

一种快速的零空间算法

卢桂馥^{1,2}, 王勇¹, 邹健^{1,2}

(1. 安徽工程大学计算机与信息学院, 241000, 安徽芜湖; 2. 南京理工大学计算机科学与技术学院, 210094, 南京)

摘要: 为了进一步提高零空间算法的运行效率, 提出了一种新的快速的零空间算法(FINBSA). FINBSA 不需要进行特征值分解或奇异值分解, 而只需一次正交三角(QR)分解就可以求得最佳投影矩阵, 使得 FINBSA 的算法复杂度比现有的零空间算法要低. 在 PIE 人脸库上的实验结果表明, FINBSA 的识别率与现有的零空间算法相同, 但是远比现有的零空间算法要高效, 尤其是在训练样本数较多时, FINBSA 的运行时间比现有零空间算法节省了 100% 以上.

关键词: 特征提取; 线性鉴别分析; 零空间线性鉴别分析; 正交三角分解; 人脸识别

中图分类号: TP391 **文献标志码:** A **文章编号:** 0253-987X(2012)02-0059-05

A Fast Implementation of Null Space Based Algorithm

LU Guifu^{1,2}, WANG Yong¹, ZOU Jian^{1,2}

(1. School of Computer Science and Information, Anhui Polytechnic University, Wuhu, Anhui 241000, China;

2. School of Computer Science and Technology, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: To improve the efficiency of null space based algorithm, a new fast implementation of null space based algorithm (FINBSA) is presented. FINBSA is carried out without any eigen-decomposition computing and singular value decomposition, by only one step of orthogonal-triangular(QR) decomposition to obtain the optimal projection matrix. Thus the computational complexity of this algorithm gets much lower than that of the other null space based ones. The experiments on PIE face database demonstrate that the recognition accuracy of FINBSA is equivalent to the other null space based algorithms, but FINBSA is more efficient especially for large size of training samples, and the running period of FINBSA is over 100% lower than that of the other existing null space based algorithms.

Keywords: feature extraction; linear discrimination analysis; null space based linear discrimination analysis; orthogonal-triangular decomposition; face recognition

在模式识别和机器学习中, 经常会遇到数据特征的高维问题, 亦即人们所说的维数灾难^[1]. 为了有效地进行数据分析, 降低数据特征维数就显得异常重要. 主分量分析(PCA)^[2]和基于 Fisher 准则线性鉴别分析(LDA)^[2]是应用最广泛的两个线性降维方法.

PCA 是一种无监督学习算法, PCA 方法通过最大化训练样本总体散布(即最小化样本重建误差)来获得样本的最优投影方法. 由于 PCA 并没有利用

类别信息, 使得以最优重建为目的的 PCA 方法不适合分类问题. LDA 作为一种有监督学习算法, 其原理是使得降维后同类样本之间的距离最小, 而异类样本之间的距离最大. LDA 虽然可以有效地提取各类之间的鉴别信息, 但由于在计算过程中需要保证类内散布矩阵可逆, 而在模式识别和机器学习中经常遇到高维小样本问题(如人脸识别), 类内散布矩阵往往是奇异的, 因此 LDA 算法很难直接计算.

收稿日期: 2011-07-02. 作者简介: 卢桂馥(1976—), 男, 博士生, 安徽工程大学副教授. 基金项目: 国家自然科学基金资助项目(60873151).

网络出版时间: 2011-12-20

网络出版地址: <http://www.cnki.net/kcms/detail/61.1069.T.20111220.1806.012.html>

Fisherface^[3]是一种最常见的解决小样本问题的算法,其基本原理是先利用 PCA 对样本进行降维,使样本的类内散布矩阵非奇异,然后再利用 LDA 获得鉴别投影空间.但是,在 PCA 降维的过程中有可能损失鉴别信息,且无法保证降维后的类内散布矩阵一定非奇异.直接 LDA(Direct LDA, DLDA)方法^[4]则首先将样本投影到类间散度矩阵的值域空间,然后在此空间内求解最佳鉴别矢量使类内散布最小.杨健等^[5-6]提出了直接基于图像矩阵的图像投影鉴别分析方法,这不仅可以克服小样本问题,也提高了特征抽取的速度.

