ largest eigenvalues and $K_{Li}$ consists of the eigenvectors of $H_{Li}$ corresponding to the first $l_1$ 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 $H_{Li}$ and $d_2$ be the sum of the first $l_2$ largest eigenvalues of $H_{Li}$. 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

$$\sum_{i=1}^{N} [U^TX_i][U^TX_i]^T = \sum_{i=1}^{N} [U^T X_i R \bar{R} \bar{X}_i L] = [U^T \sum_{i=1}^{N} X_i R \bar{R} \bar{X}_i ] L = [U^T S_i L]$$  \hfill (32)

Applying Corollary 1. we have

$$tr(U^T S_i L) \leq tr(S_i)$$  \hfill (33)
Since
\[ \text{tr}(S_i) = \text{tr}\left( \sum_{j=1}^{N} X_j R X_j^T \right) = \text{tr}(R^T G_i R) \leq \text{tr}(G_i) \]  (34)

From Eq. (33) and Eq. (34), we can obtain
\[ \text{tr}(U^T S_i L) \leq \text{tr}(G_i) \]  (35)

Then it is not difficult to obtain \( R = F Q_{s,i}^T \), where \( Q_{s,i}^T \) is any orthogonal matrix. Substitute \( R = F Q_{s,i}^T \) into \( S_i \) and obtain \( H_{s,i} \), we can have \( L = K Q_{s,i}^T \). Furthermore, it is straightforward to verify that the value of Eq. (22) is equal to \( d_i \).

(b) In the same way we can have
\[ \sum_{i=1}^{N} U^T X_i R X_i^T = \sum_{i=1}^{N} \text{tr}(U^T X_i R X_i^T) = \sum_{i=1}^{N} \text{tr}(R^T X_i^T L U X_i R) \]
\[ = \text{tr}\left( R^T \left( \sum_{i=1}^{N} X_i^T L U X_i \right) \right) = \text{tr}(R^T S_i R) \]  (36)

Applying Corollary 1, we have
\[ \text{tr}(R^T S_i R) \leq \text{tr}(S_i) \]  (37)

Since
\[ \text{tr}(S_i) = \text{tr}\left( \sum_{i=1}^{N} X_i^T L U X_i \right) = \text{tr}(U^T G_i L) \leq \text{tr}(G_i) \]  (38)

From Eq. (37) and Eq. (38), we can obtain
\[ \text{tr}(R^T S_i R) \leq \text{tr}(G_i) \]  (39)

Then it is not difficult to obtain \( L = F Q_{s,i}^T \), where \( Q_{s,i}^T \) is any orthogonal matrix. Substitute \( L = F Q_{s,i}^T \) into \( S_i \) and obtain \( H_{s,i} \), we can have \( R = K Q_{s,i}^T \). Furthermore, it is straightforward to verify that the value of Eq. (22) is equal to \( d_i \). From (a) and (b), the theorem is proven. From this proof, it is not difficult to derive the non-iterative GLRAM as follows:

**Algorithm – Non-iterative GLRAM**

**Step 1**
Compute the matrices \( G_i \) and \( G_s \).

**Step 2**
Compute eigenvectors of the matrices \( G_i \) and \( G_s \), let \( R = F Q_{s,i}^T \) and \( L = F Q_{s,i}^T \).

**Step 3**
Compute eigenvectors of the matrices \( H_{s,i} \) and \( H_{s,r} \), and obtain \( L = K Q_{s,i}^T \).
corresponding to $R$ in step 2 and $R = KQ_{x0}^T$, 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 $Y = E^T X R$

### 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:

$$D_w = \sum_{j=1}^{C} \sum_{x \in C_j} \|X_j - M\|^2$$

$$= tr \left( \sum_{j=1}^{C} \sum_{x \in C_j} (X_j - M_j)(X_j - M_j)^T \right)$$

$$D_b = \sum_{j=1}^{C} \sum_{x \in C_j} N_j \|M_j - M\|^2$$

$$= tr \left( \sum_{j=1}^{C} \sum_{x \in C_j} N_j (M_j - M)(M_j - M)^T \right)$$

(40)

