Weighted Generalized LDA for Undersampled Problems

JING YANG
School of Computer Science and Technology
Nanjing University of Science and Technology
Xiao Lin Wei Street No.200, 210094, Nanjing
CHINA
yangjing860204@163.com

LIYA FAN
School of Mathematics Sciences
Liaocheng University
Hunan Road 1, 252059, Liaocheng
CHINA
fanliya63@126.com

QUANSEN SUN
School of Computer Science and Technology
Nanjing University of Science and Technology
Xiao Lin Wei Street No.200, 210094, Nanjing
CHINA
sunquansen@mail.njust.edu.cn

Abstract: Linear discriminant analysis (LDA) is a classical approach for dimensionality reduction. It aims to maximize between-class scatter and minimize within-class scatter, thus maximize the class discriminant. However, for undersampled problems where the data dimensionality is larger than the sample size, all scatter matrices are singular and the classical LDA encounters computational difficulty. Recently, many LDA extensions have been developed to overcome the singularity, such as, Pseudo-inverse LDA (PILDA), LDA based on generalized singular value decomposition (LDA/GSVD), null space LDA (NLDA) and range space LDA (RSLDA). Moreover, they endure the Fisher criterion that is nonoptimal with respect to classification rate. To remedy this problem, weighted schemes are presented for several LDA extensions in this paper and called them weighted generalized LDA algorithms. Experiments on Yale face database and AT&T face database are performed to test and evaluate effective of the proposed algorithms and affect of weighting functions.

Key–Words: Feature extraction; dimensionality reduction; undersampled problem; weighting function; misclassification rate

1 Introduction

Many machine learning, data mining and bioinformatics problems involve data in very high-dimensional space. Undersampled problems where data dimension is larger than the sample size, frequently occur in many applications including information retrieval [1-3], face recognition [4-6], and microarray data analysis [7].

The feature extraction process is an important part of pattern recognition and machine learning, which can results in computation cost decreasing and classification performance increasing. An appropriate representation of data from all features is an important problem in machine learning and data mining problems. All original features can not always beneficial for classification or regression tasks. Some features are irrelevant or redundant in distribution of data set. These features can decrease the classification performance. In order to increase the classification performance and to reduce computation cost of classifier, the feature selection process should be used in classification or regression problems [8].

Linear discriminant analysis (LDA) [9] is one of the most popular linear projection techniques for feature extraction. However, the classical LDA usually encounters two difficulties. One is the singularity problem caused by the undersampled problem. In recent years, many LDA extensions have been developed to deal with this problem, such as, pseudo-inverse LDA (PILDA) [10-11], LDA/GSVD [12-14], null space LDA (NLDA) [12,15], range space LDA [16], orthogonal LDA (OLDA) [11,17], uncorrelated LDA (ULDA) [11,18] and regularized LDA (RLDA) [19]. In PILDA, the inverse of the scatter matrix is replaced by the pseudo-inverse. It is equivalent to approximating the solution using a least-squares solution method. The optimal transformation is computed through the simultaneous diagonalization of scatter matrices. LDA/GSVD is one of generalizations of LDA based on GSVD, it overcomes the singularity of the scatter matrices by applying the GSVD to

solve the generalized eigenvalue problem. The classical LDA solution is a special case of LDA/GSVD method. In NLDA, the between-class distance is maximized in the null space of the within-class scatter matrix. It is a two-step approach, the transformation using a basis of null space of the within-class scatter matrix is performed in the first stage and then in the transformed space the second projective directions are searched. In range space LDA, the within-class distance is minimizes in the range space of the between-class scatter matrix. Similarly, we propose a method based on the transformation by a basis of the range space of the within-class scatter matrix to handle undersampled problems.

Another drawback of LDA-based algorithms is that the Fisher separability criterion is not directly related to classification rate. A promising solution to this problem is to introduce weighted schemes into the criteria. We can see from [20-21,25] that weighting functions are close with classification accuracy. Different weighting functions can lead different classification error. Selecting suitable weighting function can increase classification accuracy. In this paper, we focus on weighted versions of PILDA, LDA/GSVD, NLDA and range space LDA with five weighting functions for each weighted scheme, in which the K-Nearest neighbors (KNN) method [22] is used for a classifier. We apply the Euclidean distance $d_{ij} = \|m_i - m_j\|$ between the means of class i and j in weighting functions $w(d_{ij})$. A weighting function is generally a monotonically decreasing function because classes that are closer to one another are likely to have a greater confusion and should be given a greater weightage. For weighting functions, we first apply two special cases of the weighting function $w(d_{ij}) = (d_{ij})^{-p}$ proposed by Lotlikar et al. in [20] with $p = 1$ and $p = 2$, and then an improved version of weighting function $w(d_{ij}) = \frac{1}{2d_{ij}^2} erf(\frac{d_{ij}}{2\sqrt{2}})$ presented by Loog in [25] where the Mahanalobis distance is replaced by the Euclidean distance. In addition, according to the feature of weighting functions, we present two new weighting functions.