Cheng 等^[7]证明了最佳鉴别矢量应位于类内散布矩阵的零空间内,并且提出了一种零空间算法,但是 Cheng 等的算法存在计算量大的问题.为了减少零空间算法的计算量, Huang 等^[8]提出了 PCA+零空间方法,此算法首先将所有样本映射到总体散布矩阵的值域空间内,然后在此空间内采用零空间方法进行计算.但是,PCA+零空间方法需进行多次奇异值分解(或特征值分解)才能求得最佳投影矩阵,与矩阵的 QR 分解相比,对矩阵进行奇异值分解(或特征值)的效率较低^[9].因此,PCA+零空间方法依然存在计算量较大的问题.为了进一步减少零空间算法的计算量,最近文献^[10]提出了一种新的零空间实现算法,利用两次 QR 分解来求得最佳投影矩阵,由于算法避免了奇异值分解(或特征值分解),使得算法的效率比 PCA+零空间算法有了进一步的提高.但是,由于文献^[10]算法需进行两次 QR 分解,当样本数较多时,算法计算效率仍然较低.

为了进一步提高零空间算法的效率,本文提出了一种快速的零空间算法(FINBSA).FINBSA 只需一次 QR 分解就可以求得最佳投影矩阵,使得计算效率比现有的零空间算法有了进一步的提高.理论分析和在 PIE 人脸库上的实验表明,FINBSA 的识别率与现有的零空间算法相同,但是远比现有的零空间算法要高效.

1 零空间算法

设 $\mathbf{X} = \{\mathbf{x}_j^i, j=1, 2, \dots, n_i; i=1, 2, \dots, k\} \in \mathbf{R}^{m \times n}$ 为训练样本集,其中 n_i 为第 i 类的样本数, k 为类别数, $n = kn_i$ 为总样本数, m 为样本的维数, x_j^i 表示第 i 类的第 j 个样本. LDA 算法的目标是找到一个最优投影矩阵 $\mathbf{G}$,使得

$$\mathbf{G} = \arg \max_{\mathbf{G}} \text{trace}((\mathbf{G}^T \mathbf{S}_w \mathbf{G})^{-1} \mathbf{G}^T \mathbf{S}_b \mathbf{G}) \quad (1)$$

式中: $\mathbf{S}_b$ 为类间散布矩阵; $\mathbf{S}_w$ 为类内散布矩阵. $\mathbf{S}_b$

和 $\mathbf{S}_w$ 分别定义为

$$\mathbf{S}_b = \sum_{i=1}^k n_i (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T \quad (2)$$

$$\mathbf{S}_w = \sum_{i=1}^k \sum_{j=1}^{n_i} (\mathbf{x}_j^i - \mathbf{m}_i)(\mathbf{x}_j^i - \mathbf{m}_i)^T \quad (3)$$

式中: $\mathbf{m}_i = (1/n_i) \sum_{j=1}^{n_i} \mathbf{x}_j^i$ 为第 i 类的均值; $\mathbf{m} = (1/n) \sum_{i=1}^k \sum_{j=1}^{n_i} \mathbf{x}_j^i$ 为所有样本的均值. 最优投影矩阵 $\mathbf{G}$

的每一列即为矩阵 $\mathbf{S}_w^{-1} \mathbf{S}_b$ 的特征向量. 但是,当把 LDA 算法应用于人脸识别时,将面临小样本问题.此时,类内散布矩阵 $\mathbf{S}_w$ 是奇异的,也就是说我们没法求解 $\mathbf{S}_w$ 的逆矩阵 $\mathbf{S}_w^{-1}$.

为了解决 LDA 算法所面临的小样本问题,人们提出了许多解决方案,其中零空间算法^[7-8,10]是重要的一种方法.零空间算法的目标函数为

$$\mathbf{G} = \arg \max_{\mathbf{G}^T \mathbf{S}_w \mathbf{G} = \mathbf{0}, \mathbf{G}^T \mathbf{G} = \mathbf{I}} \text{trace}(\mathbf{G}^T \mathbf{S}_b \mathbf{G}) \quad (4)$$

式中: $\mathbf{I}$ 为单位阵. 文献^[7-8,10]分别提出了不同的算法来求解式(4),这些零空间算法的识别率相同,文献^[10]算法的求解效率最高,它的计算过程如算法 1 所示.

算法 1: 文献^[10]Chu 等的零空间算法.

输入: 数据矩阵 $\mathbf{X}$.

输出: 最佳投影矩阵 $\mathbf{G}$.

(1) 计算 $\mathbf{W}_i$ 和 $\mathbf{W}$, 其中 $\mathbf{W}_i (i=1, \dots, k)$ 和 $\mathbf{W}$ 分别是向量 $\mathbf{e}^{(i)} = [1, \dots, 1]^T \in \mathbf{R}^{n_i \times 1} (i=1, \dots, k)$ 和 $[(n_1)^{1/2}, (n_2)^{1/2}, \dots, (n_k)^{1/2}]^T \in \mathbf{R}^{k \times 1}$ 的 Householder 变换矩阵.

(2) 计算数据矩阵 $\mathbf{X}$ 的瘦 QR 分解 $\mathbf{X} = \mathbf{U}_1 \mathbf{R}$, 其中 $\mathbf{U}_1 \in \mathbf{R}^{m \times n}$, $\mathbf{R} \in \mathbf{R}^{n \times n}$.

(3) 利用

$$[\mathbf{R}_1 \ \mathbf{R}_2 \ \mathbf{R}_3] = \mathbf{R} \begin{bmatrix} \mathbf{W}_1 & & \\ & \ddots & \\ & & \mathbf{W}_k \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{W} & \\ & \mathbf{I} \end{bmatrix}$$

计算 $\mathbf{R}_2$ 和 $\mathbf{R}_3$, 其中 $\mathbf{R}_1 \in \mathbf{R}^{n \times 1}$, $\mathbf{R}_2 \in \mathbf{R}^{n \times (k-1)}$, $\mathbf{R}_3 \in \mathbf{R}^{n \times (n-k)}$. $\mathbf{P}$ 为置换矩阵, 可通过对一个 $n \times n$ 的单位阵交换其第 i 列和 $\sum_{j=1}^{i-1} n_j + 1$ 列得到, 其中 $i = 2, \dots, k$.

(4) 计算 $[\mathbf{R}_3 \ \mathbf{R}_2]$ 的瘦 QR 分解 $[\mathbf{R}_3 \ \mathbf{R}_2] = [\mathbf{Q}_1 \ \mathbf{Q}_2] \begin{bmatrix} \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{R}_{23} \end{bmatrix}$, 其中 $\mathbf{Q}_1 \in \mathbf{R}^{n \times (n-k)}$, $\mathbf{Q}_2 \in \mathbf{R}^{n \times (k-1)}$.

(5) $G=U_1Q_2$.

由算法 1 可知,在文献[10]算法中,最费时的步骤为第 2 步 QR 分解、第 4 步 QR 分解和第 5 步矩阵相乘.

2 快速的零空间算法

我们提出一种新的快速的零空间算法(FINB-SA),只需进行一次 QR 分解就能求得最佳投影矩阵 G .

定理 1 设 $X \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} P \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} = [X_1 \ X_2 \ X_3]$,其中 $X_1 \in \mathbf{R}^{m \times 1}$, $X_2 \in \mathbf{R}^{m \times (k-1)}$, $X_3 \in \mathbf{R}^{m \times (n-k)}$,则 $[X_3 \ X_2]$ 的瘦 QR 分解为

$[X_3 \ X_2] = [G_1 \ G_2] \begin{bmatrix} R_{12} & R_{13} \\ 0 & R_{23} \end{bmatrix}$ (5)

式中: $G_1=U_1Q_1 \in \mathbf{R}^{m \times (n-k)}$, $G_2=U_1Q_2 \in \mathbf{R}^{m \times (k-1)}$.

证明:由于 $X=U_1R$,并且

$X \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} P \begin{bmatrix} W & \\ & I \end{bmatrix} = [X_1 \ X_2 \ X_3]$

则有

$U_1R \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} P \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} = [X_1 \ X_2 \ X_3]$ (6)

由于

$[R_1 \ R_2 \ R_3] = R \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} P \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix}$

于是有

$U_1[R_1 \ R_2 \ R_3] = [X_1 \ X_2 \ X_3]$ (7)

故可得到

$U_1R_2 = X_2; \quad U_1R_3 = X_3$ (8)

由于

$[R_3 \ R_2] = [Q_1 \ Q_2] \begin{bmatrix} R_{12} & R_{13} \\ 0 & R_{23} \end{bmatrix}$

于是有

$U_1[R_3 \ R_2] = U_1[Q_1 \ Q_2] \begin{bmatrix} R_{12} & R_{13} \\ 0 & R_{23} \end{bmatrix}$ (9)

由式(8) 和式(9) ,可得

$[X_3 \ X_2] = [U_1Q_1U_1Q_2] \begin{bmatrix} R_{12} & R_{13} \\ 0 & R_{23} \end{bmatrix}$ (10)

证毕.

由定理 1 可知,我们只需对 $[X_3 \ X_2]$ 进行一次瘦 QR 分解,就可以得到最佳投影矩阵 $G=U_1Q_2$. 接下来考虑如何快速地求得 X_2 和 X_3 . 如果直接利用

$X \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} P \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix} = [X_1 \ X_2 \ X_3]$

来求 X_2 和 X_3 ,则此矩阵相乘的算法复杂度为 $O(mn^2)$,这样做比较费时. 我们可以利用 Householder 矩阵 W_i 、 W 以及 P 的特殊结构来减少计算量.

这里设

$Z_1 = X \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix}; Z_2 = Z_1P; Z_3 = Z_2 \begin{bmatrix} W & 0 \\ 0 & I \end{bmatrix}$

首先考虑如何快速地计算 Z_1 . W_i 可以表示为

$W_i = I_i - u_iu_i^T, \quad i = 1, \dots, k$ (11)

式中: I_i 为 $n_i \times n_i$ 的单位阵;

$u_i = \left(\begin{bmatrix} 1 - (n_i)^{1/2} \\ 1 \\ \vdots \\ 1 \end{bmatrix} / (n_i - n_i)^{1/2} \right) \in \mathbf{R}^{n_i \times 1}$

注意到

$X \begin{bmatrix} W_1 & & \\ & \ddots & \\ & & W_k \end{bmatrix} = [X(:, 1:n_1)W_1 \ \dots \ X(:, n_1 + \dots + n_{k-1} + 1:n_1 + \dots + n_k)W_k]$ (12)

式中: $X(:, 1:n_1)$ 指的是矩阵 X 的第 1 到 n_1 列的所有元素; $X(:, n_1 + \dots + n_{k-1} + 1:n_1 + \dots + n_k)$ 指的是矩阵 X 的第 $n_1 + \dots + n_{k-1} + 1$ 到 $n_1 + \dots + n_k$ 列的所有元素. 由式(11),有

$X(:, 1:n_1)W_1 = X(:, 1:n_1) - (X(:, 1:n_1)u_1)u_1^T$ (13)

这样计算 $X(:, 1:n_1)W_1$ 的复杂度是 $O(mn_1)$,易知计算 Z_1 的复杂度可以减为 $O(mn)$.

接下来考虑如何快速地计算 Z_2 . 由于 P 是置换矩阵,不需要对 Z_1 和 P 做矩阵乘法就能得到 Z_2 ,只需根据 P 来交换 Z_1 相应的列就可以得到 Z_2 . 最后,考虑如何快速地计算 Z_3 . W 可以表示为

$W = I - uu^T$ (14)

$u = \left(\begin{bmatrix} (n_1)^{1/2} - (n)^{1/2} \\ (n_2)^{1/2} \\ \vdots \\ (n_k) \end{bmatrix} / (n - (m_1)^{1/2})^{1/2} \right) \in \mathbf{R}^{k \times 1}$

注意到

$$\mathbf{Z}_2 \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = [\mathbf{Z}_2(:, 1:k) \mathbf{W} \quad \mathbf{Z}_2(:, k+1:n)] \tag{15}$$

由式(14),有

$$\mathbf{Z}_2(:, 1:k) \mathbf{W} = \mathbf{Z}_2(:, 1:k) - (\mathbf{Z}_2(:, 1:k) \mathbf{u}) \mathbf{u}^T \tag{16}$$

这样计算 $\mathbf{Z}_2(:, 1:k) \mathbf{W}$ 的复杂度为 $O(mk)$, 易知由 $\mathbf{Z}_2$ 计算 $\mathbf{Z}_3$ 的复杂度也为 $O(mk)$. 综上所述, 可以计算出

$$\mathbf{X} \begin{bmatrix} \mathbf{W}_1 & & \\ & \ddots & \\ & & \mathbf{W}_k \end{bmatrix} \mathbf{P} \begin{bmatrix} \mathbf{W} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} = [\mathbf{X}_1 \quad \mathbf{X}_2 \quad \mathbf{X}_3]$$

的复杂度减为 $O(mn)$.

本文提出的快速的零空间算法总结如下.
算法 2:快速的零空间算法(FINBSA).

- 输入:数据矩阵 $\mathbf{X}$.
输出:最佳投影矩阵 $\mathbf{G}$.
- (1)计算 $\mathbf{u}_1$ 和 $\mathbf{u}$.
 - (2)根据式(12)、式(13)计算 $\mathbf{Z}_1$.
 - (3)利用置换矩阵 $\mathbf{P}$ 计算 $\mathbf{Z}_2$ 和式(15),利用 $\mathbf{Z}_3$ 计算式(16).

(4) 计算 $[\mathbf{X}_3 \quad \mathbf{X}_2]$ 的瘦 QR 分解 $[\mathbf{X}_3 \quad \mathbf{X}_2] = [\mathbf{G}_1 \quad \mathbf{G}_2] \begin{bmatrix} \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{0} & \mathbf{R}_{23} \end{bmatrix}$.

(5) $\mathbf{G} = \mathbf{G}_2$.

为了比较文献[10]算法和 FINBSA 的效率,表 1 给出了两种算法每一步的计算复杂度比较.

表 1 文献[10]算法和 FINBSA 复杂度的比较

步骤	复杂度	
	文献[10]	FINBSA
1	$O(k)$	$O(k)$
2	$4mn^2 - 4n^3/3$	$O(mn)$
3	$O(n^2)$	$O(mk)$
4	$8n^3/3$	$4mn^2 - 4n^3/3$
5	$2mn(k-1)$	

从表 1 可以看出,本文提出的 FINBSA 比文献[10]算法要高效. 下面的实验部分也验证了我们的理论分析.

3 实验及分析

为了验证本文提出的 FINBSA 在人脸识别中的有效性,利用 PIE 人脸图像库进行了实验,在实验中使用的分类器是基于余弦距离的最近邻分类

器. 编程环境为 MATLAB 7.0, 操作系统为 Windows XP, 所用计算机的内存为 896 MB, CPU 为 Intel Pentium T1600 双核处理器(同时使用了双核的处理能力),主频为 1.67 GHz.

3.1 PIE 人脸数据库

PIE 人脸实验库由 68 人的 41 368 幅不同姿态、不同光照、不同表情的图像组成. 我们从中选择了包含姿态、光照、表情变化的 5 个子集(C05, C07, C09, C27, C29)共 68 人的正面人脸图像,每人 170 幅图像. 实验中,将图像处理成 64×64 的形式. 图 1 为 PIE 人脸数据库中某个人的 25 幅图像.



图 1 PIE 人脸库中的 25 幅图像

3.2 人脸识别性能比较

采用文献[11]方法,比较 FINBSA 和文献[10]算法所得到的投影向量是否相同. 设 $\mathbf{v}$ 、 $\mathbf{v}'$ 分别为 FINBSA 和文献[10]算法得到的单位投影向量,通过计算 $\mathbf{v}$ 和 $\mathbf{v}'$ 的点积 $\mathbf{v} \cdot \mathbf{v}'$ 来比较 $\mathbf{v}$ 和 $\mathbf{v}'$ 的相似性. $\mathbf{v} \cdot \mathbf{v}'$ 的值越大,说明 $\mathbf{v}$ 和 $\mathbf{v}'$ 越相近;当 $\mathbf{v} \cdot \mathbf{v}' = 1$ 时, $\mathbf{v}$ 和 $\mathbf{v}'$ 则完全相同.

从 PIE 人脸库中选择 5 幅图像作为训练样本,分别利用 FINBSA 和文献[10]算法来计算投影向量. 计算了所有 67 个投影向量的点积值,得到的所有投影向量的点积都为 1,也就是说这两种算法得到的投影向量是相同的,这也验证了前面的理论分析,即 FINBSA 和文献[10]算法是等价的.

随机在 PIE 库中选择 $i(i=5, 10, 20, 30)$ 幅图像作为训练样本,剩余的图像作为测试样本对二维 PCA(2DPCA)算法、文献[10]算法、FINBSA 的识别率进行了对比. 实验重复了 20 次,见表 2 各算法的平均识别率和标准方差.

表 2 在 PIE 人脸库中不同方法的识别率对比

训练样本数	平均识别率/%		
	2DPCA	文献[10]算法	FINBSA
340	60.2±1.0	74.3±1.0	74.3±1.0
680	72.8±0.7	84.6±1.0	84.6±1.0
1 360	83.4±0.4	89.9±0.5	89.9±0.5
2 040	87.4±0.5	92.2±0.3	92.2±0.3

从表 2 可以看出,FINBSA 和文献[10]算法比 2DPCA 算法的识别率要高,说明零空间算法可以较好地解决小样本问题. 另外, FINBSA 和文献[10]算法的识别率相同,这进一步验证了前面的理论分析,即 FINBSA 和文献[10]算法是等价的.

最后,比较 FINBSA 和文献[10]算法的运行时间. 表 3 记录了 FINBSA 和文献[10]算法在 PIE 库上运行 20 次的平均时间.

表 3 不同算法运行时间的比较

训练样本数	运行时间/s	
	文献[10]算法	FINBSA
340	2. 95	2. 42
680	9. 93	6. 81
1 360	45. 98	24. 61
2 040	135. 90	49. 22

从表 3 可以看出,FINBSA 的运行时间远比文献[10]算法要少,当样本数增加时,节省的时间更明显. 由于在文献[10]中,Chu 等证实了其提出的零空间算法与文献[7-8]提出的零空间算法等价,但计算效率要远高于文献[7-8]算法,因此我们可以得出结论:本文提出的零空间算法与文献[7-8,10]的算法等价,但计算效率要远高于这些零空间算法.

4 结束语

为了进一步提高零空间算法的效率,本文提出了一种快速的零空间算法(FINBSA). FINBSA 只需一次 QR 分解就可以求得最佳投影矩阵,使得计算效率比现有零空间算法有了进一步的提高. 理论分析和在 PIE 人脸库上的实验表明,FINBSA 的识别率与现有的零空间算法相同,但是远比现有的零空间算法要高效.

参考文献:

[1] RAUDYS S J, JAIN A K. Small sample size effects in statistical pattern recognition: recommendations for practitioners[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1991, 13(3): 252-264.

[2] MARTINEZ A M, KAK A C. PCA versus LDA[J].

IEEE Trans on Pattern Analysis and Machine Intelligence, 2001, 23(2): 228-233.

[3] BELHUMEOUR P N, HESPLANHA J P, KRIEGMAN D J. Eigenfaces vs. fisherfaces: recognition using class specific linear projection[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 1997, 19(7): 711-720.

[4] YU Hua, YANG Jie. A direct LDA algorithm for high-dimensional data with application to face recognition [J]. Pattern Recognition, 2001, 34(10): 2067-2070.

[5] YANG Jian, ZHANG D, FRANGI A F, et al. Two-dimensional PCA: a new approach to appearance-based face representation and recognition[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2004, 26(1): 131-137.

[6] 杨健, 杨静宇. 具有统计不相关性的图像投影鉴别分析及人脸识别[J]. 计算机研究与发展, 2003, 40(3): 447-452.

YANG Jian, YANG Jingyu. Uncorrelated image projection discriminant analysis and face recognition [J]. Journal of Computer Research and Development, 2003, 40(3): 447-452

[7] CHEN L F, LIAO H Y M, KO M T, et al. A new LDA-based face recognition system which can solve the small sample size problem[J]. Pattern Recognition, 2000, 33(10): 1713-1726.

[8] HUANG R, LIU Q, LU H, et al, Solving the small size problem of LDA[C]//Proc of 16th Inter Conf on Pattern Recognition. Piscataway, NJ, USA: IEEE Computer Society, 2002: 29-32.

[9] GOLUB G H, LOAN C F V. Matrix computations [M]. 3rd ed. Baltimore, Maryland, USA: The Johns Hopkins University Press, 1996:253-254.

[10] CHU Delin, THYE G S. A new and fast implementation for null space based linear discriminant analysis [J]. Pattern Recognition, 2010, 43(4): 1373-1379.

[11] WENG Juyang, ZHANG Yilu, HWANG Wey-Shiuan. Candid covariance-free incremental principal component analysis[J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2003, 25(8): 1034-1040.

(编辑 杜秀杰)