(41)

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

$$\tilde{D}_w = tr \left( \sum_{j=1}^{C} \sum_{x \in C_j} E^T (X_j - M_j)R R^T (X_j - M_j)^T L \right)$$

(42)

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

(43)

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

$$S^w = \sum_{j=1}^{C} \sum_{x \in C_j} (X_j - M_j)R R^T (X_j - M_j)^T$$

$$S^b = \sum_{j=1}^{C} \sum_{x \in C_j} N_j (M_j - M)R R^T (M_j - M)^T$$

(44)

(45)
After defining those matrices we can derive the iterative 2DLDA algorithm as follows:

**Algorithm – Iterative 2DLDA**

**Step 0**
Initialize \( R = R^{(0)} = [I_x, 0]^T \) and set \( k = 0 \).

**Step 1**
Compute
\[
S_r^{(k)} = \sum_{x \in T_k} (X_i - M_j)^T L L^T (X_i - M_j)
\]
\[
S_c^{(k)} = \sum_{j \in T_k} N_j (M_j - M)^T L L^T (M_j - M)
\]

**Step 2**
Compute \( l \) eigenvectors \( \{\Phi_i^{(l,k)}\}_{i=1}^l \) of the matrix \( S_r^{(k)} \) and form \( L^{(k)} = [\Phi_1^{(l,k)}, \Phi_2^{(l,k)}, \ldots, \Phi_l^{(l,k)}] \).

**Step 3**
Compute
\[
S_r^{(l,k)} = \sum_{x \in T_k} (X_i - M_j)^T L^{(k)} L^{(k)T} (X_i - M_j)
\]
\[
S_c^{(l,k)} = \sum_{j \in T_k} N_j (M_j - M)^T L^{(k)} L^{(k)T} (M_j - M)
\]

**Step 4**
Compute \( l \) eigenvectors \( \{\Phi_i^{(l,k)}\}_{i=1}^l \) of the matrix \( S_c^{(l,k)} \) and form \( R^{(k)} = [\Phi_1^{(l,k)}, \Phi_2^{(l,k)}, \ldots, \Phi_l^{(l,k)}] \).

**Step 5**
If \( L^{(k)} \), \( R^{(k)} \) are not convergent then set increase \( k \) by 1 and go to Step 1; otherwise proceed to Step 6.

**Step 6**
Let \( L' = L^{(k)} \), \( R' = R^{(k)} \) and compute \( Y_i = L'^T X R' \) for \( i = 1..N \).

### 4.4 Non-iterative 2DLDA

Iterative 2DLDA computes \( L \) and \( R \) in turn with the initialization \( R = R^{(0)} = [I_x, 0]^T \). Alternatively, we can consider another algorithm that computes \( L \) and \( R \) in turn with the initialization \( L = L^{(0)} = [I_x, 0]^T \). By unifying them, in this subsection, we can select \( L \) and
which give larger $F(L, R)$ and form the selective algorithm as follow (Inoue, K. & Urahama, K. 2006)

**Algorithm – Selective 2DLDA**

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

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

**Step 3**
If $f(L^{(0)}, R^{(0)}) \geq f(L^{(2)}, R^{(2)})$ then output $L = L^{(0)}$ and $R = R^{(0)}$, 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:

$$S'_r = \sum_{j=1}^{C} \sum_{i \in \Pi_j} (X_j - M_j)(X_j - M_j)^T$$  \hspace{1cm} (48)

$$S'_c = \sum_{j=1}^{C} N_j (M_j - M)(M_j - M)^T$$  \hspace{1cm} (49)

The optimal left side transformation matrix $L$ would maximize $\text{tr}(L^T S'_r L) / \text{tr}(L^T S'_c L)$. This optimization problem is equivalent to the following constrained optimization problem:

$$\max_{L} \frac{\text{tr}(L^T S'_r L)}{\text{tr}(L^T S'_c L)} \quad \text{s.t.} \quad L^T S'_c L = I_i$$  \hspace{1cm} (50)