The rest of the paper is organized as follows. In Section 2, we briefly review generalized LDA algorithms. Weighted versions of generalized LDA algorithms and weighting functions are introduced in Section 3. Extensive experiments with proposed algorithms have been performed in Section 4, the results demonstrate the effective of the proposed algorithms and the affect of weighting functions. Conclusion follows in Section 5.

2 Generalized LDA

In this section, we review several generalized LDA algorithms which can overcome the limitation of the classical LDA. Given a data matrix $A = [a_1, \cdots, a_n] \in \mathbb{R}^{m\times n}$, where $a_i \in R^m, i = 1, \cdots n$. Assume the original data is already clustered and partitioned into r classes. Let $A = [A_1, \cdots, A_r]$, where $A_i \in R^{m\times n_i}$ is the data matrix belonged to the i-th class and $\sum_{i=1}^{r} n_i = n$. Let N_i be the set of indices of i-th class, i.e., a_j belongs to the i-th class for $j \in N_i$.

The aim of LDA is to find a linear transformation $G \in \mathbb{R}^{m\times l}$ that maps each a_i to $y_i \in R^l$ by $y_i = G^T a_i$ and optimally preserves the cluster structure in the reduced-dimensional space. Let the between-class, within-class and total scatter matrices are defined as $S_b = \sum_{i=1}^{r} n_i(c_i - c)(c_i - c)^T$, $S_w = \sum_{i=1}^{r}\sum_{j\in N_i}(a_j - c_i)(a_j - c_i)^T$ and $S_t = \sum_{j=1}^{n}(a_j - c)(a_j - c)^T$ (see [1]), where $c_i = (1/n_i)\sum_{j\in N_i} a_j$ and $c = (1/n)\sum_{j=1}^{n} a_j$ are class centroids and the global centroid, respectively. We can easily see that the trace of S_w measures the within-class closeness and the trace of S_b measures the between-class separation. In the lower-dimensional space obtained from the transformation G, three scatter matrices above become $S_w^L = G^T S_w G, S_b^L = G^T S_b G$ and $S_t^L = G^T S_t G$. An optimal transformation G would maximize trace (S_b^L) and minimize trace (S_w^L), simultaneously. In classical LDA which requires S_w or S_b is nonsingular, common optimizations include

$$\begin{aligned} &\max_G\{\mathrm{trace}((S_w^L)^{-1}S_b^L)\} \\ &\min_G\{\mathrm{trace}((S_b^L)^{-1}S_w^L)\}. \end{aligned} \tag{1}$$

The problem (1) is equivalent to finding the eigenvectors satisfying the generalized eigen equation $S_b x = \lambda S_w x$ for $\lambda \neq 0$. The solution can be obtained by solving an eigenvalue problem on the matrix $S_w^{-1}S_b$ if S_w is nonsingular or on $S_b^{-1}S_w$ if S_b is nonsingular. There are at most $r - 1$ eigenvectors corresponding to nonzero eigenvalues since $\mathrm{rank}(S_b) \leq r - 1$. Therefore, the reduced dimension by classical LDA is at most $r - 1$. A stable way to solve this eigenvalue problem is to apply singular value decomposition (SVD) on the scatter matrices, the details can be found in [5].

When the data dimensionality is larger than the sample size, all scatter matrices are singular and the classical LDA is no longer applicable. In order to solve the small sampled size problems, several generalizations of LDA have been proposed.

2.1 Pseudo-inverse LDA

The pseudo-inverse of a matrix A, denoted as A^+, refers to a unique matrix satisfying $A^+AA^+ = A^+, AA^+A = A, (AA^+)^T = AA^+$ and $(A^+A)^T = A^+A$. The pseudo-inverse A^+ is commonly computed by SVD [23]. If $A = U \begin{bmatrix} \Sigma & 0 \\ 0 & 0 \end{bmatrix} V^T$ is the SVD of the matrix A, where U and V are orthogonal and Σ is diagonal with positive diagonal entries, then $A^+ = V \begin{bmatrix} \Sigma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T$.

In this subsection, we consider the criterion $F_1(G) = \text{trace}((S_b^L)^+ S_w^L)$ proposed in [10], which is a natural extension of (1) since the inverse of a matrix may not exist, but the pseudo-inverse of a matrix is well-defined. Moreover, when the matrix is invertible, its pseudo-inverse is the same as its inverse. In PILDA, the optimal transformation matrix G can be obtained by solving a minimization problem

$$\min_G F_1(G). \tag{2}$$

The main technique for solving the problem (2) is to use GSVD. Define the matrices

$$\begin{aligned} H_w &= [A_1 - c_1(e_1)^T, \cdots, A_r - c_r(e_r)^T], \\ H_b &= [\sqrt{n_1}(c_1 - c), \cdots, \sqrt{n_r}(c_r - c)], \\ H_t &= [a_1 - c, \cdots, a_n - c], \end{aligned}$$

where $e_i = (1, \cdots, 1)^T \in \mathbb{R}^{n_i}$. Then, the scatter matrices can be expressed as $S_w = H_w H_w^T, S_b = H_b H_b^T$ and $S_t = H_t H_t^T$. Let GSVD be applied to the matrix pair (H_b^T, H_w^T) such that

$$\begin{aligned} U_b^T H_b^T X &= [\ \Gamma_b \quad 0\ ], \\ U_w^T H_w^T X &= [\ \Gamma_w \quad 0\ ], \end{aligned} \tag{3}$$

where $U_b \in \mathbb{R}^{r\times r}$ and $U_w \in \mathbb{R}^{n\times n}$ are orthogonal, $X \in \mathbb{R}^{m\times m}$ is a nonsingular matrix, $\Gamma_b^T\Gamma_b + \Gamma_w^T\Gamma_w = I_s, s = \text{rank} \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$ and $\Gamma_b^T\Gamma_b$ and $\Gamma_w^T\Gamma_w$ are diagonal matrices with nonincreasing and nondecreasing diagonal components, respectively. The simultaneous diagonalizations of S_b and S_w can be obtained by

$$\begin{aligned} X^T S_b X &= \begin{bmatrix} \Gamma_b^T\Gamma_b & \\ & 0_{m-s} \end{bmatrix} \\ &= \begin{bmatrix} \mathrm{I}_\mu & & & \\ & D_\tau & & \\ & & 0_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix} \\ &= D_b, \\ X^T S_w X &= \begin{bmatrix} \Gamma_w^T\Gamma_w & \\ & 0_{m-s} \end{bmatrix} \\ &= \begin{bmatrix} 0_\mu & & & \\ & E_\tau & & \\ & & \mathrm{I}_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix} \\ &= D_w, \end{aligned} \tag{4}$$

where I and 0 denote identity and zero matrices, respectively. $D_\tau = \text{diag}(\alpha_{\mu+1}, \cdots, \alpha_{\mu+\tau})$, $E_\tau = \text{diag}(\beta_{\mu+1}, \cdots, \beta_{\mu+\tau})$ with $\alpha_{\mu+1} \geq \cdots \geq \alpha_{\mu+\tau} > 0, 0 < \beta_{\mu+1} \leq \cdots \leq \beta_{\mu+\tau}$ and $\alpha_i^2 + \beta_i^2 = 1$ for $i = \mu+1, \cdots, \mu+\tau$. By (4), we can deduce that

$$X^T S_t X = X^T S_b X + X^T S_w X = \begin{bmatrix} \mathrm{I}_s & 0 \\ 0 & 0 \end{bmatrix}.$$

To solve the problem (2), we need the following three lemmas, where the proofs of the first two are straightforward from the definition of the pseudo-inverse and the third lemma can be found in [10].

Lemma 1 *For any matrix $A \in \mathbb{R}^{m\times n}$, we have trace$(AA^+)$ = rank(A).*

Lemma 2 *For any matrix $A \in \mathbb{R}^{m\times n}$, we have $(AA^T)^+ = (A^+)^T A^+$.*

Lemma 3 *Let $\Sigma = diag(\sigma_1, \cdots, \sigma_s)$ be any diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_s > 0$. Then, for any matrix $M \in \mathbb{R}^{m\times s}$ with rank$(M) = \delta$, the following inequality holds:*

$$trace((M\Sigma M^T)^+ MM^T) \geq \sum_i^\delta \tfrac{1}{\sigma_i},$$

where the equality holds if and only if $M = U \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix}$ for some orthogonal matrix $U \in \mathbb{R}^{m\times m}$ and matrix $D = \Sigma_1 Q \Sigma_2 \in \mathbb{R}^{\delta\times\delta}$, where $Q \in \mathbb{R}^{\delta\times\delta}$ is orthogonal and $\Sigma_1, \Sigma_2 \in \mathbb{R}^{\delta\times\delta}$ are diagonal matrices with positive diagonal entries.

Theorem 4 *Let X be the matrix specified by the GSVD of (H_b^T, H_w^T) in (3) and X_δ the matrix consisting of the first δ columns of X, where $\delta = rank(S_b)$. Then $G = X_\delta M$ minimizes F_1 for any nonsingular M.*

Proof: By the GSVD of the matrix pair (H_b^T, H_w^T), we have $S_b^L = \tilde{G} D_b \tilde{G}^T$ and $S_w^L = \tilde{G} D_w \tilde{G}^T$, where $\tilde{G} = (X^{-1}G)^T$. Let $\tilde{G} = [G_1, G_2, G_3, G_4]$ be a partition of $\tilde{G}$ such that $G_1 \in \mathbb{R}^{l\times\mu}, G_2 \in \mathbb{R}^{l\times\tau}$, $G_3 \in \mathbb{R}^{l\times(s-\mu-\tau)}$ and $G_4 \in \mathbb{R}^{l\times(m-s)}$. Putting $G_{12} = [G_1, G_2]$, we have $S_w^L + S_b^L = G_{12}G_{12}^T + G_3G_3^T$ and $S_b^L = G_{12}\Sigma G_{12}^T$, where $\Sigma = \begin{bmatrix} I_\mu & \\ & D_\tau \end{bmatrix}$. By Lemma 1, we get

$$\begin{aligned} \text{trace}((S_b^L)^+ S_b^L) &= \text{rank}(S_b^L) \\ &\le \text{rank}(S_b) = \delta. \end{aligned} \tag{5}$$

Let $\hat{G} = G_{12}\Sigma^{1/2}$ and $\hat{G} = U \begin{bmatrix} \hat{\Sigma} & 0 \\ 0 & 0 \end{bmatrix} V^T$ be the SVD of $\hat{G}$. Then $\text{rank}(\hat{G}) = \text{rank}(G_{12})$ and $\hat{G}^+ = V \begin{bmatrix} \hat{\Sigma}^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T$. Consequently, by Lemma 3, we can deduce that

$$\begin{aligned} &\text{trace}((S_b^L)^+(S_w^L + S_b^L)) \\ &= \text{trace}((G_{12}\Sigma G_{12}^T)^+(G_{12}G_{12}^T + G_3G_3^T)) \\ &\ge \text{trace}((G_{12}\Sigma G_{12}^T)^+ G_{12}G_{12}^T) \\ &= \text{trace}((\hat{G}^+)^T \hat{G}^+ \hat{G}\Sigma^{-1}\hat{G}^T) \\ &= \text{trace}(V_\delta^T \Sigma^{-1} V_\delta) \\ &\ge \sum_{i=1}^{\mu} 1 + \sum_{i=\mu+1}^{\delta} \frac{1}{\alpha_i^2}, \end{aligned} \tag{6}$$

where V_δ is the matrix consisting of the first δ columns of V. We can easily show that the last inequality in (6) becomes equality if $V_\delta = \begin{bmatrix} Q_\delta \\ 0 \end{bmatrix}$ for any orthogonal matrix $Q_\delta \in \mathbb{R}^{\delta\times\delta}$ and $G_{12} = U \begin{bmatrix} D & 0 \\ 0 & 0 \end{bmatrix} \in \mathbb{R}^{l\times(\mu+\tau)}$, where $D = \hat{\Sigma} Q_\delta \Sigma_\delta^{-1/2}$ and Σ_δ is the δ-th principal submatrix of Σ. It follows from (5)-(6) that

$$\begin{aligned} F_1(G) &= \text{trace}((S_b^L)^+(S_w^L + S_b^L)) - \text{trace}((S_b^L)^+ S_b^L) \\ &\ge \sum_{i=\mu+1}^{\delta} \left(\frac{\beta_i}{\alpha_i}\right), \end{aligned}$$

where the equality holds if $\text{rank}(S_b^L) = \delta$, $G_{12} = U \begin{bmatrix} \hat{\Sigma} Q_\delta \Sigma_\delta^{-1/2} & 0 \\ 0 & 0 \end{bmatrix}$ and $G_3 = 0$.

We observe that the minimization of F_1 is independent of G_4 and simply set it to zero. Hence, the minimum of F_1 is attained if the partition of $\tilde{G} = [G_1, G_2, G_3, G_4]$ satisfies

$$G_{12} = U \begin{bmatrix} \hat{\Sigma} Q_\delta \Sigma_\delta^{-1/2} & 0 \\ 0 & 0 \end{bmatrix}, G_3 = 0 \text{ and } G_4 = 0.$$

Let $U = [U_1, U_2]$ be a partition of U so that $U_1 \in \mathbb{R}^{l\times\delta}$ and $U_2 \in \mathbb{R}^{l\times(l-\delta)}$. It follows that

$$\begin{aligned} G_{12} &= U \begin{bmatrix} \hat{\Sigma} Q_\delta \Sigma_\delta^{-1/2} & 0 \\ 0 & 0 \end{bmatrix} \\ &= \begin{bmatrix} U_1 \hat{\Sigma} Q_\delta \Sigma_\delta^{-1/2} & 0 \\ 0 & 0 \end{bmatrix} \\ &\equiv \begin{bmatrix} M & 0 \\ 0 & 0 \end{bmatrix}. \end{aligned}$$

Note that U is orthogonal matrix, Q_δ is an arbitrary orthogonal matrix and $\hat{\Sigma}$ and Σ_δ are diagonal matrices. Therefore, M is an arbitrary nonsingular matrix and $G = X_\delta M$ minimizes $F_1(G)$. □

This method is called PILDA, that is,

Algorithm 2.1. PILDA

1. Compute the SVD of $K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$ as $K = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} U^T$, where P and U are orthogonal and R is diagonal;
2. Let $s = \text{rank}(K)$ and $\delta = \text{rank}(H_b)$. Compute W from the SVD of $P(1:r, 1:s)$, the submatrix consisting of the first r rows and the first s columns of matrix P from Step 1, as $P(1:r, 1:s) = V\Gamma W^T$;
3. Let $X = U \begin{bmatrix} R^{-1}W & 0 \\ 0 & I \end{bmatrix}$;
4. Assign the first δ columns of the matrix X to X_δ;
5. $G = X_\delta M$, where M is any nonsingular matrix.

2.2 LDA based on GSVD

In this subsection, we review another method based on the GSVD proposed by Howland et al. [12-14]. This approach is justified to preserve the cluster structure after dimension reduction.

When $S_w = H_w H_w^T$ is nonsingular, it is well-known that $\text{trace}((S_w^L)^{-1} S_b^L)$ is maximized if $G_h \in \mathbb{R}^{m\times l}$ consists of l eigenvectors of $S_w^{-1}S_b$ corresponding to the l largest eigenvalues [1]. Let x_i denote the i-th column of X, then

$$S_b x_i = \lambda_i S_w x_i, \tag{7}$$

which means that λ_i and x_i are an eigenvalue-eigenvector pair of $S_w^{-1}S_b$ and $\text{trace}(S_w^{-1}S_b) = \lambda_1 + \cdots + \lambda_m$. Expressing λ_i as α_i^2/β_i^2, the eigenvalue problem (7) becomes

$$\beta_i^2 H_b H_b^T x_i = \alpha_i^2 H_w H_w^T x_i. \quad (8)$$

Let $X = [X_1, X_2, X_3, X_4] \in \mathbb{R}^{m\times m}$ be a partition of X obtained in section 2.1 with $X_1 \in \mathbb{R}^{m\times\mu}$, $X_2 \in \mathbb{R}^{m\times\tau}$, $X_3 \in \mathbb{R}^{m\times(s-\mu-\tau)}$ and $X_4 \in \mathbb{R}^{m\times(m-s)}$. Defining $\alpha_i = 1, \beta_i = 0$ for $i = 1, \cdots, \mu$ and $\alpha_i = 0, \beta_i = 1$ for $i = \mu+\tau+1, \cdots s$, we can see that (8) is satisfied for $1 \le i \le s$. For the remaining $m-s$ columns of X, $H_b H_b^T x_i$ and $H_w H_w^T x_i$ are zero. So, (8) is satisfied for arbitrary values of α_i and β_i when s + 1 $\le i \le m$ and then the columns of X are the generalized singular vectors for the matrix pair (H_b^T, H_w^T). A question that remains is which columns of X to include in G_h. If S_w is singular, [14] argues in terms of the simultaneous optimization

$$\begin{aligned} &\max_{G_h^T}(\text{trace}(G_h^T S_b G_h)) \\ &\min_{G_h^T}(\text{trace}(G_h^T S_w G_h)). \end{aligned} \quad (9)$$

Letting g_j represent a column of G_h, we have $\text{trace}(G_h^T S_b G_h) = \sum g_j^T S_b g_j$ and $\text{trace}(G_h^T S_w G_h) = \sum g_j^T S_w g_j$. If x_i is the one of the leftmost μ columns of X, then $x_i \in \text{null}(S_w) \cap \text{null}(S_b)^c$ (the superscript c denotes the complement), which indicates that including x_i in G_h can increase $\text{trace}(G_h^T S_b G_h)$ while leave $\text{trace}(G_h^T S_w G_h)$ unchanged. If x_i is the one of the rightmost $m-s$ columns of X, then $x_i \in \text{null}(S_w) \cap \text{null}(S_b)$, which implies that adding x_i to G_h has no effect on these trace and does not contribute to either maximization or minimization in (9). In either case, G_h should be comprised of the leftmost $\mu + \tau = \text{rank}(H_b^T)$ columns of X, as illustrated in [24]. Hence, in LDA/GSVD the transformation matrix is $G_h = [X_1, X_2]$. An efficient algorithm for LDA/GSVD was presented in [12] as follows.

Algorithm 2.2. LDA/GSVD

1. Compute the EVD of S_t:
$$S_t = \begin{bmatrix} U_1 & U_2 \end{bmatrix} \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^T \\ U_2^T \end{bmatrix}.$$
2. Compute V from the EVD of $\tilde{S}_b = \Sigma_1^{-1/2} U_1^T S_b U_1 \Sigma_1^{-1/2}$: $\tilde{S}_b = V \Gamma_b^T \Gamma_b V^T$.
3. Assign the first $r-1$ columns of $U_1 \Sigma_1^{-1/2} V$ to G_h.

2.3 Null space LDA

Chen et al. [15] proposed the null space LDA (NLDA) for dimensionality reduction of undersampled problems, which is a two-stage LDA method. This approach projects the original space onto the null space of S_w by using an orthonormal basic of null (S_w), and then in the projected space, a transformation that maximizes the between-class scatter is computed. Let the EVD of $S_w \in R^{m\times m}$ be

$$\begin{aligned} S_w &= U_w \Sigma_w U_w^T \\ &= \begin{bmatrix} U_{w1} & U_{w2} \end{bmatrix} \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix}, \end{aligned}$$

where $s_1 = \text{rank}(S_w), U_w$ is orthogonal, Σ_{w1} is a diagonal matrix with nonincreasing positive diagonal elements and U_{w1} contains the first s_1 columns of the orthogonal matrix U_w. We can easily show that $\text{null}(S_w) = \text{span}(U_{w2})$ and the transformation by $U_{w2} U_{w2}^T$ projects the original data to $\text{null}(S_w)$. The between-class scatter matrix $\tilde{S}_b$ in the transformed space is $\tilde{S}_b = U_{w2} U_{w2}^T S_b U_{w2} U_{w2}^T$. Consider the EVD of $\tilde{S}_b$:

$$\tilde{S}_b = \tilde{U}_b \tilde{\Sigma}_b \tilde{U}_b^T = \begin{bmatrix} \tilde{U}_{b1} & \tilde{U}_{b2} \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix},$$

where $s_2 = \text{rank}(\tilde{S}_b)$, $\tilde{U}_{b1} \in \mathbb{R}^{m\times s_2}$ and $\tilde{\Sigma}_{b1} \in \mathbb{R}^{s_2\times s_2}$. In NLDA, the optimal transformation matrix G_e is obtained by $G_e = U_{w2} U_{w2}^T \tilde{U}_{b1}$.

2.4 Range space LDA

In this subsection, we propose another approach to solve undersampled problems. This method first transforms the original space by using a basis of range(S_w) and then in the transformed space the maximization of between-class scatter is pursued. We denote shortly this method by RSLDA. Let the EVD of $S_w \in \mathbb{R}^{m\times m}$ be

$$\begin{aligned} S_w &= U_w \Sigma_w U_w^T \\ &= \begin{bmatrix} U_{w1} & U_{w2} \end{bmatrix} \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix}, \end{aligned}$$

where $s_1 = \text{rank}(S_w), \Sigma_{w1}$ is a diagonal matrix and $U_{w1} \in \mathbb{R}^{m\times s_1}$. We can easily show that $\text{range}(S_w) = \text{span}(U_{w1})$ and the transformation by $V_y = U_{w1} \Sigma_{w1}^{-1/2}$ projects the original data to range(S_w). The within-class scatter matrix $\tilde{S}_w$ in the transformed space is $\tilde{S}_w = V_y^T S_w V_y = I_{s_1}$. Let the EVD of $\tilde{S}_b \equiv V_y^T S_b V_y$ be

$$\tilde{S}_b = \tilde{U}_b \tilde{\Sigma}_b \tilde{U}_b^T = \begin{bmatrix} \tilde{U}_{b1} & \tilde{U}_{b2} \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix},$$

where $s_3 = \text{rank}(\tilde{S}_b), \tilde{\Sigma}_{b1}$ is a diagonal matrix and $\tilde{U}_{b1} \in \mathbb{R}^{s_1\times s_3}$. In RSLDA, the optimal transformation matrix G_y is obtained by $G_y = V_y \tilde{U}_{b1} = U_{w1} \Sigma_{w1}^{-1/2} \tilde{U}_{b1}$.

3 Weighted versions and weighting functions

As in [25], S_b can be rewritten as

$$S_b = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \frac{n_i n_j}{n} (m_i - m_j)(m_i - m_j)^T.$$

From this formulation, it can be observed that classical LDA maximize the Euclidean distance between the class means and all class pairs have the same weights irrespective of their separability in the original space, which makes that the resulting transformation preserve the distance of already well-separated classes and cause a large overlap of neighboring classes in the transformed space. In fact, the classes which are closer together are more likely to have more confusion and should therefore be more heavily weighted.

Two similarly motivated solutions to this problem have been proposed: weighted pairwise Fisher criteria [25] and fractional-step LDA [20]. The fractional-step LDA is iterative and very time-consuming. In this section, in order to improve the classification accuracy, we study the weighted versions of generalized LDA mentioned in Section 2. We define a weighted between-class scatter matrix as

$$\hat{S}_b = \sum_{i=1}^{r-1} \sum_{j=i+1}^{r} \frac{n_i n_j}{n} w(d_{ij})(m_i - m_j)(m_i - m_j)^T, \tag{10}$$

where $d_{ij} = \|m_i - m_j\|$ is the Euclidean distance between the means of class i and j and $w(d_{ij})$ is a weighting function. Apparently, the weighted between-class scatter matrix $\hat{S}_b$ degenerates to the conventional between-class scatter matrix S_b if the weighting function in (10) gives a constant weight value. If the between-class scatter matrix S_b is replaced by the weighted between-class scatter matrix $\hat{S}_b$, by means of the algorithms obtained in Section 2, we can get some weighted generalized LDA.

3.1 Weighted generalized LDA

With Algorithms 2.1 and 2.2, we can derive weighted PILDA and weighted LDA/GSVD algorithms.

Algorithm 3.1. Weighted PILDA

1. Compute the EVD of S_t:
$S_t = [\ U_{t1}\ U_{t2}\ ] \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix}$;
2. Compute the weighted between-class scatter matrix $\hat{S}_b$;
3. Compute V from the EVD of $\check{S}_b = \Sigma_{t1}^{-1/2} U_{t1}^T \hat{S}_b U_{t1} \Sigma_{t1}^{-1/2} : \check{S}_b = V \hat{\Gamma}_b^T \hat{\Gamma}_b V^T$;
4. Assign the first $r-1$ columns of $U_{t1}\Sigma_{t1}^{-1/2}V$ to $\hat{X}_\delta$;
5. $\hat{G} = \hat{X}_\delta M$, where M is any nonsingular matrix.

Algorithm 3.2. Weighted LDA/GSVD

1. Compute the EVD of S_t:
$S_t = [\ U_{t1}\ U_{t2}\ ] \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix}$;
2. Compute the weighted between-class scatter matrix $\hat{S}_b$;
3. Compute V from the EVD of $\check{S}_b = \Sigma_{t1}^{-1/2} U_{t1}^T \hat{S}_b U_{t1} \Sigma_{t1}^{-1/2} : \check{S}_b = V \hat{\Gamma}_b^T \hat{\Gamma}_b V^T$;
4. Assign the first $r-1$ columns of $U_{t1}\Sigma_{t1}^{-1/2}V$ to $\hat{G}_h$.

From Algorithms 3.1 and 3.2, we can see that weighted LDA/GSVD is a special case of weighted PILDA with M being the identity matrix. We will discuss the affect of the choice of M to classification accuracy in next section.

Similarly, we can obtain weighted NLDA and weighted RSLDA.

Algorithm 3.3. Weighted NLDA

1. Compute the EVD of S_w:
$S_w = [\ U_{w1}\ U_{w2}\ ] \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix}$;
2. Compute the weighted between-class scatter matrix $\hat{S}_b$;
3. Compute the EVD of $\tilde{S}_b = U_{w2} U_{w2}^T \hat{S}_b U_{w2} U_{w2}^T$:
$\tilde{S}_b = [\ \tilde{U}_{b1}\ \tilde{U}_{b2}\ ] \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix}$;
4. $\hat{G}_e = U_{w2} U_{w2}^T \tilde{U}_{b1}$.

Algorithm 3.4. Weighted RSLDA

1. Compute the EVD of S_w:
$S_w = [\ U_{w1}\ U_{w2}\ ] \begin{bmatrix} \Sigma_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{w1}^T \\ U_{w2}^T \end{bmatrix}$;
2. Compute the weighted between-class scatter matrix $\hat{S}_b$;

3. Compute the EVD of $\tilde{S}_b = \Sigma_{w1}^{-1/2} U_{w1}^T \hat{S}_b U_{w1} \Sigma_{w1}^{-1/2}$:

$$\tilde{S}_b = [\ \tilde{U}_{b1}\ \tilde{U}_{b2}\ ] \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix};$$

4. $\hat{G}_y = U_{w1} \Sigma_{w1}^{-1/2} \tilde{U}_{b1}$.

3.2 Weighting functions

We can see from [20-21,25] that weighting functions have close relationships with classification accuracy. Different weighting functions can product different classification error. Selecting suitable weighting function can increase classification accuracy. In this paper, we consider four weighted schemes Algorithms 3.1-3.4 with five weighting functions for each weighted scheme. We apply the Euclidean distance $d_{ij} = \|m_i - m_j\|$ between the means of class i and j in weighting functions $w(d_{ij})$. A weighting function is generally a monotonically decreasing function because classes that are closer to one another are likely to have a greater confusion and should be given a greater weightage.

