Pattern Recognition 45 (2012) 2205–2213

Contents lists available at SciVerse ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

A new perspective to null linear discriminant analysis method and its fast implementation using random matrix multiplication with scatter matrices

Alok Sharma [a,b,c,*], Kuldip K. Paliwal [a]

[a] Signal Processing Lab, Griffith University, Brisbane, Australia
[b] Human Genome Center, Institute of Medical Science, University of Tokyo, Tokyo, Japan
[c] School of Engineering & Physics, University of the South Pacific, Suva, Fiji

ARTICLE INFO

Article history:
Received 6 October 2010
Received in revised form
13 October 2011
Accepted 28 November 2011
Available online 24 December 2011

Keywords:
Null LDA
Small Sample Size Problem
Dimensionality Reduction

ABSTRACT

Null linear discriminant analysis (LDA) method is a popular dimensionality reduction method for solving small sample size problem. The implementation of null LDA method is, however, computationally very expensive. In this paper, we theoretically derive the null LDA method from a different perspective and present a computationally efficient implementation of this method. Eigenvalue decomposition (EVD) of $\mathbf{S}_T^+\mathbf{S}_B$ (where $\mathbf{S}_B$ is the between-class scatter matrix and $\mathbf{S}_T^+$ is the pseudoinverse of the total scatter matrix $\mathbf{S}_T$) is shown here to be a sufficient condition for the null LDA method. As EVD of $\mathbf{S}_T^+\mathbf{S}_B$is computationally expensive, we show that the utilization of random matrix together with $\mathbf{S}_T^+\mathbf{S}_B$ is also a sufficient condition for null LDA method. This condition is used here to derive a computationally fast implementation of the null LDA method. We show that the computational complexity of the proposed implementation is significantly lower than the other implementations of the null LDA method reported in the literature. This result is also confirmed by conducting classification experiments on several datasets.

1. Introduction

Dimensionality reduction is an important aspect of pattern classification. It helps in improving the robustness (or generalization capability) of the pattern classifier and in reducing its computational complexity. The two well known dimensionality reduction techniques are principal component analysis (PCA) and linear discriminant analysis (LDA) [12]. PCA is an unsupervised learning algorithm and LDA is a supervised learning technique.[1]

The LDA technique finds an orientation matrix $\mathbf{W}$ that transforms high dimensional feature vectors belonging to different classes to lower dimensional feature vectors such that the projected feature vectors of a class are well separated from the feature vectors of other classes. If the dimensionality reduction is from d-dimensional space to h-dimensional space (where $h<d$), then the orientation (or transformation) matrix $\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\ldots\mathbf{w}_h]$ belongs to $\Re^{d\times h}$ and is of rank h; i.e., it has h non-zero column vectors that are linearly independent. For a c-class problem, the value of h will be $c-1$ or less; i.e., $1\leq h\leq c-1$. The orientation $\mathbf{W}$ is obtained by maximizing the Fisher's criterion function. This criterion function depends on three factors: orientation $\mathbf{W}$, within-class scatter matrix ($S_W \in \Re^{d\times d}$) or total scatter matrix ($S_T \in \Re^{d\times d}$) and between-class scatter matrix ($S_B \in \Re^{d\times d}$). The Fisher's discriminant ratio can be given by $|\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}|/|\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}|$. It has been shown in the literature [12] that the modified version of Fisher's criterion

$$J(\mathbf{W})=\frac{|\mathbf{W}^{\mathrm{T}}\mathbf{S}_B\mathbf{W}|}{|\mathbf{W}^{\mathrm{T}}\mathbf{S}_T\mathbf{W}|} \tag{1}$$

produces similar results. In the conventional LDA technique, the within-class scatter matrix ($\mathbf{S}_W$) or total scatter matrix ($\mathbf{S}_T$) (depending upon the criterion taken) needs to be non-singular.

In this paper, we are interested in a small sample size (SSS) problem [12], where the dimensionality of the feature space (d) is very large compared to the number of training samples (n). A number of pattern recognition applications (such as cancer classification from microarray data, face recognition, etc.) fall in this category. When the number of training samples is less than the dimensionality, the scatter matrices $\mathbf{S}_W$ and $\mathbf{S}_T$ become singular and it is not possible to use the conventional LDA technique for dimensionality reduction. This drawback is considered to be the main problem of LDA and is known as the SSS problem [12].

Several methods have been proposed to overcome the SSS problem. These include pseudo-inverse LDA method [40,32], regularized LDA method [11,14], Fisherface LDA method [37,2], direct LDA method [47], and null LDA method [6]. Some other related methods are reported in [22,16,15,34,35,23,27,28,26,25,30,5]. Among these methods, the null LDA method is a highly competitive method

[*] Corresponding author.
E-mail address: sharma_al@usp.ac.fj (A. Sharma).

[1] For detailed description about pattern classification capabilities of PCA and LDA techniques see Ref. [19] and Jiang [20].

0031-3203/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2011.11.018

in terms of classification performance and has been very popular in the pattern recognition literature.

In the null LDA method, the dimensionality is reduced in two stages. In the first stage, the training samples are projected on the null space of within-class scatter matrix $\mathbf{S}_W$ (i.e., the range space of $\mathbf{S}_W$ is discarded). In the second stage, the dimensionality is reduced by choosing h eigenvectors of the transformed between-class scatter matrix corresponding to the highest eigenvalues. Therefore, the null LDA method optimizes the Fisher's criterion sequentially in two stages.

The computational complexity of the null LDA method is approximately $O(d^2n)$, which is very high when the dimensionality of the feature space is very large. In order to reduce this computational complexity, the principal component analysis (PCA) plus null space method has been proposed [17,43]. In this method, a pre-processing step is introduced where PCA technique is applied to reduce the dimensionality from d to $n-1$ by removing the null space of total-scatter matrix $\mathbf{S}_T$ (assuming feature vectors are linearly independent and, thus, rank($\mathbf{S}_T$) $= n-1$). It has been shown [17,43] that this pre-processing step does not discard any useful discriminative information as $\mathbf{S}_B$ and $\mathbf{S}_W$ are zero in the null space of $\mathbf{S}_T$. In the reduced $n-1$ dimensional space, it is manageable to compute the null space of $\mathbf{S}_W$. This pre-processing step is then followed by the two steps of the null LDA method. The computational complexity of PCA+null LDA method is estimated to be $16dn^2+4dnc$ flops (for $d \gg n$). The computational complexity is also reduced by Ye [44]. He has proposed orthogonal LDA (OLDA) method which is shown to be equivalent to the null space based method under a mild condition; i.e., when the training vectors are linearly independent [45]. In his method, the orientation matrix $\mathbf{W}$ is obtained by simultaneously diagonalizing scatter matrices. The computational complexity of OLDA method is estimated to be $14dn^2+4dnc+2dc^2$ flops (where c is the number of classes). Recently, Chu and Thye [7] proposed a new implementation of null LDA method by doing QR decomposition. There approach requires approximately $4dn^2+2dnc$ computations. Though these methods exhibit faster implementations of null LDA method, their computational complexity is still high (as d and n grow larger and $d \gg n$).

In this paper, we present a new computationally fast procedure for the null space method. The computational complexity of our implementation is dn^2+2dnc and can be reduced to $O(dn^{1.376})$, which is significantly lower than other implementations of null LDA method. Here, we derive this procedure theoretically and demonstrate its effectiveness empirically on several datasets.

2. Null LDA method: alternative derivation

2.1. Basic notations

Let $\mathcal{X}$ be a set of n training vectors (samples or patterns) in a d-dimensional feature space, and $\Omega = \{\omega_i : i = 1,2,...,c\}$ be the finite set of c class labels, where ω_i denotes the ith class label. The set $\mathcal{X}$ can be subdivided into c subsets $\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_c$ (where subset $\mathcal{X}_i$ belongs to ω_i); i.e., $\mathcal{X}_i \subset \mathcal{X}$ and $\mathcal{X}_1 \cup \mathcal{X}_2 \cup \cdots \cup \mathcal{X}_c = \mathcal{X}$. Let n_i be the number of samples in class ω_i such that:

$$n = \sum_{i=1}^{c} n_i$$

The samples or vectors of set $\mathcal{X}$ can be written as:

$$\mathcal{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}, \text{ where } \mathbf{x}_j \in R^d.$$

Let $\boldsymbol{\mu}_j$ be the centroid of χ_j and $\boldsymbol{\mu}$ be the centroid of χ, then the between-class scatter matrix $\mathbf{S}_B$ is given by

$$\mathbf{S}_B = \sum_{j=1}^{c} n_j(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^{\mathrm{T}}$$

The within-class scatter matrix $\mathbf{S}_W$ is defined as

$$\mathbf{S}_W = \sum_{j=1}^{c} \mathbf{S}_j,$$

where

$$\mathbf{S}_j = \sum_{\mathbf{x} \in \chi_j} (\mathbf{x} - \boldsymbol{\mu}_j)(\mathbf{x} - \boldsymbol{\mu}_j)^{\mathrm{T}}$$

The total-class scatter matrix $\mathbf{S}_T$ is defined as

$$\mathbf{S}_T = \sum_{j=1}^{n} (\mathbf{x}_j - \boldsymbol{\mu})(\mathbf{x} - \boldsymbol{\mu})^{\mathrm{T}}.$$