Let $S'_c = U \Lambda U^T$ be the eigen-decomposition of $S'_c$, where $\Lambda$ is a diagonal matrix whose diagonal elements are eigenvalues of $S'_c$ and $U$ is an orthonormal matrix whose columns are the corresponding eigenvectors. Substitution of $L = U \Lambda^{-1/2}$ into (50) gives

$$\max_{L} \frac{\text{tr}(U \Lambda^{-1/2} L^T S'_c L U \Lambda^{-1/2} L)}{\text{tr}(U \Lambda^{-1/2} L^T S'_c L U \Lambda^{-1/2} L)} \quad \text{s.t.} \quad U L = I_i$$  \hspace{1cm} (51)

Compute $l_1$ eigenvectors $\{\Phi_{i}\}_{i=1}^{l_1}$ of the matrix $A^{-1/2} U^T S'_c U A^{-1/2}$ and form the optimal solution of (50) as $L = U \Lambda^{-1/2}$, where $L = [\Phi_1, \Phi_2, \ldots, \Phi_{l_1}]$. Alternatively, we define the column-column within-class and between-class scatter matrix as follows:

$$S'_l = \sum_{j=1}^{C} \sum_{i \in \Pi_j} (X_j - M_j)^T (X_j - M_j)$$  \hspace{1cm} (52)
The optimal left side transformation matrix \( R \) would maximize \( \frac{\text{tr}(R^T S_b R)}{\text{tr}(R^T S_w R)} \).
This optimization problem is equivalent to the following constrained optimization problem:

\[
\max_x \ \text{tr}(R^T S_b R) \\
\text{s.t.} \ \ R^T S_w R = I_s
\]

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

\[
\max_x \ \text{tr}(R^T \Lambda^{1/2} V^T S_b V \Lambda^{1/2} R) \\
\text{s.t.} \ \ R^T \Lambda R = I_s
\]

Compute \( i_l \) eigenvectors \( \{ \Psi_l \} \) of the matrix \( \Lambda^{1/2} V S_b V \Lambda^{1/2} \) and form the optimal solution of (54) as \( R = V \Lambda^{1/2} \tilde{R} \) where \( \tilde{R} = [\Psi_1, \Psi_{i_l}] \). The parallel 2DLDA can be described as follows.

**Algorithm – Parallel 2DLDA**

**Step A1**
- Compute \( S'_c \) and \( S'_w \)

**Step A2**
- Compute eigen-decomposition \( S'_w = U \Lambda U^T \)

**Step A3**
- Compute the first \( i_l \) eigenvectors \( \{ \Phi_l \} \) of the matrix \( \Lambda^{1/2} U S'_w U \Lambda^{1/2} \) and compute \( L = U \Lambda^{1/2} \tilde{L} \) where \( \tilde{L} = [\Phi_1, \Phi_{i_l}] \)

**Step B1**
- Compute \( S'_c \) and \( S'_s \)

**Step B2**
- Compute eigen-decomposition \( S'_s = V \Lambda V^T \)

**Step B3**
- Compute the first \( i_l \) eigenvectors \( \{ \Psi_l \} \) of the matrix \( \Lambda^{1/2} V S'_s V \Lambda^{1/2} \) and compute \( R = V \Lambda^{1/2} \tilde{R} \) where \( \tilde{R} = [\Psi_1, \Psi_{i_l}] \)

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, these 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


This book will serve as a handbook for students, researchers and practitioners in the area of automatic (computer) face recognition and inspire some future research ideas by identifying potential research directions. The book consists of 28 chapters, each focusing on a certain aspect of the problem. Within every chapter the reader will be given an overview of background information on the subject at hand and in many cases a description of the authors' original proposed solution. The chapters in this book are sorted alphabetically, according to the first author's surname. They should give the reader a general idea where the current research efforts are heading, both within the face recognition area itself and in interdisciplinary approaches.

How to reference
In order to correctly reference this scholarly work, feel free to copy and paste the following:
