

Research Article

Fast Second-Order Orthogonal Tensor Subspace Analysis for Face Recognition

Yujian Zhou,¹ Liang Bao,² and Yiqin Lin¹

¹ Department of Mathematics and Computational Science, Institute of Computational Mathematics, Hunan University of Science and Engineering, Yongzhou 425100, China

² Department of Mathematics, East China University of Science and Technology, Shanghai 200237, China

Correspondence should be addressed to Liang Bao; nlbao@aliyun.com

Received 21 August 2013; Accepted 5 December 2013; Published 2 January 2014

Academic Editor: Jen-Tzung Chien

Copyright © 2014 Yujian Zhou et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Tensor subspace analysis (TSA) and discriminant TSA (DTSA) are two effective two-sided projection methods for dimensionality reduction and feature extraction of face image matrices. However, they have two serious drawbacks. Firstly, TSA and DTSA iteratively compute the left and right projection matrices. At each iteration, two generalized eigenvalue problems are required to solve, which makes them inapplicable for high dimensional image data. Secondly, the metric structure of the facial image space cannot be preserved since the left and right projection matrices are not usually orthonormal. In this paper, we propose the orthogonal TSA (OTSA) and orthogonal DTSA (ODTSA). In contrast to TSA and DTSA, two trace ratio optimization problems are required to be solved at each iteration. Thus, OTSA and ODTSA have much less computational cost than their nonorthogonal counterparts since the trace ratio optimization problem can be solved by the inexpensive Newton-Lanczos method. Experimental results show that the proposed methods achieve much higher recognition accuracy and have much lower training cost.

1. Introduction

Many applications in the field of information process, such as data mining, information retrieval, machine learning, and pattern recognition, require dealing with high-dimensional data. Dimensionality reduction has been a key technique for achieving high efficiency in manipulating the high-dimensional data. In dimensionality reduction, the high-dimensional data are transformed into a low-dimensional subspace with limited loss of information.

Principal component analysis (PCA) [1] and linear discriminant analysis (LDA) [2] are two of the most well-known and widely used dimension reduction methods. PCA is an unsupervised method, which aims to find the projection directions by maximizing variance of features in the low-dimensional subspace. It is also considered as the best data representation method in that the mean squared error between the original data and the data reconstructed using the PCA transform result is the minimum. LDA is a supervised method and is based on the following idea: the transform results of the data points of different classes should be

far as much as possible from each other and the transform results of the data points of the same class should be close as much as possible to each other. To achieve this goal, LDA seeks to find optimal linear transformation by minimizing the within-class distance and maximizing the between-class distance simultaneously. The optimal transformation of LDA can be computed by solving a generalized eigenvalue problem involving scatter matrices. LDA has been applied successfully for decades in many important applications including pattern recognition [2–4], information retrieval [5], face recognition [6, 7], microarray data analysis [8, 9], and text classification [10]. One of main drawbacks of LDA is that the scatter matrices are required to be nonsingular, which is not true when the data dimension is larger than the number of data samples. This is known as the undersampled problem and also called a small sampled problem [2]. To make LDA applicable for undersampled problems, researchers have proposed several variants of LDA including PCA + LDA [7, 11], LDA/GSVD [12, 13], two-stage LDA [14], regularized LDA [15–18], orthogonal LDA [19, 20], null space LDA [21, 22], and uncorrelated LDA [20, 23].

As is well known, both PCA and LDA only take into account the global Euclidean structure of the original data. However, the high-dimensional data in real-world often lie on or near a smooth low-dimensional manifold. So, it is important to preserve the local structure. Locality preserving projection (LPP) [24] is a locality structure preserving method that aims to preserve the intrinsic geometry of the original data. Usually, LPP has better performance than the methods only preserving the global structure information such as PCA and LDA for recognition problems. Moreover, LPP is not sensitive to noise and outliers. In its original form, LPP is only an unsupervised dimension reduction method. The supervised version of LPP (SLPP) [25] exploits the class label information of the training samples and thus has a higher classification accuracy than the unsupervised LPP. Other improvements to LPP include the discriminant locality preserving projection (DLPP) [26] and the orthogonal discriminant locality preserving projection method (ODLPP) [27].

During dealing with two-dimensional data such as images, the traditional approach is first to transform these image matrices into one-dimensional vectors and then apply these dimension reduction methods mentioned above to the vectorized image data. The approach of vectorizing image matrices can bring high computational cost and a loss of the underlying spatial structure information of the images. In order to overcome disadvantages of the vectorization approach, researchers have proposed 2D-PCA [28], 2D-LDA [29], 2D-LPP [30], and 2D-DLPP [31]. These methods are directly based on the matrix expression of image data. However, these two-dimensional methods only employ single-side projection and thus cannot still preserve the intrinsic spatial structure information of the images.

In the last decade, some researchers have developed several second-order tensor methods for dimension reduction of image data. These methods aim to find two subspaces for two-sided projection. Ye [32] has proposed a generalized low-rank approximation method (GLRAM), which seeks to find the left and right projections by minimizing the reconstruction error. Moreover, an iterative procedure is presented. One of the main drawbacks of GLRAM is that one eigenvalue decomposition is required at each iteration step. So, the computational cost is high. To overcome this disadvantage, Ren and Dai [33] have proposed to replace the projection vectors obtained from the eigenvalue decomposition by the bilinear Lanczos vectors at each iteration step of GLRAM. Experimental results show that the approach based on bilinear Lanczos vectors is competitive with the conventional GLRAM in classification accuracy, while it has a much lower computational cost. We note that GLRAM is an unsupervised method and only preserves the global Euclidean structure of the image data. Tensor subspace analysis (TSA) [34] is another two-sided projection method for dimension reduction and feature extraction of image data. TSA preserves the local structure information of the original data, while it does not employ the discriminant information. Wang et al. [35] have proposed a discriminant TSA (DTSA) by combining TSA with the discriminant information. Like GLRAM, both TSA and DTSA use an iterative procedure to compute the optimal solution of two projection matrices. At each iteration of TSA and DTSA,

two generalized eigenvalue problems are required to solve, which makes them inapplicable for dimension reduction and feature extraction of high-dimensional image data.

In this paper, we propose the orthogonal TSA (OTSA) and orthogonal DTSA (ODTSA) by constraining the left and right projection matrices to orthogonal matrices. Similarly to TSA and DTSA, OTSA and ODTSA also iteratively compute the left and right projection matrices. However, instead of solving two generalized eigenvalue problems as in TSA and DTSA, we require solving two trace ratio optimization problems at each iteration of OTSA and ODTSA during iteratively computing the left and right projection matrices. Thus, OTSA and ODTSA have much less computational cost than their nonorthogonal counterparts since the trace ratio optimization problem can be solved by the inexpensive Newton-Lanczos method. Two experiments on face recognition are conducted to evaluate the efficiency and effectiveness of the proposed OTSA and ODTSA. Experimental results show that these methods proposed in this paper achieve much higher recognition accuracy and have much lower training cost than TSA and DTSA.

The remainder of the paper is organized as follows. In Section 2, we briefly review TSA and DTSA. In Section 3, we firstly propose the OTSA and ODTSA. Then, we give a brief review of the trace ratio optimization problem and outline the Newton's method and the Newton-Lanczos for solving the trace ratio optimization problem. Finally, we present the algorithms for computing the left and right projection matrices of OTSA and ODTSA. Section 4 is devoted to numerical experiments. Some concluding remarks are provided in Section 5.

2. A Brief Review of TSA and DTSA

In this section, we give a brief review of TSA and DTSA, which are two recently proposed linear methods for dimension reduction and feature extraction of face recognition.

Given a set of N image data,

$$\mathcal{X} = \{X_1, X_2, \dots, X_N\}, \quad (1)$$

where $X_i \in \mathbb{R}^{L_1 \times L_2}$. For simplicity of discussion, we assume that the given data set \mathcal{X} is partitioned into C classes as

$$\mathcal{W} = \{X_1^{(1)}, X_2^{(1)}, \dots, X_{N_1}^{(1)}, X_1^{(2)}, X_2^{(2)}, \dots, X_{N_2}^{(2)}, \dots, X_1^{(C)}, X_2^{(C)}, \dots, X_{N_C}^{(C)}\}, \quad (2)$$

where N_c is the number of samples in the c th class and $N = \sum_{c=1}^C N_c$.

Let $W \in \mathbb{R}^{N \times N}$ denote the total within-class similarity matrix. Its entry is defined by

$$W_{ij} = \begin{cases} \exp\left(\frac{-\|X_i - X_j\|}{t}\right), & \text{if } X_i, X_j \text{ are from} \\ & \text{the same class,} \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where t is a positive parameter which can be determined empirically and $\|\cdot\|$ denotes the Frobenius norm for a matrix,

that is, $\|A\| = \sqrt{\sum_i \sum_j A_{ij}^2}$. Note that the total within-class similarity matrix W has a block diagonal form, where the c th block is the within-class similarity matrix W_c of the c th class and the size of the c th block is equal to the number N_c of samples in the c th class; that is,

$$W = \text{diag}(W_1, W_2, \dots, W_C). \quad (4)$$

The between-class similarity matrix B is defined as follows:

$$B_{ij} = \exp\left(\frac{-\|\bar{X}_i - \bar{X}_j\|}{t}\right), \quad i, j = 1, 2, \dots, C, \quad (5)$$

where

$$\bar{X}_i = \frac{1}{N_i} \sum_{k=1}^{N_i} X_k^{(i)} \quad (6)$$

is the mean of samples in the i th class.

Define the diagonal matrix $D = \text{diag}(d_1, d_2, \dots, d_N)$ with

$$d_i = \sum_{j=1}^N W_{ij}. \quad (7)$$

Then, $L_W = D - W$ is called the within-class Laplacian matrix and is symmetric positive semidefinite. Similarly, the between-class Laplacian matrix is defined as $L_B = E - B$, where E is a diagonal matrix with its i th entry being the row sum of the i th row of B .

In two-sided projection methods such as TSA and DTSA for dimension reduction and feature extraction of matrix data, we aim to find two projection matrices $U \in \mathbb{R}^{L_1 \times l_1}$, $V \in \mathbb{R}^{L_2 \times l_2}$ with $l_1 \leq L_1$ and $l_2 \leq L_2$ such that the low-dimensional data

$$Y_i = U^T X_i V \quad (8)$$

are easier to be distinguished.

2.1. Tensor Subspace Analysis. In TSA, we seek to find the left and right transformation matrices U, V by solving the following optimization problem:

$$\begin{aligned} & \max_{U, V} \frac{\sum_{i=1}^N d_i \|Y_i\|^2}{\sum_{i=1}^N \sum_{j=1}^N W_{ij} \|Y_i - Y_j\|^2} \\ & = \max_{U, V} \frac{\sum_{i=1}^N d_i \|U^T X_i V\|^2}{\sum_{i=1}^N \sum_{j=1}^N W_{ij} \|U^T X_i V - U^T X_j V\|^2}. \end{aligned} \quad (9)$$

The numerator part of the objective function in (9) denotes the global variance on the manifold in low-dimensional subspace, while the denominator part of the objective function is a measure of nearness of samples from the same class. Therefore, by maximizing the objective function, the samples from the same class are transformed into data points close to

each other and samples from different classes are transformed into data points far from each other.

Define

$$P_U = \begin{bmatrix} U^T X_1 \\ U^T X_2 \\ \vdots \\ U^T X_N \end{bmatrix}, \quad P_V = \begin{bmatrix} V^T X_1^T \\ V^T X_2^T \\ \vdots \\ V^T X_N^T \end{bmatrix}. \quad (10)$$

These two matrices, respectively, are called the total left and right transformation matrices in [35].

The optimization problem (9) can be equivalently rewritten as the following optimization problem:

$$\max_{U, V} \frac{\text{tr}(V^T P_U^T (D \otimes I_{l_1}) P_U V)}{\text{tr}(V^T P_U^T (L_W \otimes I_{l_1}) P_U V)}, \quad (11)$$

or

$$\max_{U, V} \frac{\text{tr}(U^T P_V^T (D \otimes I_{l_2}) P_V U)}{\text{tr}(U^T P_V^T (L_W \otimes I_{l_2}) P_V U)}. \quad (12)$$

Here and in the following, I_l denotes an identity matrix of order l and \otimes represents the Kronecker product of the matrices.

Clearly, from the equivalence between the maximization problem (9) and the optimization problem (11) or (12), we have the following results, from which an iterative algorithm for the computation of the transformation matrices U and V results.

Theorem 1. *Let U and V be the solution of the maximization problem (9). Then, Consider the following.*

- (1) For a given U, V consists of the l_2 eigenvectors of the generalized eigenvalue problem

$$[P_U^T (D \otimes I_{l_1}) P_U] v = \lambda [P_U^T (L_W \otimes I_{l_1}) P_U] v \quad (13)$$

corresponding to the largest l_2 eigenvalues.

- (2) For a given V, U consists of the l_1 eigenvectors of the generalized eigenvalue problem

$$[P_V^T (D \otimes I_{l_2}) P_V] u = \lambda [P_V^T (L_W \otimes I_{l_2}) P_V] u \quad (14)$$

corresponding to the largest l_1 eigenvalues.

Based on Theorem 1, iterative implementation of TSA has been given in Algorithm 1; see, also, [34].

2.2. Discriminant Tensor Subspace Analysis. In this subsection, we simply review the second-order DTSA, which is proposed in [35] for face recognition. DTSA combines the advantages of tensor methods and manifold methods and thus preserves the spatial structure information of the original image data and the local structure of the samples distribution. Moreover, by integrating the class label information

Input: A set of N sample matrices $\{X_i\}_{i=1}^N$ with class label information, l_1, l_2 .
Output: left and right transformation matrices U and V .
(1) Initialize U with an identity matrix;
(2) Until convergence Do:
(2.1) Form the matrix $M_D^{(U)} = P_U^T (D \otimes I_{l_1}) P_U$;
(2.2) Form the matrix $M_L^{(U)} = P_U^T (L_W \otimes I_{l_1}) P_U$;
(2.3) Compute the l_2 eigenvectors $\{v_i\}_{i=1}^{l_2}$ of the pencil $(M_D^{(U)}, M_L^{(U)})$ corresponding to the largest l_2 eigenvalues.
(2.4) Set $V = [v_1, v_2, \dots, v_{l_2}]$;
(2.5) Form the matrix $M_D^{(V)} = P_V^T (D \otimes I_{l_2}) P_V$;
(2.6) Form the matrix $M_L^{(V)} = P_V^T (L_W \otimes I_{l_2}) P_V$;
(2.7) Compute the l_1 eigenvectors $\{u_i\}_{i=1}^{l_1}$ of the pencil $(M_D^{(V)}, M_L^{(V)})$ corresponding to the largest l_1 eigenvalues.
(2.8) Set $U = [u_1, u_2, \dots, u_{l_1}]$;
End Do

ALGORITHM 1: TSA.

into TSA, DTSA obtains higher recognition accuracy for face recognition.

In DTSA, the optimization problem is described as follows:

$$\begin{aligned} & \max_{U,V} \frac{\sum_{i=1}^C \sum_{j=1}^C \|\bar{Y}_i - \bar{Y}_j\|^2 B_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \|Y_i - Y_j\|^2 W_{ij}} \\ & = \max_{U,V} \frac{\sum_{i=1}^C \sum_{j=1}^C \|U^T \bar{X}_i V - U^T \bar{X}_j V\|^2 B_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \|Y_i - Y_j\|^2 W_{ij}}, \end{aligned} \quad (15)$$

where \bar{X}_i is the mean of samples in the i th class.

We note that the objective function in (15) has the same denominator part as that of the objective function in (9) and however has a different numerator part from that of the objective function in (9). Since the numerator part of the objective function in (15) is established based on the class label information, DTSA has better performance than TSA for transforming samples from different classes into data points far from each other.

Define the mean left and right transformation matrices Q_U, Q_V by

$$Q_U = \begin{bmatrix} U^T \bar{X}_1 \\ U^T \bar{X}_2 \\ \vdots \\ U^T \bar{X}_C \end{bmatrix}, \quad Q_V = \begin{bmatrix} V^T \bar{X}_1^T \\ V^T \bar{X}_2^T \\ \vdots \\ V^T \bar{X}_C^T \end{bmatrix}. \quad (16)$$

Then, similarly, the optimization problem (15) can be equivalently formulated as the optimization problem

$$\max_{U,V} \frac{\text{tr}(V^T Q_U^T (L_B \otimes I_{l_1}) Q_U V)}{\text{tr}(V^T P_U^T (L_W \otimes I_{l_1}) P_U V)} \quad (17)$$

or the optimization problem

$$\max_{U,V} \frac{\text{tr}(U^T Q_V^T (L_B \otimes I_{l_2}) Q_V U)}{\text{tr}(U^T P_V^T (L_W \otimes I_{l_2}) P_V U)}, \quad (18)$$

where L_W is the within-class Laplacian matrix and L_B is the between-class Laplacian matrix.

Similarly, for the optimization problem (15), we have the following result.

Theorem 2. Let U and V be the solution of the maximization problem (15). Then, Consider the following.

- (1) For a given U , V consists of the l_2 eigenvectors of the generalized eigenvalue problem

$$[Q_U^T (L_B \otimes I_{l_1}) Q_U] v = \lambda [P_U^T (L_W \otimes I_{l_1}) P_U] v \quad (19)$$

corresponding to the largest l_2 eigenvalues.

- (2) For a given V , U consists of the l_1 eigenvectors of the generalized eigenvalue problem

$$[Q_V^T (L_B \otimes I_{l_2}) Q_V] u = \lambda [P_V^T (L_W \otimes I_{l_2}) P_V] u \quad (20)$$

corresponding to the largest l_1 eigenvalues.

The algorithm proposed in [35] for implementing DTSA is described in Algorithm 2.

3. Orthogonal TSA and DTSA

Although TSA and DTSA are two effective methods for dimension reduction and feature extraction of facial images, they still have two serious defects. Firstly, as shown in the section above, the column vectors of the left and right transformation matrices U and V are the eigenvectors of symmetric positive semidefinite pencils. So, they are not usually

Input: A set of N sample matrices $\{X_i\}_{i=1}^N$ with class label information, I_1, I_2 .
Output: left and right transformation matrices U and V .
(1) Initialize U with an identity matrix;
(2) Until convergence Do:
(2.1) Form the matrix $M_{L_B}^{(U)} = Q_U^T (L_B \otimes I_{I_1}) Q_U$;
(2.2) Form the matrix $M_{L_W}^{(U)} = P_U^T (L_W \otimes I_{I_1}) P_U$;
(2.3) Compute the l_2 eigenvectors $\{v_i\}_{i=1}^{l_2}$ of the pencil $(M_{L_B}^{(U)}, M_{L_W}^{(U)})$ corresponding to the largest l_2 eigenvalues.
(2.4) Set $V = [v_1, v_2, \dots, v_{l_2}]$;
(2.5) Form the matrix $M_{L_B}^{(V)} = Q_V^T (L_B \otimes I_{I_2}) Q_V$;
(2.6) Form the matrix $M_{L_W}^{(V)} = P_V^T (L_W \otimes I_{I_2}) P_V$;
(2.7) Compute the l_1 eigenvectors $\{u_i\}_{i=1}^{l_1}$ of the pencil $(M_{L_B}^{(V)}, M_{L_W}^{(V)})$ corresponding to the largest l_1 eigenvalues.
(2.8) Set $U = [u_1, u_2, \dots, u_{l_1}]$;
End Do

ALGORITHM 2: DTSA.

orthonormal. The requirement of the orthogonality of the columns of projection matrices is common in that orthogonal projection matrices preserve the metric structure of the facial image space. Thus, orthogonal methods have better locality preserving power and higher discriminating power than nonorthogonal methods. Secondly, at each iteration step of TSA algorithm or DTSA algorithm, two generalized eigenvalue problems are required to solve for iteratively computing the left and right projection matrices. As a result, when computational efficiency is critical, relatively high computational complexities of TSA and DTSA make them inapplicable for real applications.

In this section, we propose the orthogonal TSA (OTSA) and the orthogonal DTSA (ODTSA) for dimension reduction and feature extraction of facial images.

In OTSA, we seek to obtain the orthogonal projection matrices U and V by solving the optimization problem

$$\max_{U \in \mathbb{R}^{L_1 \times l_1}, V \in \mathbb{R}^{L_2 \times l_2}} \frac{\sum_{i=1}^N d_i \|Y_i\|^2}{\sum_{i=1}^N \sum_{j=1}^N W_{ij} \|Y_i - Y_j\|^2}, \quad (21)$$

$$U^T U = I_{l_1}, \quad V^T V = I_{l_2}$$

while in ODTSA, the optimization problem to be solved is

$$\max_{U \in \mathbb{R}^{L_1 \times l_1}, V \in \mathbb{R}^{L_2 \times l_2}} \frac{\sum_{i=1}^C \sum_{j=1}^C \|\bar{Y}_i - \bar{Y}_j\|^2 B_{ij}}{\sum_{i=1}^N \sum_{j=1}^N \|Y_i - Y_j\|^2 W_{ij}}, \quad (22)$$

$$U^T U = I_{l_1}, \quad V^T V = I_{l_2}.$$

Clearly, for OTSA and ODTSA, we have the following theorems.

Theorem 3. Let U and V be the solution of the maximization problem (21). Then, Consider the following.

- (1) For a given U , V is the solution of the trace ratio optimization problem

$$\max_{V \in \mathbb{R}^{L_2 \times l_2}, V^T V = I_{l_2}} \frac{\text{tr}(V^T P_U^T (D \otimes I_{I_1}) P_U V)}{\text{tr}(V^T P_U^T (L_W \otimes I_{I_1}) P_U V)}. \quad (23)$$

- (2) For a given V , U is the solution of the trace ratio optimization problem

$$\max_{U \in \mathbb{R}^{L_1 \times l_1}, U^T U = I_{l_1}} \frac{\text{tr}(U^T P_V^T (D \otimes I_{I_2}) P_V U)}{\text{tr}(U^T P_V^T (L_W \otimes I_{I_2}) P_V U)}. \quad (24)$$

Theorem 4. Let U and V be the solution of the maximization problem (22). Then, Consider the following.

- (1) For a given U , V is the solution of the trace ratio optimization problem

$$\max_{V \in \mathbb{R}^{L_2 \times l_2}, V^T V = I_{l_2}} \frac{\text{tr}(V^T Q_U^T (L_B \otimes I_{I_1}) Q_U V)}{\text{tr}(V^T P_U^T (L_W \otimes I_{I_1}) P_U V)}. \quad (25)$$

- (2) For a given V , U is the solution of the trace ratio optimization problem

$$\max_{U \in \mathbb{R}^{L_1 \times l_1}, U^T U = I_{l_1}} \frac{\text{tr}(U^T Q_V^T (L_B \otimes I_{I_2}) Q_V U)}{\text{tr}(U^T P_V^T (L_W \otimes I_{I_2}) P_V U)}. \quad (26)$$

The only difference between OTSA and TSA or between ODTSA and DTSA is that U and V are constrained to orthogonal matrices in OTSA and ODTSA. However, the projection matrices U and V of orthogonal methods are quite different from those of nonorthogonal methods. In nonorthogonal methods, U and V can be formulated by some eigenvectors of the generalized eigenvalue problems, while those of orthogonal methods are the solutions of the trace ratio optimization problems.

3.1. Trace Ratio Optimization. In this subsection, we consider the following trace ratio optimization problem:

$$\max_{V \in \mathbb{R}^{n \times l}, V^T V = I} \frac{\text{tr}(V^T A V)}{\text{tr}(V^T B V)}, \quad (27)$$

where $A, B \in \mathbb{R}^{n \times n}$ are symmetric matrices.

For the trace ratio optimization problem (27), we have the following result, which is given in [36].

Theorem 5. *Let A, B be two symmetric matrices and assume that B is positive semidefinite with rank greater than $n-l$. Then the ratio (27) admits a finite maximum value ρ_* .*

Define the function $f(\rho)$ as follows:

$$f(\rho) = \max_{V \in \mathbb{R}^{n \times l}, V^T V = I} \text{tr}(V^T (A - \rho B) V). \quad (28)$$

We collect some important properties presented in [36] on the function $f(\rho)$ in the following theorem. Some of them indicate the relation between the trace ratio optimization problem (27) and the function $f(\rho)$.

Theorem 6. *Let A, B be two symmetric matrices and assume that B is positive semidefinite with rank greater than $n-l$. Then*

- (1) $f(\rho)$ is a non-increasing function of ρ ;
- (2) $f(\rho) = 0$ if and only if $\rho = \rho_*$, where ρ_* is the finite maximum value of the ratio (27);
- (3) the derivative of $f(\rho)$ is given by

$$\frac{df(\rho)}{d\rho} = -\text{tr}[V(\rho)^T B V(\rho)], \quad (29)$$

where

$$V(\rho) = \arg \max_{V^T V = I} \text{tr}(V^T (A - \rho B) V); \quad (30)$$

- (4) the columns of the solution matrix V_* of the trace ratio optimization problem (27) consists of the l eigenvectors of the matrix $A - \rho_* B$ corresponding to the largest l eigenvalues, that is,

$$\begin{aligned} V_* &\equiv \arg \max_{V \in \mathbb{R}^{n \times l}, V^T V = I} \frac{\text{tr}(V^T A V)}{\text{tr}(V^T B V)} \\ &= \arg \max_{V^T V = I} \text{tr}(V^T (A - \rho_* B) V). \end{aligned} \quad (31)$$

Theorem 6 shows that instead of solving the trace ratio optimization problem (27), V_* , the solution of the trace ratio optimization problem (27), can be obtained through two steps:

- (1) compute the solution ρ_* of the nonlinear equation $f(\rho) = 0$;
- (2) compute the l eigenvectors of the matrix $A - \rho_* B$ corresponding to the largest l eigenvalues.

Newton's method [37] is the most well-known and widely used method for solving a nonlinear equation. The iterative scheme of Newton's method for solving $f(\rho) = 0$ takes the form

$$\begin{aligned} \rho_{k+1} &= \rho_k - \frac{f(\rho_k)}{f'(\rho_k)} \\ &= \rho_k - \frac{\text{tr}(V(\rho_k)^T (A - \rho_k B) V(\rho_k))}{-\text{tr}(V(\rho_k)^T B V(\rho_k))} \\ &= \frac{\text{tr}(V(\rho_k)^T A V(\rho_k))}{\text{tr}(V(\rho_k)^T B V(\rho_k))}, \end{aligned} \quad (32)$$

where $V(\rho_k) \in \mathbb{R}^{n \times l}$ consists of the l eigenvectors of the matrix $A - \rho_k B$ corresponding to the largest l eigenvalues.

We now outline the procedure of Newton's method for solving the trace ratio optimization problem (27) in Algorithm 3.

We remark that since Newton's method is commonly of quadratic convergence, only several iterations are required in Algorithm 3 for obtaining a good approximation of V_* . The main cost at each iteration in Algorithm 3 is due to the computation of the l eigenvectors of a symmetric matrix corresponding to the largest l eigenvalues.

3.2. Lanczos Vectors. In this subsections we review the Lanczos procedure for generating the Lanczos vectors of a symmetric matrix and the Newton-Lanczos method for solving the trace ratio optimization problem (27).

Given a symmetric matrix A and an initial unit vector v . Let $\mathcal{K}_l(A, v)$ denote the Krylov subspace associated with A and v , which is defined as

$$\mathcal{K}_l(A, v) = \text{span}\{v, Av, A^2v, \dots, A^{l-1}v\}. \quad (33)$$

The Lanczos vectors v_1, v_2, \dots, v_l , which form an orthonormal basis of the Krylov subspace $\mathcal{K}_l(A, v)$, can be established by the 3-term recurrence

$$\beta_{k+1} v_{k+1} = A v_k - \alpha_k v_k - \beta_k v_{k-1} \quad (34)$$

with $\beta_1 q_0 = 0$. The coefficients α_k and β_{k+1} are computed so as to ensure that $v_{k+1}^T v_k = 0$ and $\|v_{k+1}\|_2 = 1$. The pseudocode of the Lanczos procedure for constructing the Lanczos vectors v_1, v_2, \dots, v_l is outlined in Algorithm 4.

It is known [38] that Lanczos vectors are commonly good approximation of the eigenvectors of a symmetric matrix corresponding to the largest eigenvalues. So it is reasonable that the l eigenvectors of the matrix $A - \rho B$ corresponding to the largest l eigenvalues in Algorithm 3 are substituted by the Lanczos vectors of the matrix $A - \rho B$ to save the expensive cost for computing the l eigenvectors. This substitution deduces the Newton-Lanczos method for solving the trace ratio optimization problem (27), which is outlined in Algorithm 5; see, also, [36].

Input: A, B and a dimension l
Output: V which solves the trace ratio optimization problem (27)

- (1) Select an initial $n \times l$ unitary matrix V ;
- (2) Compute

$$\rho = \frac{\text{tr}(V^T AV)}{\text{tr}[V^T BV]}$$
- (3) Until convergence Do:
 - (3.1) Compute the l eigenvectors $\{v_i\}_{i=1}^l$ of the matrix $A - \rho B$ corresponding to the largest l eigenvalues.
 - (3.2) Set $V = [v_1, v_2, \dots, v_l]$;
 - (3.3) Compute

$$\rho = \frac{\text{tr}(V^T AV)}{\text{tr}[V^T BV]}$$

End Do

ALGORITHM 3: Newton's method for trace ratio optimization.

Input: A, v and a dimension l
Output: Lanczos vectors v_1, v_2, \dots, v_l

- (1) Set $v_1 = v, \beta_1 = 0$ and $v_0 = 0$;
- (2) For $k = 1 : l$
 - (2.1) $w_k = Av_k - \beta_k v_{k-1}$;
 - (2.2) $\alpha_k = w_k^T v_k$;
 - (2.3) $w_k = w_k - \alpha_k v_k$;
 - (2.4) $\beta_{k+1} = \|w_k\|_2$;
 - (2.5) $v_{k+1} = w_k / \beta_{k+1}$;
- End For

ALGORITHM 4: Lanczos procedure.

3.3. *OTSA and ODTSA.* Similarly, from Theorem 3, we can obtain two iterative procedures for computing the left and right transformation matrices U and V of OTSA and ODTSA. Algorithms 6 and 7 summarize the steps to compute U and V for OTSA and ODTSA, respectively.

The trace ratio optimization problem in Algorithms 6 and 7 can be solved by Newton's method or Newton-Lanczos method. For distinguishing these two cases, we use OTSA-N and ODTSA-N to denote the OTSA and ODTSA algorithms with the trace ratio optimization problem being solved by Newton's method and use OTSA-NL and ODTSA-NL to denote the OTSA and ODTSA algorithms with the trace ratio optimization problem being solved by Newton-Lanczos method.

3.4. *Computational Complexity Analysis.* We now discuss the computational complexity of TSA, DTSA, OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL.

In each iteration of TSA, it costs about $2l_1 L_1 L_2 N$, $2l_2 L_1 L_2 N$, $2l_1 L_2^2 N$, $2l_2 L_1^2 N$, and $2l_2 L_1^2 N$ flops (floating-point operations) for computing $P_U, P_V, M_D^U, M_L^U, M_D^V$, and M_L^V , respectively. Moreover, it takes $66L_2^3$ flops for computing the eigenvectors of the pencil (M_D^U, M_L^U) and $66L_1^3$

for (M_D^V, M_L^V) . So, the total cost for each iteration of TSA is about $(2(l_1 + l_2)L_1 L_2 + 4l_1 L_2^2 + 4l_2 L_1^2)N + 66(L_1^3 + L_2^3)$ flops.

The main difference between DTSA and TSA is that the matrix M_D^U, M_D^V in TSA is replaced by $M_{L_B}^U, M_{L_B}^V$ in DTSA, respectively. For computing $Q_U, Q_V, M_{L_B}^U$, and $M_{L_B}^V$ in each iteration of DTSA, it will spend about $2l_1 L_1 L_2 C, 2l_2 L_1 L_2 C, 2l_1 L_2^2 C$, and $2l_2 L_1^2 C$ flops. Thus, DTSA costs $(2(l_1 + l_2)L_1 L_2 + 2l_1 L_2^2 + 2l_2 L_1^2)(N + C) + 66(L_1^3 + L_2^3)$ flops for each iteration. In case $C \ll N$, The computation amount of DTSA is less than TSA.

It is known that for solving the trace ratio optimization problem (27), it costs $9Tn^3$ flops in Newton's method (Algorithm 3) and $2Tln^2$ in the Newton-Lanczos method (Algorithm 5), where T is the number of the Newton's iteration steps. So, the total cost for each iteration of OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL is about $(2(l_1 + l_2)L_1 L_2 + 4l_1 L_2^2 + 4l_2 L_1^2)N + 9T(L_1^3 + L_2^3)$, $(2(l_1 + l_2)L_1 L_2 + 2l_1 L_2^2 + 2l_2 L_1^2)(N + C) + 9T(L_1^3 + L_2^3)$, $(2(l_1 + l_2)L_1 L_2 + 4l_1 L_2^2 + 4l_2 L_1^2)N + 2T(l_1 L_1^2 + l_2 L_2^2)$, and $(2(l_1 + l_2)L_1 L_2 + 2l_1 L_2^2 + 2l_2 L_1^2)(N + C) + 2T(l_1 L_1^2 + l_2 L_2^2)$ flops, respectively. In general, 2 or 3 iteration steps are enough for the convergence of the Newton's iteration. Therefore, OTSA and ODTSA require less computation amount than TSA and DTSA.

Input: A, B and a dimension l
Output: V which solves the trace ratio optimization problem (27)

- (1) Select an initial $n \times l$ unitary matrix V ;
- (2) Compute

$$\rho = \frac{\text{tr}(V^T AV)}{\text{tr}[V^T BV]}$$
- (3) Until convergence Do:
 - (3.1) Compute the l Lanczos vectors $\{v_i\}_{i=1}^l$ of $A - \rho B$ by Algorithm 4.
 - (3.2) Set $V = [v_1, v_2, \dots, v_l]$;
 - (3.3) Compute

$$\rho = \frac{\text{tr}(V^T AV)}{\text{tr}[V^T BV]}$$

End Do

ALGORITHM 5: Newton-Lanczos method for trace ratio optimization.

Input: A set of N sample matrices $\{X_i\}_{i=1}^N$ with class label information, l_1, l_2 .
Output: left and right transformation matrices U and V .

- (1) Initialize U with an identity matrix;
- (2) Until convergence Do:
 - (2.1) Form the matrix $M_D^{(U)} = P_U^T (D \otimes I_{l_1}) P_U$;
 - (2.2) Form the matrix $M_L^{(U)} = P_U^T (L_W \otimes I_{l_1}) P_U$;
 - (2.3) Compute V by solving the trace ratio optimization problem (27) with $A = M_D^{(U)}$ and $B = M_L^{(U)}$;
 - (2.4) Form the matrix $M_D^{(V)} = P_V^T (D \otimes I_{l_2}) P_V$;
 - (2.5) Form the matrix $M_L^{(V)} = P_V^T (L_W \otimes I_{l_2}) P_V$;
 - (2.6) Compute U by solving the trace ratio optimization problem (27) with $A = M_D^{(V)}$ and $B = M_L^{(V)}$.

End Do

ALGORITHM 6: OTSA.

The time complexity for TSA, DTSA, OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL is presented in Table 1. We note that the space complexity of all the methods is $L_1 L_2 N$.

It is well known that Newton's iterative method for a nonlinear equation is commonly of quadratic convergence. So, it converges very fast for the nonlinear equation $f(\rho) = 0$, where $f(\rho)$ is defined in (28). We have observed in our numerical experiments that 5 Newton's iteration steps are enough for convergence. Therefore, the total computation costs of OTSA and ODTSA are much less than those of TSA and DTSA for obtaining the left and right transformation matrices U and V .

4. Experimental Results

In order to evaluate the performance of the proposed OTSA-N, ODTSA-N, OTSA-NL, ODTSA-NL algorithms, two well-known face image databases, that is, ORL (<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>) and Yale (<http://www.cad.zju.edu.cn/home/dengcai/Data/data.html>), are used in the experiments. We compare the recognition

performance of OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL algorithms with TSA [34] and DTSA [35]. In the experiments, the nearest neighbor classifier is used to classify the transformed results of samples obtained using different methods.

4.1. Experiment on the ORL Database of Face Images. The ORL database contains 400 images of 40 individuals. Each individual has 10 images, which were taken at different time, different lighting conditions, different facial expressions, and different accessories (glasses/no glasses). The sample images of one individual from the ORL database are shown in Figure 1.

We randomly select i ($i = 2, 3, \dots, 7, 8$) samples of each individual for training, and the remaining ones are used for testing. Based on the training set, the project matrices are obtained by TSA, DTSA, OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL. Then all the testing samples are projected to generate the low-dimensional samples, which will be recognized by using the nearest neighbor classifier. We repeat the process 10 times and calculate the mean and standard deviation of recognition rates.

Input: A set of N sample matrices $\{X_i\}_{i=1}^N$ with class label information, l_1, l_2 .
Output: left and right transformation matrices U and V .
(1) Initialize U with an identity matrix;
(2) Until convergence Do:
(2.1) Form the matrix $M_{L_B}^{(U)} = Q_U^T (L_B \otimes I_{l_1}) Q_U$;
(2.2) Form the matrix $M_{L_W}^{(U)} = P_U^T (L_W \otimes I_{l_1}) P_U$;
(2.3) Compute V by solving the trace ratio optimization problem (27) with $A = M_{L_B}^{(U)}$ and $B = M_{L_W}^{(U)}$;
(2.4) Form the matrix $M_{L_B}^{(V)} = Q_V^T (L_B \otimes I_{l_2}) Q_V$;
(2.5) Form the matrix $M_{L_W}^{(V)} = P_V^T (L_W \otimes I_{l_2}) P_V$;
(2.6) Compute U by solving the trace ratio optimization problem (27) with $A = M_{L_B}^{(V)}$ and $B = M_{L_W}^{(V)}$.
End Do

ALGORITHM 7: ODTSA.

TABLE 1: Time complexity of TSA, DTSA, OTSA-N, ODTSA-N, OTSA-NL, and ODTSA-NL.

Method	Time complexity
TSA	$(2(l_1 + l_2)L_1L_2 + 4l_1L_2^2 + 4l_2L_1^2)N + 66(L_1^3 + L_2^3)$
DTSA	$(2(l_1 + l_2)L_1L_2 + 2l_1L_2^2 + 2l_2L_1^2)(N + C) + 66(L_1^3 + L_2^3)$
OTSAN	$(2(l_1 + l_2)L_1L_2 + 4l_1L_2^2 + 4l_2L_1^2)N + 9T(L_1^3 + L_2^3)$
ODTSAN	$(2(l_1 + l_2)L_1L_2 + 2l_1L_2^2 + 2l_2L_1^2)(N + C) + 9T(L_1^3 + L_2^3)$
OTSA-NL	$(2(l_1 + l_2)L_1L_2 + 4l_1L_2^2 + 4l_2L_1^2)N + 2T(l_1L_1^2 + l_2L_2^2)$
ODTSA-NL	$(2(l_1 + l_2)L_1L_2 + 2l_1L_2^2 + 2l_2L_1^2)(N + C) + 2T(l_1L_1^2 + l_2L_2^2)$

TABLE 2: Recognition accuracy (%) on ORL database (mean \pm std).

	TSA	DTSA	OTSAN	OTSA-NL	ODTSAN	ODTSA-NL
2 Train	75.62 \pm 3.02	78.03 \pm 3.28	77.89 \pm 3.34	77.44 \pm 3.24	81.25 \pm 3.04	80.78 \pm 3.16
3 Train	84.56 \pm 2.42	88.07 \pm 2.26	87.32 \pm 2.45	87.54 \pm 2.35	89.87 \pm 2.44	89.57 \pm 2.47
4 Train	90.19 \pm 1.55	92.25 \pm 1.80	91.67 \pm 1.56	91.87 \pm 1.42	93.95 \pm 1.84	93.64 \pm 1.78
5 Train	92.55 \pm 1.16	94.35 \pm 1.23	93.30 \pm 2.08	93.35 \pm 1.78	95.39 \pm 1.28	95.42 \pm 1.38
6 Train	93.77 \pm 1.44	95.63 \pm 1.79	94.65 \pm 1.80	94.46 \pm 1.69	96.34 \pm 1.76	94.50 \pm 1.58
7 Train	94.66 \pm 1.36	96.25 \pm 1.98	95.67 \pm 2.19	95.48 \pm 1.98	97.88 \pm 2.09	97.60 \pm 1.23
8 Train	96.50 \pm 1.75	97.25 \pm 1.65	97.32 \pm 1.44	97.52 \pm 1.56	98.76 \pm 1.30	98.45 \pm 1.57

TABLE 3: Training time (second) on ORL database.

	TSA	DTSA	OTSAN	OTSA-NL	ODTSAN	ODTSA-NL
4 Train	1.5440	1.6573	0.8912	0.2438	0.8766	0.2518
8 Train	2.1647	2.2371	1.0576	0.3429	1.1592	0.3262

In our experiments, the parameters l_1 and l_2 in all the methods are set to be 10. The parameter t is set to 1. The mean and standard deviation of recognition accuracy of 10 runs of tests of six algorithms are presented in Table 2. The training time for each method is presented in Table 3. It shows that for all methods, the recognition increases with the increase in training sample size. Moreover, the orthogonal methods have higher recognition accuracy than their nonorthogonal versions, and the orthogonal methods based on the Newton-Lanczos approach cost least computational time.

4.2. Experiment on the Yale Database. The Yale face database contains 165 gray-scale images for 15 individuals where each individual has 11 images. These facial images have variations in lighting conditions (left-light, center-light, right-light), facial expressions (normal, happy, sad, sleepy, surprised, and wink), and with/without glasses. The 11 sample images of one individual from the Yale database are shown in Figure 2.

As in the previous experiments, the parameters l_1 and l_2 are set to be 10, and t is set to 1. The mean and standard deviation of recognition accuracy of 10 runs of tests for the

TABLE 4: Recognition accuracy (%) on Yale database (mean \pm std).

	TSA	DTSA	OTSA-N	OTSA-NL	ODTSA-N	ODTSA-NL
2 Train	41.78 \pm 5.04	45.41 \pm 5.97	43.41 \pm 5.97	43.51 \pm 6.10	47.59 \pm 5.36	47.23 \pm 5.87
3 Train	52.25 \pm 4.43	56.17 \pm 3.40	54.72 \pm 3.44	54.61 \pm 4.09	58.33 \pm 3.15	58.47 \pm 3.54
4 Train	59.14 \pm 3.90	63.10 \pm 4.00	62.10 \pm 4.04	62.76 \pm 4.20	66.45 \pm 4.32	66.33 \pm 4.15
5 Train	64.22 \pm 3.40	65.78 \pm 4.51	65.34 \pm 4.41	65.69 \pm 4.53	67.98 \pm 4.09	67.63 \pm 4.60
6 Train	69.00 \pm 4.54	70.13 \pm 5.59	70.52 \pm 5.33	70.46 \pm 5.84	72.88 \pm 5.67	72.59 \pm 5.48
7 Train	72.17 \pm 3.43	74.29 \pm 4.65	74.47 \pm 4.92	74.65 \pm 3.70	76.79 \pm 4.28	76.72 \pm 4.12
8 Train	74.78 \pm 4.38	76.89 \pm 5.62	76.56 \pm 5.13	76.83 \pm 6.27	79.43 \pm 5.05	79.26 \pm 5.42

TABLE 5: Training time (second) on Yale database.

	TSA	DTSA	OTSA-N	OTSA-NL	ODTSA-N	ODTSA-NL
4 Train	0.5497	0.5744	0.3468	0.0980	0.3362	0.0933
8 Train	0.7808	0.8059	0.4378	0.1350	0.5484	0.1311



FIGURE 1: Sample images for one individual of the ORL database.



FIGURE 2: Sample images for one individual of the Yale database.

Yale database are presented in Table 4. The training time of each method for the Yale database is presented in Table 5. Clearly, ODTSA-N and ODTSA-NL perform better than TSA, DTSA, OTSA-N, and OTSA-NL for this database, and ODTSA-NL and OTSA-NL outperform TSA, DTSA, OTSA-N, and ODTSA-N according to computational time.

5. Conclusion

In this paper, we propose an orthogonal TSA and orthogonal DTSA for face recognition by constraining the left and right projection matrices to orthogonal matrices. Similarly to TSA and DTSA, OTSA and ODTSA also iteratively compute the left and right projection matrices. However, instead of solving two generalized eigenvalue problems as in TSA and DTSA, it requires solving two trace ratio optimization problems at each iteration of OTSA and ODTSA during iteratively computing the left and right projection matrices. Thus, OTSA and ODTSA have much less computational cost than their nonorthogonal counterparts since the trace ratio optimization problem can be solved by the inexpensive Newton-Lanczos method. Experimental results show that these methods proposed in this paper achieve much higher recognition accuracy and have much lower training cost than TSA and DTSA.

Conflict of Interests

The authors declare that there is no conflict of interests.

Acknowledgments

Yiqin Lin is supported by the National Natural Science Foundation of China under Grant 10801048, the Natural Science Foundation of Hunan Province under Grant 11JJ4009, the Scientific Research Foundation of Education Bureau of Hunan Province for Outstanding Young Scholars in University under Grant 10B038, the Science and Technology Planning Project of Hunan Province under Grant 2010JT4042, and the Chinese Postdoctoral Science Foundation under Grant 2012M511386. Liang Bao is supported by the National Natural Science Foundation of China under Grants 10926150 and 11101149 and the Fundamental Research Funds for the Central Universities.

References

- [1] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, Computer Science and Scientific Computing, Academic Press, Boston, Mass, USA, 2nd edition, 1990.
- [3] C. M. Bishop, *Pattern Recognition and Machine Learning*, Information Science and Statistics, Springer, New York, NY, USA, 2006.
- [4] R. Q. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, John Wiley & Sons, New York, NY, USA, 2nd edition, 2001.
- [5] G. Kowalski, *Information Retrieval Systems: Theory and Implementation*, Kluwer Academic Publishers, Norwell, Mass, USA, 1997.
- [6] Z. Jin, J.-Y. Yang, Z.-S. Hu, and Z. Lou, "Face recognition based on the uncorrelated discriminant transformation," *Pattern Recognition*, vol. 34, no. 7, pp. 1405–1416, 2001.
- [7] D. L. Swets and J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831–836, 1996.
- [8] P. Baldi and G. W. Hatfield, *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*, Cambridge University Press, Cambridge, Mass, USA, 2002.
- [9] S. Dudoit, J. Fridlyand, and T. P. Speed, "Comparison of discrimination methods for the classification of tumors using gene expression data," *Journal of the American Statistical Association*, vol. 97, no. 457, pp. 77–87, 2002.

- [10] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice Hall Advanced Reference Series, Prentice Hall, Englewood Cliffs, NJ, USA, 1988.
- [11] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. fisherfaces: recognition using class specific linear projection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711–720, 1997.
- [12] P. Howland and H. Park, "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 995–1006, 2004.
- [13] J. Ye, R. Janardan, C. H. Park, and H. Park, "An optimization criterion for generalized discriminant analysis on undersampled problems," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 982–994, 2004.
- [14] J. Ye and Q. Li, "A two-stage linear discriminant analysis via QR-decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 929–941, 2005.
- [15] D.-Q. Dai and P. C. Yuen, "Regularized discriminant analysis and its application to face recognition," *Pattern Recognition*, vol. 36, no. 3, pp. 845–847, 2003.
- [16] J. H. Friedman, "Regularized discriminant analysis," *Journal of the American Statistical Association*, vol. 84, no. 405, pp. 165–175, 1989.
- [17] Y. Guo, T. Hastie, and R. Tibshirani, "Regularized linear discriminant analysis and its application in microarrays," *Biostatistics*, vol. 8, no. 1, pp. 86–100, 2007.
- [18] W.-K. Ching, D. Chu, L.-Z. Liao, and X. Wang, "Regularized orthogonal linear discriminant analysis," *Pattern Recognition*, vol. 45, no. 7, pp. 2719–2732, 2012.
- [19] D. Chu and S. T. Goh, "A new and fast orthogonal linear discriminant analysis on undersampled problems," *SIAM Journal on Scientific Computing*, vol. 32, no. 4, pp. 2274–2297, 2010.
- [20] J. Ye, "Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems," *Journal of Machine Learning Research*, vol. 6, pp. 483–502, 2005.
- [21] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "New LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713–1726, 2000.
- [22] D. Chu and G. S. Thye, "A new and fast implementation for null space based linear discriminant analysis," *Pattern Recognition*, vol. 43, no. 4, pp. 1373–1379, 2010.
- [23] D. Chu, S. T. Goh, and Y. S. Hung, "Characterization of all solutions for undersampled uncorrelated linear discriminant analysis problems," *SIAM Journal on Matrix Analysis and Applications*, vol. 32, no. 3, pp. 820–844, 2011.
- [24] X. He and P. Niyogi, "Locality preserving projections," *Advances in Neural Information Processing Systems*, vol. 16, pp. 153–160, 2004.
- [25] Z. Zheng, F. Yang, W. Tan, J. Jia, and J. Yang, "Gabor feature-based face recognition using supervised locality preserving projection," *Signal Processing*, vol. 87, no. 10, pp. 2473–2483, 2007.
- [26] W. Yu, X. Teng, and C. Liu, "Face recognition using discriminant locality preserving projections," *Image and Vision Computing*, vol. 24, no. 3, pp. 239–248, 2006.
- [27] L. Zhu and S. Zhu, "Face recognition based on orthogonal discriminant locality preserving projections," *Neurocomputing*, vol. 70, no. 7–9, pp. 1543–1546, 2007.
- [28] J. Yang, D. Zhang, A. F. Frangi, and J.-Y. Yang, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131–137, 2004.
- [29] M. Li and B. Yuan, "2D-LDA: a statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 527–532, 2005.
- [30] S. Chen, H. Zhao, M. Kong, and B. Luo, "2D-LPP: a two-dimensional extension of locality preserving projections," *Neurocomputing*, vol. 70, no. 4–6, pp. 912–921, 2007.
- [31] W. Yu, "Two-dimensional discriminant locality preserving projections for face recognition," *Pattern Recognition Letters*, vol. 30, no. 15, pp. 1378–1383, 2009.
- [32] J. Ye, "Generalized low rank approximations of matrices," *Machine Learning*, vol. 61, no. 1–3, pp. 167–191, 2005.
- [33] C.-X. Ren and D.-Q. Dai, "Bilinear Lanczos components for fast dimensionality reduction and feature extraction," *Pattern Recognition*, vol. 43, no. 11, pp. 3742–3752, 2010.
- [34] X. He and P. Niyogi, "Tensor subspace analysis," in *Advances in Neural Information Processing Systems*, vol. 18, 2005.
- [35] S.-J. Wang, C.-G. Zhou, N. Zhang, X.-J. Peng, Y.-H. Chen, and X. Liu, "Face recognition using second-order discriminant tensor subspace analysis," *Neurocomputing*, vol. 74, no. 12–13, pp. 2142–2156, 2011.
- [36] T. T. Ngo, M. Bellalij, and Y. Saad, "The trace ratio optimization problem for dimensionality reduction," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 5, pp. 2950–2971, 2010.
- [37] C. T. Kelley, *Solving Nonlinear Equations with Newton's Method*, vol. 1 of *Fundamentals of Algorithms*, SIAM, Philadelphia, Pa, USA, 2003.
- [38] B. N. Parlett, *The Symmetric Eigenvalue Problem*, vol. 20 of *Classics in Applied Mathematics*, SIAM, Philadelphia, Pa, USA, 1998.