It can be shown [10] that $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$. The matrix $\mathbf{S}_T$ can also be formed as follows $\mathbf{S}_T = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, where $\mathbf{A} \in \Re^{d \times n}$ is defined as

$$\mathbf{A} = \left[(\mathbf{x}_1 - \boldsymbol{\mu}), (\mathbf{x}_2 - \boldsymbol{\mu}), \ldots, (\mathbf{x}_n - \boldsymbol{\mu})\right]$$

In a similar way, $\mathbf{S}_B$ can be formed as $\mathbf{S}_B = \mathbf{B}\mathbf{B}^{\mathrm{T}}$, where rectangular matrix $\mathbf{B} \in \Re^{d \times c}$ can be defined as

$$\mathbf{B} = [\sqrt{n_1}(\boldsymbol{\mu}_1 - \boldsymbol{\mu}), \sqrt{n_2}(\boldsymbol{\mu}_2 - \boldsymbol{\mu}), \ldots, \sqrt{n_c}(\boldsymbol{\mu}_c - \boldsymbol{\mu})]$$

It can be seen that $\mathbf{S}_T$, $\mathbf{S}_B$ and $\mathbf{S}_W$ are symmetric matrices. In this paper, we assume that the n training vectors or patterns are linearly independent. Therefore, the ranks of matrices S_T,S_B, andS_W are $t = n-1$, $b = c-1$ and $n-c$, respectively. Thus, rank($\mathbf{S}_T$) = rank($\mathbf{S}_B$) + rank($\mathbf{S}_W$).

2.2. Basis

The essence of null LDA method is to find the orientation or transformation matrix $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_h] \in \Re^{d \times h}$ (of rank h) that satisfies the following two criteria (or conditions):

$$\mathbf{S}_W \mathbf{W} = 0, \tag{2}$$

and

$$\mathbf{S}_B \mathbf{W} \neq 0 \tag{3}$$

Under these two conditions (Eqs. (2) and (3)), it can be seen that the modified Fisher's ratio (Eq. (1)) attains a maximum value of 1; i.e., $J(\mathbf{w}_i) = 1$ for $i = 1 \ldots h$. Note that $\mathbf{W}$ satisfying these two conditions will have $c-1$ independent column vectors. When $h = c-1$, then this $\mathbf{W}$ defines the transformation matrix for the null LDA method. However, when $1 \leq h < c-1$, then the transformation matrix $\mathbf{W} \in \Re^{d \times h}$ (with h column vectors) for null LDA method can be obtained by

$$\mathbf{W} = \mathbf{arg} \max_{|W^{\mathrm{T}} S_W W| = 0} \left| \mathbf{W}^{\mathrm{T}} \mathbf{S}_B \mathbf{W} \right| \tag{4}$$

Since $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$, we can write $S_W \mathbf{W} = 0$ as

$$\begin{aligned} &(\mathbf{S}_T - \mathbf{S}_B)\mathbf{W} = 0 \\ &\text{or} \quad \mathbf{S}_B \mathbf{W} = \mathbf{S}_T \mathbf{W} \\ &\text{or} \quad \mathbf{W} = \mathbf{S}_T^{-1} \mathbf{S}_B \mathbf{W} \end{aligned} \tag{5}$$

Eq. (5) is a necessary condition for null LDA method; i.e., if $\mathbf{W}$ defines the transformation matrix for the null LDA method, then it has to satisfy this equation. This condition can also be shown to be sufficient for the null LDA method; i.e., if $\mathbf{W} \in \Re^{d \times h}$ (of rank h) satisfies this equation, then it will satisfy the two above-mentioned criteria of the null LDA method (see Appendix-A for the proofs). The problem with Eq. (5) is that $\mathbf{S}_T$ becomes singular in SSS problem and it is not possible to compute the inverse of matrix $\mathbf{S}_T$. Therefore, for singular cases, the approximation of the inverse of $\mathbf{S}_T$ is used:

$$\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W} \tag{6}$$

where $\mathbf{S}_T^{+}$ is the pseudo inverse of $\mathbf{S}_T$. It will be shown in the next sub-section that Eq. (6) is a sufficient condition for the null LDA method. Though this equation can be used to compute the transformation $\mathbf{W}$ for the null LDA method, it has the problem that it requires eigenvalue decomposition of $\mathbf{S}_T^{+}\mathbf{S}_B$ which is very difficult to compute due to the large size of $\mathbf{S}_T^{+}\mathbf{S}_B$. In order to make the computation of $\mathbf{W}$ to be easier, we replace $\mathbf{W}$ on the right hand side of Eq. (6) by a random matrix $\mathbf{Y}\in\Re^{d\times(c-1)}$of rank $c-1$; i.e., we use the following equation to compute the orientation matrix $\mathbf{W}$:

$$\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{Y} \tag{7}$$

We will prove in the next sub-section that this equation is a sufficient condition for the null LDA method. The matrix $\mathbf{W}\in\Re^{d\times(c-1)}$ obtained in this manner has $c-1$ linearly independent vectors as its columns and these vectors may not be orthonormal. However, if we want to have $\mathbf{W}\in\Re^{d\times h}$ with h orthonormal vectors (where $1\leq h\leq c-1$), then eigen-value decomposition (EVD) can be applied to select h leading orthonormal eigenvectors of $\mathbf{W}^{\mathrm{T}}S_B\mathbf{W}$. If $h=c-1$, then QR decomposition can also be applied on the column vectors of $\mathbf{W}$ to make these vectors orthonormal. Thus, we can ensure that the h eigenvectors obtained by using either EVD or QR decomposition will always be orthonormal. The matrix $\mathbf{W}\in\Re^{d\times h}$ obtained from these h eigenvectors defines the orientation or transformation matrix for the proposed null LDA method and it is used for reducing the dimensionality from d to h.

The random matrix $\mathbf{Y}\in\Re^{d\times(c-1)}$ used in eq. 7 has the following two properties: 1) it is $c-1$ column vectors are linearly independent; i.e., the rank of random matrix is $c-1$, and 2) when it is multiplied with the matrix $\mathbf{S}_T^{+}\mathbf{S}_B$ of rank $c-1$, it is product $(\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{Y})$ will also have rank equal to $c-1$ as no two elements of random matrix Y are identical and its elements Y_{ij} are random numbers uniformly distributed in the range $0<Y_{ij}<1$. This will make sure that all the $c-1$ column vectors of $\mathbf{W}$ are independent.

Furthermore, the training feature vectors are assumed to be linearly independent; i.e., the rank of $\mathbf{S}_T^{+}\mathbf{S}_B$ will always be equal to the number of classes minus one ($c-1$). This will ensure the dimensionality reduction of feature vectors from d-dimensional space to $c-1$ dimensional space. If these vectors are linearly dependent, then dependent vectors can be surgically removed (though we have never observed this case for the databases we have investigated in this paper).

2.3. Proof

In this sub-section we first prove that Eq. (6) is a sufficient condition for the null LDA method and then prove the sufficiency of Eq. (7) for the null LDA method.

2.3.1. Proof of equation 6 being a sufficient condition for the null LDA method

Here we show that $\mathbf{W}$ given by Eq. (6) is sufficient condition for the null LDA method; i.e., when $\mathbf{W}$ satisfies Eq. (6), it will also satisfy $\mathbf{S}_W\mathbf{W}=0$ and $\mathbf{S}_B\mathbf{W}\neq 0$. This proof is given below in the form of Theorems 1 and 2.

Theorem 1. If the matrix $\mathbf{W}\in\Re^{d\times h}$ satisfies the relation $\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W}$, then it is in the null space of $\mathbf{S}_W$; i.e., $\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}=0$.

Proof 1. Let us define

$$\mathbf{H}(\mathbf{W})=\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W} \tag{T1.1}$$

Using $\mathbf{S}_T=\mathbf{S}_B+\mathbf{S}_W$, it becomes

$$\mathbf{H}(\mathbf{W})=\mathbf{W}^{\mathrm{T}}(\mathbf{S}_T-\mathbf{S}_B)\mathbf{W} \tag{T1.2}$$

It is given that

$$\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W} \tag{T1.3}$$

Substituting this value of $\mathbf{W}$ in Eq. (T1.2), we get

$$\begin{aligned} H(W)&=W^{\mathrm{T}}S_BS_T^{+}(S_T-S_B)S_T^{+}S_BW\\ &=W^{\mathrm{T}}S_BS_T^{+}S_TS_T^{+}S_BW-W^{\mathrm{T}}S_B(S_T^{+}S_B)^2W \end{aligned}$$

We have shown in Appendix-B (Lemma A3) that if $G=S_T^{+}S_B$, then $G^2=G$. Using this and the matrix identity $A^{+}AA^{+}=A^{+}$, we get

$$\mathbf{H}(\mathbf{W})=\mathbf{W}^T\mathbf{S}_B\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W}-\mathbf{W}^T\mathbf{S}_B\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W}$$

or $\mathbf{H}(\mathbf{W})=0$

i.e. $\mathbf{W}^{\mathrm{T}}\mathbf{S}_W\mathbf{W}=0$

This concludes the proof of the Theorem.

Theorem 2. If the matrix $\mathbf{W}\in\Re^{d\times h}$ of rank h satisfies the relation $\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W}$, then $\mathbf{W}$ is not in the null space of $\mathbf{S}_B$; i.e., $\mathbf{S}_B\mathbf{W}\neq 0$.

Proof 2. Since $\mathbf{W}\in\Re^{d\times h}$ is a matrix of rank h, it contains h linearly independent vectors; i.e., $\mathbf{W}=[\mathbf{w}_1,\mathbf{w}_2,\ldots\mathbf{w}_h]$ and $\mathbf{w}_i\neq 0$ for $i=1\ldots h$. Since $\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{W}$, it follows $\mathbf{w}_i=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{w}_i$ for $i=1\ldots h$. In order to prove this theorem, we first prove that if $\mathbf{w}_i=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{w}_i$, then $\mathbf{S}_B\mathbf{w}_i\neq 0$. To do this, we use the method of contradiction. Assume that $\mathbf{S}_B\mathbf{w}_i=0$. Then by substituting $\mathbf{S}_B\mathbf{w}_i=0$ in the relation $\mathbf{w}_i=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{w}_i$, we get

$$\begin{aligned} \mathbf{w}_i&=\mathbf{S}_T^{+}(\mathbf{S}_B\mathbf{w}_i)\\ &=\mathbf{S}_T^{+}(0)\\ &=0 \end{aligned}$$

But since $\mathbf{w}_i\neq 0$, so the relation $\mathbf{S}_B\mathbf{w}_i=0$ can not be true. Thus, from contradiction, we have shown that $\mathbf{S}_B\mathbf{w}_i\neq 0$. Since it is true for $i=1\ldots h$, we can say that $\mathbf{S}_B\mathbf{W}\neq 0$.

This concludes the proof of the Theorem.□

2.3.2. Proof of Eq. (7) being a sufficient condition for null LDA method

Here we prove that $\mathbf{W}$ given by Eq. (7) is sufficient condition for the null LDA method; i.e., when $\mathbf{W}$ satisfies Eq. (7), it will also satisfy $\mathbf{S}_W\mathbf{W}=0$ and $\mathbf{S}_B\mathbf{W}\neq 0$. The proof is given below in the form of Theorem 3.

Theorem 3. If the orientation matrix $\mathbf{W}\in\Re^{d\times(c-1)}$ is obtained by using the relation $\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{Y}$ (where $\mathbf{Y}\in\Re^{d\times(c-1)}$ is any random matrix of rank $c-1$), then it satisfies the two criteria of null LDA method (Eqs. (2) and (3)).

Proof 1. It is given that

$$\mathbf{W}=\mathbf{S}_T^{+}\mathbf{S}_B\mathbf{Y} \tag{T3.1}$$

where $\mathbf{Y}\in\Re^{d\times(c-1)}$ is any random matrix of rank $c-1$. Therefore,

$\mathrm{rank}(\mathbf{W})=\min[\mathrm{rank}(\mathbf{S}_T^{+}),\ \mathrm{rank}(\mathbf{S}_B),\mathrm{rank}(\mathbf{Y})]=\min[n-1,c-1,c-1]=c-1.$

Thus, the rank of $\mathbf{W}\in\Re^{d\times(c-1)}$ obtained by Eq. (T3.1) will be $c-1$.

Table 1
Fast implementation of null LDA method.

1. Compute eigenvalues $\mathbf{E}_1 \in \Re^{n\times t}$and eigenvectors $\mathbf{D}_1 \in \Re^{t\times t}$ of $\mathbf{A}^{\mathrm{T}}\mathbf{A} \in \Re^{n\times n}$.
2. Compute transformed matrix $\hat{\mathbf{B}}$ (from eq. 10).
3. Form $t \times (c-1)$ matrix $\hat{\mathbf{Y}}$ randomly (i.e. $\hat{\mathbf{Y}} = \mathrm{rand}(t,c-1)$[a]). Note that the rank of $\hat{\mathbf{Y}}$ should be $c-1$.
4. Compute $\hat{\mathbf{W}} = \mathbf{K}_1\mathbf{K}_2$, where $\mathbf{K}_1 = \mathbf{D}_1^{-1}\hat{\mathbf{B}}$ and $\mathbf{K}_2 = \hat{\mathbf{B}}^{\mathrm{T}}\hat{\mathbf{Y}}$.
5. If orthonormal $\hat{\mathbf{W}}$ is required then $\hat{\mathbf{W}} \leftarrow \mathrm{qr}(\hat{\mathbf{W}})$.
6. Compute $\mathbf{W} = \mathbf{D}_1^{-1/2}\hat{\mathbf{W}}$, then$\mathbf{W} \leftarrow \mathbf{E}_1\mathbf{W}$ and then$\mathbf{W} \leftarrow \mathbf{A}\mathbf{W}$.

Note: Matlab code is available from http://www.hgc.jp/~aloks/
[a] Here function rand is used to generate random numbers uniformly distributed between 0 and 1.

Let us define

$$\mathbf{L}(\mathbf{W}) = \mathbf{S}_T^+\mathbf{S}_B\mathbf{W} \tag{T3.2}$$

Substituting value of $\mathbf{W}$ from Eq. (T3.1) in Eq. (T3.2), we get

$$\mathbf{L}(\mathbf{W}) = \mathbf{S}_T^+\mathbf{S}_B\mathbf{S}_T^+\mathbf{S}_B\mathbf{Y}$$

From Appendix-B (Lemma A3), we substitute $(\mathbf{S}_T^+\mathbf{S}_B)^2 = \mathbf{S}_T^+\mathbf{S}_B$ to get

$$\mathbf{L}(\mathbf{W}) = \mathbf{S}_T^+\mathbf{S}_B\mathbf{Y}$$

Using Eq. (T3.1), this becomes

$$\mathbf{L}(\mathbf{W}) = \mathbf{W}$$

or $\mathbf{S}_T^+\mathbf{S}_B\mathbf{W} = \mathbf{W}$

This equation is same as Eq. (6). Thus, $\mathbf{W} \in \Re^{d\times(c-1)}$ given by Eq. (T3.1) has the following properties: 1) it satisfies Eqs. (6) and (2)) it is of rank $c-1$. From Theorems 1 and 2, we know that $\mathbf{W}$ following these properties satisfies the two criteria of null LDA method. Thus, $\mathbf{W}$ given by Eq. (T3.1) will be a sufficient condition for the null LDA method. This concludes the proof of the Theorem.

3. Implementation of the fast procedure

In the preceding section, we have proposed an alternative null LDA procedure. In this section, we describe its fast implementation. In the proposed null LDA procedure, the orientation matrix $\mathbf{W} \in \Re^{d\times(c-1)}$ is computed by utilizing $\mathbf{W} = \mathbf{S}_T^+\mathbf{S}_B\mathbf{Y}$ (Eq. (7)), where $\mathbf{Y} \in \Re^{d\times(c-1)}$ is any random matrix of rank $c-1$.

In order to compute $\mathbf{W}$ by Eq. (7), we need $\mathbf{S}_T^+$. This requires the EVD of $\mathbf{S}_T = A\mathbf{A}^{\mathrm{T}} \in \Re^{d\times d}$, which is computationally very expensive as d is very large in the SSS problem. A computationally faster way would be to compute the EVD of $\mathbf{A}^{\mathrm{T}}\mathbf{A} \in \Re^{n\times n}$ instead of $\mathbf{S}_T = A\mathbf{A}^{\mathrm{T}} \in \Re^{d\times d}$ [12]. This will reduce the computational complexity significantly to $O(n^3)$. If the eigenvectors and eigenvalues of $\mathbf{A}^{\mathrm{T}}\mathbf{A} \in \Re^{n\times n}$ are $\mathbf{E} \in \Re^{n\times n}$ and $\mathbf{D} \in \Re^{n\times n}$, respectively, then

$$\mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{E}\mathbf{D}\mathbf{E}^{\mathrm{T}}$$
$$= [\mathbf{E}_1,\mathbf{E}_2]\begin{bmatrix}\mathbf{D}_1 & \\ & 0\end{bmatrix}\begin{bmatrix}\mathbf{E}_1^{\mathrm{T}} \\ \mathbf{E}_2^{\mathrm{T}}\end{bmatrix}$$

where

$$\mathbf{E}_1 \in \Re^{n\times t},\mathbf{E}_2 \in \Re^{n\times(n-t)} \text{ and } \mathbf{D}_1 \in \Re^{t\times t}$$
$$= \mathbf{E}_1\mathbf{D}_1\mathbf{E}_1^{\mathrm{T}} \tag{8}$$

and orthonormal eigenvectors $\mathbf{U}_1$ defining the range space of $\mathbf{S}_T$ can be given as

$$\mathbf{U}_1 = \mathbf{A}\mathbf{E}_1\mathbf{D}_1{}^{-1/2}.$$

Since discarding the null space of $\mathbf{S}_T$ does not cause any loss of discriminant information [17], we can use $\mathbf{U}_1 \in \Re^{d\times t}$ to transform the original d-dimensional space to a lower t-dimensional space. The matrices $\mathbf{A}$ and $\mathbf{B}$ can be written in the lower dimensional space as follows:

$$\begin{aligned}\hat{\mathbf{A}} &= \mathbf{U}_1^{\mathrm{T}}\mathbf{A} \in \Re^{t\times n}\\ &= \mathbf{D}_1{}^{-1/2}\mathbf{E}_1^{\mathrm{T}}\mathbf{A}^{\mathrm{T}}\mathbf{A}\\ &= \mathbf{D}_1{}^{-1/2}\mathbf{E}_1^{\mathrm{T}}\mathbf{E}_1\mathbf{D}_1\mathbf{E}_1^{\mathrm{T}} \text{ (from Eq. (8))}\\ &= \mathbf{D}_1{}^{1/2}\mathbf{E}_1^{\mathrm{T}}\end{aligned} \tag{9}$$

and

$$\begin{aligned}\hat{\mathbf{B}} &= \mathbf{U}_1^{\mathrm{T}} \in \Re^{t\times c}\\ &= \mathbf{D}_1{}^{-1/2}\mathbf{E}_1^{\mathrm{T}}(\mathbf{A}^{\mathrm{T}}\mathbf{B})\end{aligned}$$

Computing $\hat{\mathbf{B}}$ using this equation is expensive as d is very large. This computation, however, can be reduced by constructing $\hat{\mathbf{B}}$ from $\hat{\mathbf{A}}$. In order to do this, we first write the transformed matrix $\hat{\mathbf{A}}$ as $\hat{\mathbf{A}} = [\mathbf{v}_1,\mathbf{v}_2,\ldots,\mathbf{v}_n]$ and then compute $\hat{\mathbf{B}}$ as

$$\hat{\mathbf{B}} = \left[\frac{1}{\sqrt{n_1}}\sum_{j=1}^{n_1}\mathbf{v}_j,\frac{1}{\sqrt{n_2}}\sum_{j=n_1+1}^{n_1+n_2}\mathbf{v}_j,\ldots,\frac{1}{\sqrt{n_c}}\sum_{j=n_1+n_2+\cdots+n_{c-1}+1}^{n}\mathbf{v}_j\right] \tag{10}$$

This will give transformed between-class scatter $\hat{\mathbf{S}}_B = \hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}$. From Eq. (9), the transformed total-scatter matrix $\hat{\mathbf{S}}_T = \hat{\mathbf{A}}\hat{\mathbf{A}}^{\mathrm{T}} = \mathbf{D}_1^{1/2}\mathbf{E}_1^{\mathrm{T}}\mathbf{E}_1\mathbf{D}_1^{1/2} = \mathbf{D}_1$. The Eq. (7) can now be used with $\hat{\mathbf{S}}_T$ and $\hat{\mathbf{S}}_B$ to obtain transformation matrix $\hat{\mathbf{W}} \in \Re^{t\times(c-1)}$ for the null LDA method in the lower t-dimensional space as follows:

$$\begin{aligned}\hat{\mathbf{W}} &= \hat{\mathbf{S}}_T^+\hat{\mathbf{S}}_B\hat{\mathbf{Y}}\\ &= \mathbf{D}_1^{-1}\hat{\mathbf{B}}\hat{\mathbf{B}}^{\mathrm{T}}\hat{\mathbf{Y}}\end{aligned} \tag{11}$$

where $\hat{\mathbf{Y}} \in \Re^{t\times(c-1)}$ is a matrix formed from any $t \times (c-1)$ random numbers. Let us define $\mathbf{K}_1 = \mathbf{D}_1^{-1}\hat{\mathbf{B}} \in \Re^{t\times c}$ and $\mathbf{K}_2 = \hat{\mathbf{B}}^{\mathrm{T}}\hat{\mathbf{Y}} \in \Re^{c\times c}$, then $\hat{\mathbf{W}} = \mathbf{K}_1\mathbf{K}_2$.

Note that this $\hat{\mathbf{W}} \in \Re^{t\times(c-1)}$ will not be orthogonal. If needed, it can be made orthogonal by QR decomposition (for $h = c-1$)[2]. Thus, in the proposed null LDA procedure, we transform the d-dimensional space to h-dimensional space using the transformation

$$\mathbf{W} = \mathbf{U}_1\hat{\mathbf{W}} = \mathbf{A}\mathbf{E}_1\mathbf{D}_1^{-1/2}\hat{\mathbf{W}} = \mathbf{A}\mathbf{K}_4\mathbf{K}_3,$$

where $\mathbf{K}_3 = \mathbf{D}_1^{-1/2}\hat{\mathbf{W}}$ and $\mathbf{K}_4 = \mathbf{E}_1\mathbf{K}_3$. The implementation of the proposed fast procedure is summarized in Table 1.

4. Computation complexity and storage requirements

In this section, we discuss the computational complexity and storage requirements of the proposed implementation and

[2] If $1 \le h \le c-1$, then EVD can be applied to select h leading orthonormal vectors of $\hat{\mathbf{W}}^{\mathrm{T}}\hat{\mathbf{S}}_B\hat{\mathbf{W}}$.

Table 2
Computational complexity of the fast implementation procedure.

Steps	Complexities
multiplication of $\mathbf{A}^{\mathrm{T}}\mathbf{A} \in \Re^{n\times n}$	dn^2
computation of $\mathbf{E}_1 \in \Re^{n\times t}$ and $\mathbf{D}_1 \in \Re^{t\times t}$ using eigenvalue decomposition of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$	$17n^3$
Computation of transformed matrix $\hat{\mathbf{B}}$ (from eq. 10)	n^2
computation of $\mathbf{K}_1$ and $\mathbf{K}_2$	$tc+2tc(c-1)$
computation of $\hat{\mathbf{W}}=\mathbf{K}_1\mathbf{K}_2$	$2tc(c-1)$
Orthogonalization of $\hat{\mathbf{W}} \in \Re^{t\times(c-1)}$ (if QR decomposition is used)	$4tc^2-4c^3/3$
multiplication of $\mathbf{W}=\mathbf{D}_1^{-1/2}\hat{\mathbf{W}}$, $\mathbf{W}\leftarrow\mathbf{E}_1\mathbf{W}$ and $\mathbf{W}\leftarrow\mathbf{A}\mathbf{W}$	$t(c-1)+2nt(c-1)+2dn(c-1)$
Total estimated	$dn^2+2dnc+17n^3+2n^2c+8nc^2+2nc-\frac{4}{3}c^3$ (since $t\approx n$ and $c-1\approx c$)

Table 3
Computational complexities of different implementation of the null space LDA method.

Implementations	**Computational complexity (for $d \gg n$ and $n > c$)**
Null LDA	$4d^2n$
PCA+null LDA	$16dn^2+4dnc$
OLDA	$14dn^2+4dnc+4dc^2$
QR–NLDA [7]	$4dn^2+2dnc$
Proposed implementation of null LDA	dn^2+2dnc

Table 4
Storage requirements of different implementation of the null space LDA method.

Implementations	**Storage**
Null LDA	dh (where $1\le h\le c-1$)
PCA+null LDA	dh (where $1\le h\le c-1$)
OLDA	$d(c-1)$
QR–NLDA	$d(c-1)$
Proposed implementation of null LDA	dh (where $1\le h\le c-1$)

compare it with other implementations of null LDA method. The computational complexities of the major steps of the proposed implementation are listed in Table 2 (see Appendix-C for computational complexities of some major operations).

In a typical SSS problem, where the dimensionality d is very large compared to the number of training vectors (i.e., $d \gg n$), the computational complexity of the proposed implementation is dn^2+2dnc flops (which is mainly due to the multiplication of matrices). The computational complexity can be further reduced by using efficient matrix multiplication algorithms (see Appendix-C). In this case the computational complexity will be $O(dn^{1.376})+2dnc$.

In the null LDA method [6], the computation of the null space of $\mathbf{S}_W$ is required. This can be achieved by doing singular value decomposition of $A \in \Re^{d\times n}$ for computing U and Σ_1 matrices. This step involves $4d^2n-8dn^2$ computations [13], which is very expensive. The PCA+null LDA method requires approximately $16dn^2+4dnc$ computations. In the OLDA method [44], the singular value decomposition carried out at two steps approximately requires $14dn^2-2n^3$ and $14nc^2-2c^3$ computations [13], followed by QR decomposition at one step which requires approximately $4dc^2-4c^3/3$ flops [13]. In addition, matrix multiplication requires approximately $4dnc$ computations. The QR–NLDA method [7] requires approximately $4dn^2+2dnc$ computations. The computational complexities of different implementations are listed in Table 3.

It can be observed from Table 3 that the computational complexity of the proposed implementation is much lower than the other implementations. This computational complexity can be reduced further to $O(dn^{1.376})+2dnc$. The storage requirements of different implementations are listed in Table 4. In all the cases, the orientation matrix $\mathbf{W} \in \Re^{d\times h}$ computed during training session is required to be stored for the testing session.

In summary, the computational complexity of the proposed implementation is lower than that of the other implementations. This is experimentally demonstrated in the next section.

5. Datasets and experimentation

Three types of datasets are utilized for the experimentation. These are DNA microarray gene expression data, face recognition data and text classification data. In addition to this, we use randomly generated data to investigate the effect of dimensionality d on the computation time of different implementations. 5 DNA microarray gene expression datasets[3] are utilized in this work to show the effectiveness of the proposed method. For face recognition, two commonly known datasets, namely ORL database [33] and AR database [29], are utilized for the experimentation. The ORL database contains 400 images of 40 persons (with 10 images per person). The dimensionality d of the feature space is 10,304. A subset of AR database is used here with 1400 face images from 100 persons (14 images per person). The dimensionality d is 4980. We use a subset of Dexter dataset [4] for text classification in a bag-of-word representation. This dataset has sparse continuous input variables. The description of all the datasets is given in Table 5. It can be seen from this table that the dimensionality d for each dataset is very large compared to the number of training samples. This leads to the SSS problem.

The null LDA method is used for dimensionality reduction and the nearest neighbor classifier is used for classifying the test data. As expected, the classification accuracies of PCA+null LDA, OLDA, QR–NLDA and the proposed implementation are found to be identical. However, they significantly differ in terms of their computation times as shown in Table 6.1. Here, we measure the computation time of a given implementation as the CPU time taken by its ‘Matlab’ program on a Dell computer (Optiplex 755, Core 2 Quad, 2.4 GHz). We can observe from Table 6.1 that the proposed implementation of the null LDA method requires lowest computation time. For completeness, we list the classification accuracies of all these null LDA algorithms (PCA+null LDA, OLDA, QR–NLDA and the proposed implementation) using N-fold cross validation (where $N=3$) in Table 6.2.

To investigate the computation time as a function of dimensionality, we generate random data for 100 classes with 5 training vectors per class. Therefore, the total number of training vectors is

[3] Most of the DNA microarray gene expression datasets can be downloaded from http://sdmc.lit.org.sg/GEDatasets/Datasets.html or http://cs1.shu.edu.cn/gzli/data/mirror-kentridge.html or http://leo.ugr.es/elvira/DBCRepository.

Table 5
Datasets used in the experimentation.

Datasets	Class	Dimension	Number of training samples	Number of testing samples
ALL subtype [46]	7	12,558	215	112
GCM [31]	14	16,063	144	54
Prostate Tumor [36]	2	12,600	102	34
SRBCT [21]	4	2,308	63	20
MLL [1]	3	12,582	57	15
Face ORL [33]	40	10,304	200	200
Face AR [29]	100	4,980	700	700
Dexter [4]	2	20,000	300	300

Table 6.1
Computation time of different implementations on the microarray gene expression, face recognition and text classification datasets.

Database	PCA+Null LDA CPU Time	OLDA CPU Time	QR–NLDA CPU time	Proposed implementation of null LDA CPU Time
ALL subtype	4.43	3.94	2.57	0.76
GCM	3.84	3.77	1.91	0.44
Prostate Tumor	1.51	1.44	0.89	0.23
SRBCT	0.18	0.17	0.08	0.03
MLL	0.72	0.72	0.40	0.13
Face ORL	5.10	5.12	1.81	0.59
Face AR	20.11	16.99	7.87	4.21
Dexter	8.80	7.81	5.32	1.52

Table 6.2
A comparative table showing the classification accuracy for different databases with other null LDA based algorithms (using *N*-fold cross validation, where *N*=3).

Database	PCA+Null LDA accuracy (%)	OLDA accuracy (%)	QR–NLDA accuracy (%)	Proposed implementation of null LDA accuracy (%)
ALL subtype	90.3	90.3	90.3	90.3
GCM	72.7	72.7	72.7	72.7
Prostate Tumor	88.6	88.6	88.6	88.6
SRBCT	100	100	100	100
MLL	95.7	95.7	95.7	95.7
Face ORL	96.9	96.9	96.9	96.9
Face AR	95.7	95.7	95.7	95.7
Dexter	94.5	94.5	94.5	94.5

500. We vary the dimensionality d from 5000 to 70,000 and measure the computation times of these implementations. Fig. 1 shows the computation time as a function of dimensionality. It can be seen from this figure that the computation time of OLDA is similar to the processing time of PCA+Null LDA. It can also be seen that as the dimensionality becomes large, QR–NLDA becomes computationally more efficient than OLDA and PCA+Null LDA, and the proposed implementation is the fastest.

We also show the comparative performance in terms of classification accuracy of the null LDA algorithms (PCA+Null LDA, OLDA, QR–NLDA, proposed null LDA) with the following algorithms: pseudoinverse technique (PILDA) [40], direct LDA (DLDA) technique [47], regularized LDA (RLDA) technique [11,14], Fisherface LDA [37,2] technique, principal component analysis (PCA) [12] and eigenfeature regularization (EFR) technique [18]. All the techniques (except PCA) are used to reduce the dimensionality to $c-1$ (since rank of S_B is $c-1$), where c is the number of classes. For PCA, the dimensionality is reduced to $n-1$ (since rank of covariance matrix is $n-1$), where n is the number of training samples. After dimensionality reduction, the nearest neighbor classifier (NNC) using Euclidean distance measure is used for classifying a test feature vector. The training set and test set merged in a set of samples and *N*-fold cross-validation is performed (where $N=3$) to evaluate the classification accuracy on all the datasets using the above mentioned techniques. The comparison has been depicted in Table 7. It can be observed from the table that the null LDA algorithms performs comparably well with other existing methods.

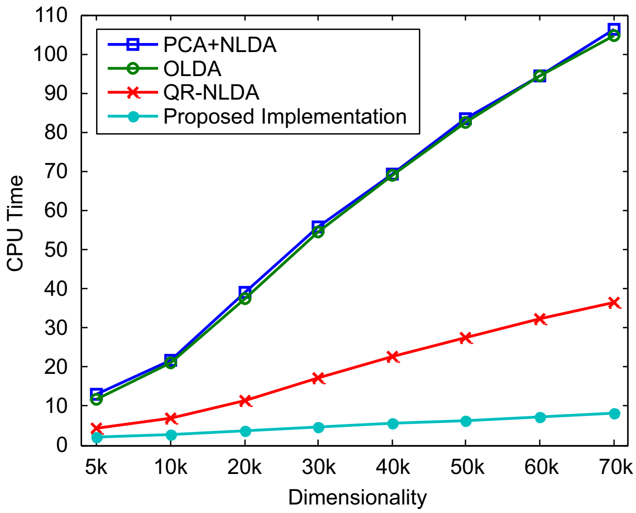

Fig. 1. Computation time as a function of dimensionality on randomly-generated data (where $c=100$ and $n=500$) using PCA+null LDA, OLDA, QR–NLDA and proposed implementation of null LDA method.

Table 7
A comparison of classification accuracy of the null LDA algorithms with other existing methods (using *N*-fold cross-validation, where *N*=3).

Database	RLDA	PILDA	DLDA	Fisherface	EFR	PCA	NLDA
ALL subtype	86.0	80.1	78.2	88.5	90.0	57.0	90.3
GCM	76.5	60.1	62.8	70.0	74.9	55.7	72.7
Prostate Tumor	81.8	76.5	73.5	88.6	82.6	62.1	88.6
SRBCT	93.6	68.0	84.6	100	100	73.1	100
MLL	94.2	87.0	91.3	95.7	95.7	91.3	95.7
Face ORL	96.4	96.7	97.2	92.5	96.7	95.8	96.9
Face AR	96.3	97.3	96.3	94.9	97.3	78.3	95.7
Dexter	94.7	73.8	91.2	94.5	94.7	85.7	94.5
average	**89.9**	**79.9**	**84.4**	**90.6**	**91.5**	**74.9**	**91.8**

6. Conclusion

In this paper, we have theoretically derived an alternative null LDA method and proposed a procedure for its fast implementation. The proposed implementation is shown to be computationally faster than the existing implementations of the null LDA method. This computational advantage is achieved without any degradation in classification performance.

Acknowledgment

The authors gratefully acknowledge helpful consultations with Prof. Gilbert Strang of Massachusetts Institute of Technology, USA.

Appendix A

Theorem 1. *If the matrix $\mathbf{W} \in \Re^{d\times h}$ satisfies the relation $\mathbf{W} = S_T^{-1} S_B \mathbf{W}$, then it is in the null space of ; i.e., $\mathbf{S}_W\mathbf{W} = 0$.*

Proof 1. It is given that

$$\mathbf{W} = \mathbf{S}_T^{-1}\mathbf{S}_B\mathbf{W}$$

Pre-multiplying both sides of this equation by $\mathbf{S}_T$, we get

$$\mathbf{S}_T\mathbf{W} = \mathbf{S}_B\mathbf{W}$$

or $(\mathbf{S}_T - \mathbf{S}_B)\mathbf{W} = 0$

Substituting $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$, we get

$$\mathbf{S}_W\mathbf{W} = 0$$

This concludes the proof of this Theorem.

Theorem 2. If the matrix $\mathbf{W} \in \Re^{d\times h}$ of rank h satisfies the relation $\mathbf{W} = S_T^{-1} S_B \mathbf{W}$, then $\mathbf{W}$ is not in the null space of ; i.e., $S_B\mathbf{W} \neq 0$.

Proof 2. Since $\mathbf{W} \in \Re^{d\times h}$ is a matrix of rank h, it contains h linearly independent vectors; i.e., $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \ldots \mathbf{w}_h]$ and $\mathbf{w}_i \neq 0$ for $i = 1\ldots h$. Since $\mathbf{W} = \mathbf{S}_T^{-1}\mathbf{S}_B\mathbf{W}$, it follows $\mathbf{w}_i = \mathbf{S}_T^{-1}\mathbf{S}_B\mathbf{w}_i$ for $i = 1\ldots h$. In order to prove this theorem, we first prove that if $\mathbf{w}_i = \mathbf{S}_T^{-1}\mathbf{S}_B\mathbf{w}_i$, then $\mathbf{S}_B w_i \neq 0$. To do this, we use the method of contradiction. Assume that $\mathbf{S}_B\mathbf{w}_i = 0$. Then by substituting $\mathbf{S}_B\mathbf{w}_i = 0$ in the relation $\mathbf{w}_i = \mathbf{S}_T^{-1}\mathbf{S}_B\mathbf{w}_i$, we get

$$\begin{aligned}\mathbf{W} &= \mathbf{S}_T^{-1}(\mathbf{S}_B\mathbf{W})\\ &= \mathbf{S}_T^{-1}(0)\\ &= 0\end{aligned}$$

But since $\mathbf{w}_i \neq 0$, so the relation $\mathbf{S}_B\mathbf{w}_i = 0$ can not be true. Thus, from contradiction, we have shown that $\mathbf{S}_B\mathbf{w}_i \neq 0$. Since it is true for $i = 1\ldots h$, we can say that $\mathbf{S}_B\mathbf{W} \neq 0$.

This concludes the Proof of the Theorem.

Appendix B

Lemma A1. If $\mathbf{I}_t = \boldsymbol{\Sigma}_B + \boldsymbol{\Sigma}_W$ (where $\mathbf{I}_t \in \Re^{t\times t}$ is an identity matrix of rank t, and $\boldsymbol{\Sigma}_B \in \Re^{t\times t}$ and $\boldsymbol{\Sigma}_W \in \Re^{t\times t}$ are diagonal matrices of ranks b and $t-b$, respectively), then b diagonal elements of matrix $\boldsymbol{\Sigma}_B$ will be unity and the remaining $t-b$ diagonal elements will be zero.

Proof A1. It is given that $\mathbf{I}_t = \boldsymbol{\Sigma}_B + \boldsymbol{\Sigma}_W$. Therefore, $\boldsymbol{\Sigma}_W$can be written as

$$\boldsymbol{\Sigma}_W = \mathbf{I}_t - \boldsymbol{\Sigma}_B \tag{AL1.1}$$

Since $\boldsymbol{\Sigma}_B$ is diagonal matrix of rank b, it can be written as

$$\boldsymbol{\Sigma}_B = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_b, \underbrace{0, 0, \ldots 0}_{t-b\text{zeros}}) \tag{AL1.2}$$

where $\lambda_j \neq 0 \; \forall \, j = 1\ldots b$. Substituting $\boldsymbol{\Sigma}_B$ in Eq. (AL1.1), we get

$$\boldsymbol{\Sigma}_W = \mathrm{diag}(1-\lambda_1, 1-\lambda_2, \ldots, 1-\lambda_b, \underbrace{1, 1, \ldots 1}_{t-b\text{ones}})$$

The rank of matrix $\boldsymbol{\Sigma}_W$ is $t-b$. This is possible only when $1-\lambda_j = 0 \forall j = 1\ldots b$, or $\lambda_j = 1 \forall j = 1\ldots b$. Substituting these values of λ_j in equation AL1.2, we get

$\boldsymbol{\Sigma}_B = \begin{bmatrix}\mathbf{I}_b & 0\\ 0 & 0\end{bmatrix}$, where$\mathbf{I}_b \in \Re^{b\times b}$is an identity matrix.

This concludes the proof of the Lemma.

Lemma A2. *Let $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$, where $\mathbf{S}_T = \mathbf{A}\mathbf{A}^{\mathrm{T}}$, $A \in \Re^{d\times n}$, $\mathbf{S}_B = \mathbf{B}\mathbf{B}^{\mathrm{T}}$, $\mathbf{B} \in \Re^{d\times c}$and $S_W \in \Re^{d\times d}$ with rank$(\mathbf{S}_T) = t = n-1$ (where $t < d$), rank$(\mathbf{S}_B) = b = c-1$(where $b < t$) and rank$(\mathbf{S}_W) = n-c$. Let $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2]$ be the matrix consisting of eigenvectors of $\mathbf{A}$ where $\mathbf{U}_1 \in \Re^{d\times t}$ corresponds to the range space of $\mathbf{S}_T$ and $\mathbf{U}_2 \in \Re^{d\times(d-t)}$ corresponds to the null space of $\mathbf{S}_T$. If a rectangular matrix $\mathbf{Q} \in \Re^{t\times c}$ is defined such that $\mathbf{Q} = \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{B}$ (where $\boldsymbol{\Sigma}_1 \in \Re^{t\times t}$ is a diagonal matrix of square root of eigenvalues of $\mathbf{S}_T$) and if eigenvalue decomposition of $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ is $R\Lambda\mathbf{R}^{\mathrm{T}}$ (where $\mathbf{R} \in \Re^{t\times t}$ is an orthogonal matrix and $\boldsymbol{\Lambda} \in \Re^{t\times t}$ is a diagonal matrix), then $\boldsymbol{\Lambda} = \begin{bmatrix}\mathbf{I}_b & 0\\ 0 & 0\end{bmatrix}$,* where $\mathbf{I}_b \in \Re^{b\times b}$ is an identity matrix.

Proof A2. Note that the proof given here is an extension of the proof provided by Ye [44]. The singular value decomposition of $\mathbf{S}_T$can be given by

$\mathbf{S}_T = \mathbf{U}\begin{bmatrix}\boldsymbol{\Sigma}_1^2 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$, where $\mathbf{U} \in \Re^{d\times d}$is an orthogonal matrix.

or $\begin{bmatrix}\boldsymbol{\Sigma}_1^2 & 0\\ 0 & 0\end{bmatrix} = \mathbf{U}^{\mathrm{T}}\mathbf{S}_T\mathbf{U}$. Substituting $\mathbf{S}_T = \mathbf{S}_B + \mathbf{S}_W$, we get

$$\begin{bmatrix}\boldsymbol{\Sigma}_1^2 & 0\\ 0 & 0\end{bmatrix} = \mathbf{U}^{\mathrm{T}}\mathbf{S}_B\mathbf{U} + \mathbf{U}^{\mathrm{T}}\mathbf{S}_W\mathbf{U} \tag{AL2.1}$$

Substituting $\mathbf{U} = [\mathbf{U}_1, \mathbf{U}_2]$, this equation becomes

$$\begin{bmatrix}\boldsymbol{\Sigma}_1^2 & 0\\ 0 & 0\end{bmatrix} = \begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1 & \mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_2\\ \mathbf{U}_2^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1 & \mathbf{U}_2^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_2\end{bmatrix} + \begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1 & \mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_2\\ \mathbf{U}_2^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1 & \mathbf{U}_2^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_2\end{bmatrix} \tag{AL2.2}$$

Since the two matrices on the right hand side of Eq. (AL2.2) are positive semidefinite, we have $\mathbf{U}_2^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_2 = 0$, $\mathbf{U}_2^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_2 = 0$, $\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_2 = 0$ and $\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_2 = 0$.

Therefore from Eqs. (AL2.1) and (AL2.2) we get

$$\mathbf{U}^{\mathrm{T}}\mathbf{S}_B\mathbf{U} = \begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1 & 0\\ 0 & 0\end{bmatrix} \tag{AL2.3}$$

and

$$\mathbf{U}^{\mathrm{T}}\mathbf{S}_W\mathbf{U} = \begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1 & 0\\ 0 & 0\end{bmatrix} \tag{AL2.4}$$

Substituting Eqs. (AL2.3) and (AL2.4) in Eq. (AL2.1), we get

$$\boldsymbol{\Sigma}_1^2 = \mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1 + \mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1$$

multiplying both sides of this equation by $\boldsymbol{\Sigma}_1^{-1}$ from left as well as from right, we get

$$\mathbf{I}_t = \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\boldsymbol{\Sigma}_1^{-1},$$

where $\mathbf{I}_t \in \Re^{t\times t}$ is an identity matrix. Using $\mathbf{S}_B = \mathbf{B}\mathbf{B}^{\mathrm{T}}$and $\mathbf{Q} = \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{B}$, we get

$$\mathbf{I}_t = \mathbf{Q}\mathbf{Q}^{\mathrm{T}} + \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\boldsymbol{\Sigma}_1^{-1} \tag{AL2.5}$$

Since rank$(\mathbf{S}_B) = c-1$ is less than the ranks of $\boldsymbol{\Sigma}_1^{-1}$and $\mathbf{U}_1$, the rank of the matrix $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$will be $c-1$. The EVD of $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ is $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = R\Lambda\mathbf{R}^{\mathrm{T}}$(where $\mathbf{R} \in \Re^{t\times t}$ is an orthogonal matrix and $\boldsymbol{\Lambda} \in \Re^{t\times t}$ is a diagonal matrix of rank $c-1$). Substituting $\mathbf{Q}\mathbf{Q}^{\mathrm{T}} = R\Lambda\mathbf{R}^{\mathrm{T}}$ in Eq. (AL2.5), we get

$$I_t = R\Lambda R^{\mathrm{T}} + \boldsymbol{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\boldsymbol{\Sigma}_1^{-1}$$

Multiplying both the sides of this equation by $\mathbf{R}^{\mathrm{T}}$ from the left and $\mathbf{R}$ from the right, we get

$$\mathbf{I}_t=\mathbf{\Lambda}+\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}S_W\mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{R}, \tag{AL2.6}$$

or $\mathbf{I}_t-\mathbf{\Lambda}=\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{R}$.

Since the left hand side of this equation is diagonal, the right hand side will also be diagonal. Since $\mathrm{rank}(\mathbf{S}_W)=n-c$ is lower than the ranks of $\mathbf{U}_1$, $\mathbf{\Sigma}_1^{-1}$ and R, the rank of the right hand side will also be $n-c$; i.e., $\mathrm{rank}(\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{R})=n-c$. In addition, the ranks of $\mathbf{I}_t$ and $\mathbf{\Lambda}$ are $t=n-1$and $b=c-1$, respectively. Thus,

$$\mathrm{rank}(\mathbf{I}_t)=\mathrm{rank}(\mathbf{\Lambda})+\mathrm{rank}(\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_W\mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{R}) \tag{AL2.7}$$

Using Lemma A1 and equations AL2.6 and AL2.7, we can deduce that $\mathbf{\Lambda}=\begin{bmatrix}\mathbf{I}_b & 0\\ 0 & 0\end{bmatrix}$, where $\mathbf{I}_b\in\Re^{b\times b}$ is an identity matrix.

This concludes the proof of the Lemma.

Lemma A3. If $\mathbf{G}=\mathbf{S}_T^{+}\mathbf{S}_B$ (where $\mathbf{S}_T^{+}$ is the pseudo inverse of the total scatter matrix $\mathbf{S}_T$ and $\mathbf{S}_B$ is the between class scatter matrix), then it satisfies the relation $\mathbf{G}^2=\mathbf{G}$.

Proof 1. Since $\mathbf{S}_T\in\Re^{d\times d}$ is of rank $t=n-1$, its eigenvalue decomposition (EVD) can be given by,

$$\mathbf{S}_T=\mathbf{U}\Sigma^2\mathbf{U}^{\mathrm{T}}=[\mathbf{U}_1,\mathbf{U}_2]\begin{bmatrix}\mathbf{\Sigma}_1^2 & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\\ \mathbf{U}_2^{\mathrm{T}}\end{bmatrix}, \tag{AL3.1}$$

where $\mathbf{U}\in\Re^{d\times d}$ is an orthogonal matrix with partitions $\mathbf{U}_1\in\Re^{d\times t}$ and $\mathbf{U}_2\in\Re^{d\times(d-t)}$, where $\mathbf{U}_1$ corresponds to the range space of $\mathbf{S}_T$ and $\mathbf{U}_2$ corresponds to the null space of $\mathbf{S}_T$, and $\mathbf{\Sigma}_1\in\Re^{t\times t}$ is a diagonal matrix. The pseudoinverse of $\mathbf{S}_T$ is given by,

$$\mathbf{S}_T^{+}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2} & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

It is given that $\mathbf{G}=\mathbf{S}_T^{+}\mathbf{S}_B$, or

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2} & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}\mathbf{S}_B \tag{AL3.2}$$

since $\mathbf{U}\mathbf{U}^{\mathrm{T}}=\mathbf{U}^{\mathrm{T}}\mathbf{U}=\mathbf{I}_{d\times d}$, Eq. (AL3.2) can be written as

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2} & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}\mathbf{S}_B\mathbf{U}\mathbf{U}^{\mathrm{T}}$$

From Eq. (AL2.3) of Lemma A2, it follows

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\mathbf{U}_1^{\mathrm{T}}\mathbf{S}_B\mathbf{U}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

Since $\mathbf{S}_B=\mathbf{B}\mathbf{B}^{\mathrm{T}}$, it follows

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2}\mathbf{U}_1^{\mathrm{T}}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{U}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

or $\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-2}\mathbf{U}_1^{\mathrm{T}}\mathbf{B}\mathbf{B}^{\mathrm{T}}\mathbf{U}_1\mathbf{\Sigma}_1^{-1}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$

Let $\mathbf{Q}=\mathbf{\Sigma}_1^{-1}\mathbf{U}_1^{\mathrm{T}}\mathbf{B}$, then

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}\mathbf{Q}\mathbf{Q}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

If the EVD of $\mathbf{Q}\mathbf{Q}^{\mathrm{T}}$ is $R\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}$ (where $\mathbf{R}\in\Re^{t\times t}$ is the orthogonal matrix and $\mathbf{\Lambda}\in\Re^{t\times t}$ is diagonal matrix), then $\mathbf{G}$ can be written as

$$\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}R\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}} \tag{AL3.3}$$

From this, $\mathbf{G}^2$ is given by

$$\mathbf{G}^2=\mathbf{G}\mathbf{G}=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}R\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}R\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

or

$$\mathbf{G}^2=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}\mathbf{R}\mathbf{\Lambda}^2\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}} \tag{AL3.4}$$

Lemma A2 shows that the diagonal matrix $\mathbf{\Lambda}\in\Re^{t\times t}$ is given by,

$$\mathbf{\Lambda}=\begin{bmatrix}\mathbf{I}_b & 0\\ 0 & 0\end{bmatrix},$$

where $\mathbf{I}_b$ is an identity matrix of rank $b=c-1$. Therefore, $\mathbf{\Lambda}^2=\mathbf{\Lambda}$. Substituting this in Eq. (AL3.4), we get

$$\mathbf{G}^2=\mathbf{U}\begin{bmatrix}\mathbf{\Sigma}_1^{-1}\mathbf{R}\mathbf{\Lambda}\mathbf{R}^{\mathrm{T}}\mathbf{\Sigma}_1 & 0\\ 0 & 0\end{bmatrix}\mathbf{U}^{\mathrm{T}}$$

Using Eq. (AL3.3), this can be written as

$$\mathbf{G}^2=\mathbf{G}$$

This concludes the proof of the Lemma.

Appendix C

Computational complexities:

i. Matrix multiplication of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ (where $\mathbf{A}\in\Re^{d\times n}$) will require dn^2 computations. This computation can be, however, reduced by splitting matrix $\mathbf{A}$ into d/n square blocks and since the square matrix multiplication has the computational complexity of $O(n^{2.376})$ [9], the block computation of $\mathbf{A}^{\mathrm{T}}\mathbf{A}$ will require approximately $O(dn^{1.376})+\frac{1}{2}(dn-n^2)$ computations.

ii. The multiplication of two rectangular matrices of sizes $p\times q$ and $q\times r$ will require $2pqr$ computations [13].

iii. Singular value decomposition of a matrix $\mathbf{G}\in\Re^{p\times q}$ (where $p>q$) to get diagonal matrix $\mathbf{\Sigma}\in\Re^{t\times t}$ and eigenvectors $\mathbf{U}_1\in\Re^{p\times t}$ (where$t=q-1=\mathrm{rank}(\mathbf{G})$) will require approximately $14pq^2-2q^3$ computations [13]. If $\mathbf{U}\in\Re^{p\times q}$ is required, then computational complexity will be $4p^2q-8pq^2$ flops.

iv. The QR decomposition of a matrix $\mathbf{G}\in\Re^{p\times q}$ (where $p>q$) to get $\mathbf{Q}_1\in\Re^{p\times t}$ (where$t=q-1=\mathrm{rank}(\mathbf{G})$) will require approximately $4pq^2-4q^3/3$ computations [13].

References

[1] S.A. Armstrong, J.E. Staunton, L.B. Silverman, R. Pieters, M.L. den Boer, M.D. Minden, S.E. Sallan, E.S. Lander, T.R. Golub, S.J. Korsemeyer, MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia, Nature Genetics 30 (2002) 41–47. [Data Source1:], [Data Source2:]〈http://sdmc.lit.org.sg/GEDatasets/Datasets.html〉 〈http://www.broad.mit.edu/cgi-bin/cancer/publications/pub_paper.cgi?mode=view&paper_id=63〉.

[2] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 711–720.

[4] C.L. Blake and C.J. Merz, UCI repository of machine learning databases, ⟨http://www.ics.uci.edu/∼mlearn⟩, Irvine, CA, University of Calif., Dept. of Information and Comp. Sci., 1998.

[5] H. Cevikalp, M. Neamtu, M. Wilkes, A. Barkana, Discriminative common vectors for face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 27 (1) (2005) 4–13.

[6] L.-F. Chen, H.-Y.M. Liao, M.-T. Ko, J.-C. Lin, G.-J. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (2000) 1713–1726.

[7] D. Chu, G.S. Thye, A new and fast implementation for null space based linear discriminant analysis, Pattern Recognition 43 (2010) 1373–1379.

[9] D. Coppersmith, S. Winograd, Matrix multiplication via arithmetic progressions, Journal of Symbolic Computation 9 (3) (1990) 251–280.

[10] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, Wiley, New York, 1973.

[11] J.H. Friedman, Regularized discriminant analysis, Journal of the American Statistical Association 84 (1989) 165–175.

[12] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press Inc., Hartcourt Brace Jovanovich, Publishers, San Diego, CA 92101-4495, USA, 1990.

[13] G.H. Golub, C.F.V. Loan, Matrix Computations, The John Hopkins University Press, Baltimore, Maryland 21218-4363, USA, 1996.

[14] Y. Guo, T. Hastie, R. Tinshirani, Regularized discriminant analysis and its application in microarrays, Biostatistics 8 (1) (2007) 86–100.

[15] O.C. Hamsici, A.M. Martinez, Bayes optimality in linear discriminant analysis, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (2008) 647–657.

[16] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 995–1006.

[17] R. Huang, Q. Liu, H. Lu, S. Ma, Solving the small sample size problem of LDA, Proceedings of ICPR 3 (2002) 29–32. 2002.

[18] X. Jiang, B. Mandal, A. Kot, Eigenfeature regularization and extraction in face recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 30 (3) (2008) 383–394.

[19] X. Jiang, Asymmetric principal component analysis and discriminant analysis for pattern classification, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (5) (2009) 931–937.

[20] X. Jiang, Linear subspace learning-based dimensionality reduction, IEEE Signal Processing Magazine 28 (2) (2011) 16–26.

[21] J. Khan, J.S. Wei, M. Ringner, L.H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C.R. Antonescu, C. Peterson, P.S. Meltzer, Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural network, Nature Medicine 7 (2001) 673–679. [Data Source: ⟨http://research.nhgri.nih.gov/microarray/Supplement/⟩].

[22] Z. Jin, J.Y. Yang, Z.M. Tang, Z.S. Hu, A theorem on the uncorrelated optimal discriminant vectors, Pattern Recognition 24 (10) (2001) 2041–2047.

[23] W.J. Krzanowski, P. Jonathan, W.V. McCarthy, M.R. Thomas, Discriminant analysis with singular covariance matrices: methods and applications to spectroscopic data, Applied Statistics 44 (1995) 101–115.

[25] J. Liu, S. Chen, X. Tan, D. Zhang, Efficient pseudoinverse linear discriminant analysis and its non linear form for face recognition, International Journal of Pattern Recognition and Artificial Intelligence 21 (2007) 1265–1278.

[26] W. Liu, Y. Wang, S.Z. Li and T. Tan, Null Space Approach of Fisher Discriminant Analysis for Face Recognition, ECCV Biometric Authentication Workshop, Prague, Czech, 2004.

[27] R. Lotlikar, R. Kothari, Fractional-step dimensionality reduction, IEEE Trans-Pattern Analysis and Machine Intelligence 22 (6) (2000) 623–627.

[28] J. Lu, K.N. Plataniotis, A.N. Venetsanopoulos, Face recognition using LDA-based algorithms, IEEE Transactions on Neural Networks. 14 (1) (2003) 195–200.

[29] A.M. Martinez, Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class, IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (6) (2002) 748–763.

[30] C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition 41 (2008) 1083–1097.

[31] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C.-H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J.P. Mesirov, T. Poggio, W. Gerald, M. Loda, E.S. Lander, T.R. Golub, Multiclass cancer diagnosis using tumor gene expression signatures, Proceedings of the National Academy of Sciences of the USA 98 (26) (2001) 15149–15154.

[32] S. Raudys, R.P.W. Duin, On expected classification error of the Fisher linear classifier with pseudo-inverse covariance matrix, Pattern Recognition Letters 19 (5–6) (1998) 385–392.

[33] F. Samaria and A. Harter, Parameterization of a stochastic model for human face identification, Proc. Second IEEE Workshop Applications of Comp. Vision, pp. 138–142, 1994.

[34] A. Sharma, K.K. Paliwal, Cancer classification by gradient LDA technique using microarray gene expression data, Data & Knowledge Engineering 66 (2008) 338–347.

[35] A. Sharma, K.K. Paliwal, A gradient linear discriminant analysis for small sample sized problem, Neural Processing Letters 27 (2008) 17–24.

[36] D. Singh, P.G. Febbo, K. Ross, D.G. Jackson, J. Manola, C. Ladd, P. Tamayo, A.A. Renshaw, A.V. D'Amico, J.P. Richie, E.S. Lander, M. Loda, P.W. Kantoff, T.R. Golub, W.R. Sellers, Gene expression correlates of clinical prostate cancer behavior, Cancer Cell 1 (2002) 203–209. [Data Source:]⟨http://sdmc.lit.org.sg/GEDatasets/Datasets.html#Prostate⟩.

[37] D.L. Swets, J. Weng, Using discriminative eigenfeatures for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 831–836.

[40] Q. Tian, M. Barbero, Z.H. Gu, S.H. Lee, Image classification by the Foley-Sammon transform, Optical Engineering 25 (7) (1986) 834–840.

[43] J. Yang, D. Zhang, J.-Y. Yang, A generased *K–L* expansion method which can deal with small samples size and high-dimensional problems, Pattern Analysis and Applications 6 (2003) 47–54.

[44] J. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, Journal of Machine Learning Research 6 (2005) 483–502.

[45] J. Ye, T. Xiong, Computational and theoretical analysis of null space and orthogonal linear discriminant analysis, Journal of Machine Learning Research 7 (2006) 1183–1204.

[46] E.J. Yeoh, M.E. Ross, S.A. Shurtleff, W.K. Williams, D. Patel, R. Mahfouz, F.G. Behm, S.C. Raimondi, M.V. Relling, A. Patel, C. Cheng, D. Campana, D. Wilkins, X. Zhou, J. Li, H. Liu, C.H. Pui, W.E. Evans, C. Naeve, L. Wong, J.R. Downing, Classification, subtype discovery, and prediction of outcome in pediatric acute lymphoblastic leukemia by gene expression profiling, Cancer 1 (2) (2002) 133–143. Data Source: ⟨http://www.stjuderesearch.org/data/ALL1/⟩].

[47] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data-with application to face recognition, Pattern Recognition 34 (2001) 2067–2070.

Alok Sharma received the BTech degree from the University of the South Pacific (USP), Suva, Fiji, in 2000 and the MEng degree, with an academic excellence award, and the PhD degree in the area of pattern recognition from Griffith University, Brisbane, Australia, in 2001 and 2006, respectively. He is currently a research fellow at the University of Tokyo. He is also with the Signal Processing Laboratory, Griffith University and the University of the South Pacific. He participated in various projects carried out in conjunction with Motorola (Sydney), Auslog Pty. Ltd. (Brisbane), CRC Micro Technology (Brisbane), and the French Embassy (Suva). He is nominated by NSERC, Canada in Visiting Fellowship program, 2009. His research interests include pattern recognition, computer security, and human cancer classification. He reviewed several articles from journals like IEEE Transactions on Neural Networks, IEEE Transaction on Systems, Man, and Cybernetics, Part A: Systems and Humans, IEEE Journal on Selected Topics in Signal Processing, IEEE Transactions on Knowledge and Data Engineering, Computers & Security, and Pattern Recognition.

Kuldip K. Paliwal received the B.S. degree from Agra University, Agra, India, in 1969, the M.S. degree from Aligarh Muslim University, Aligarh, India, in 1971 and the Ph.D. degree from Bombay University, Bombay, India, in 1978.

He has been carrying out research in the area of speech processing since 1972. He has worked at a number of organizations including Tata Institute of Fundamental Research, Bombay, India, Norwegian Institute of Technology, Trondheim, Norway, University of Keele, U.K., AT&T Bell Laboratories, Murray Hill, New Jersey, U.S.A., AT&T Shannon Laboratories, Florham Park, New Jersey, U.S.A., and Advanced Telecommunication Research Laboratories, Kyoto, Japan. Since July 1993, he has been a professor at Griffith University, Brisbane, Australia, in the School of Microelectronic Engineering. His current research interests include speech recognition, speech coding, speaker recognition, speech enhancement, face recognition, image coding, pattern recognition and artificial neural networks. He has published more than 250 papers in these research areas.

Dr. Paliwal is a Fellow of Acoustical Society of India. He has served the IEEE Signal Processing Society's Neural Networks Technical Committee as a founding member from 1991 to 1995 and the Speech Processing Technical Committee from 1999 to 2003. He was an Associate Editor of the IEEE Transactions on Speech and Audio Processing during the periods 1994–1997 and 2003–2004. He also served as Associate Editor of the IEEE Signal Processing Letters from 1997 to 2000. He was the General Co-Chair of the Tenth IEEE Workshop on Neural Networks for Signal Processing (NNSP 2000). He has co-edited two books: "Speech Coding and Synthesis" (published by Elsevier), and "Speech and Speaker Recognition: Advanced Topics" (published by Kluwer). He has received IEEE Signal Processing Society's best (senior) paper award in 1995 for his paper on LPC quantization. He is currently serving the Speech Communication journal (published by Elsevier) as its Editor-in-Chief.