We first apply two special cases of the weighting function $w(d_{ij}) = (d_{ij})^{-p}$ proposed by Lotlikar et al. in [20] with $p = 1$ and $p = 2$, and then an improved version of weighting function $w(d_{ij}) = \frac{1}{2d_{ij}^2} erf(\frac{d_{ij}}{2\sqrt{2}})$ presented by Loog in [25] where the Mahanalobis distance is replaced by the Euclidean distance. In addition, according to the feature of weighting functions mentioned above, we present two new weighting functions. They are listed below:

w_1: $w(d_{ij}) = (d_{ij})^{-2}$,
w_2: $w(d_{ij}) = \frac{1}{2d_{ij}^2} erf(\frac{d_{ij}}{2\sqrt{2}})$,
w_3: $w(d_{ij}) = (d_{ij})^{-1}$,
w_4: $w(d_{ij}) = e^{\frac{1}{d_{ij}}}$,
w_5: $w(d_{ij}) = \frac{1}{e^{d_{ij}}}$.

4 Experiments and analysis

In this section, in order to explain the effective of the proposed methods and illustrate the affect of weighting functions and any nonsingular matrix M in PILDA and weighted PILDA to classification accuracy, we conduct a series of experiments on 5 different data sets from the AT&T face database [26] and Yale face database [27].

There are ten different images of 40 distinct subjects in AT&T (or called ORL) face image database. For some subjects, the images were taken at different times, varying the lighting, facial details and facial expressions. The size of each images is 92×112 pixels, with 256 grey levels per pixel. The Yale face database contains 165 face images of 15 individuals. There are 11 images per subject, and these 11 images are respectively under the following different facial expression or configuration: center-light, wearing glasses, happy, left-light, wearing no glasses, normal, right-light, sad, sleepy, surprised, and wink. In our experiment, the images are cropped to a size of 32×32, and the gray level values of all images are rescaled to [0 1]. Some images of one person are shown in Figure 1.

Figure 1: Images of one person in Yale.

Data sets 1-3 are from Yale face database. Data set 1 chooses the 201-th dimension to 600-th dimension, it contains 15 classes and 165 examples with 400 dimensions; data set 2 chooses the 401-th dimension to 900-th dimension,it contains 15 classes and 165 examples with 500 dimensions; data set 3 chooses the 800-th dimension to 1024-th dimension, contains 15 classes and 165 examples with 225 dimensions. Data sets 4-5 are from AT&T face database. Data set 4 chooses the first 150 samples and the first 305 dimensions, it contains 15 classes and 150 examples with 305 dimensions; data set 5 chooses the first 150 samples and the 333-th dimension to 665-th dimension, it contains 15 classes and 150 examples with 333 dimensions. For all 5 data sets, the number of data samples is smaller than the dimension of data space, all scatter matrices are singular.

In the following experiments, the KNN algorithm with $K = 7$ is used as a classifier for all date sets. For each method, we applied 5-fold cross-validation to compute the misclassification rate. Experiments are repeated 5 times to obtain mean prediction misclassification rate.

4.1 Effect of the matrix M

In this subsection, we study the effect of the matrix M in PILDA to classification accuracy on five data sets. We randomly generate 5 matrices for M and compute the misclassification rates by using the optimal transformation matrices produced in PILDA and 7-NN. The experiment results are listed in Table 1, where w_0: $w(d_{ij}) = 1$.

From Table 1, we can see that matrix M_3 pro-

Table 1: Misclassification rate (%) on five data sets

kernel	$w(d_{ij})$	W-PILDA					W-LDA/GSVD	W-NLDA	W-RSLDA
		M_1	M_2	M_3	M_4	M_5			
date set 1	w_0	17.5758	20.6061	13.3333	13.9394	18.1818	16.3636	18.7879	16.9697
	w_1	18.7879	15.7576	18.7879	15.1515	15.7576	19.3939	16.9697	16.3636
	w_2	15.7576	16.3636	12.7273	18.1818	13.9394	16.9697	13.3333	18.1818
	w_3	16.9697	17.5758	18.7879	18.7879	13.9394	18.1818	14.5455	18.7879
	w_4	17.5758	16.9697	18.7879	20.0000	17.5758	18.1818	20.0000	19.3939
	w_5	16.9697	15.1515	14.5455	18.1818	13.9394	16.9697	16.3636	18.1818
date set 2	w_0	11.5152	13.3333	13.3333	8.4848	12.1212	14.5455	13.3333	15.1515
	w_1	10.9091	16.9697	9.6970	13.9394	10.9091	16.3636	10.9091	13.3333
	w_2	10.9091	13.9394	10.9091	13.9394	13.3333	14.5455	10.3030	10.9091
	w_3	10.9091	13.9394	10.3030	10.9091	10.3030	15.1515	13.3333	13.9394
	w_4	10.3030	12.7273	12.1212	12.1212	12.1212	15.1515	12.1212	15.1515
	w_5	11.5152	16.3636	12.7273	13.9394	12.7273	14.5455	12.1212	14.5455
date set 3	w_0	41.8182	40.6061	41.8182	38.1818	47.2727	38.7879	38.1818	36.3636
	w_1	50.3030	43.6364	46.0606	40.0000	40.6061	44.8485	40.0000	38.1818
	w_2	44.2424	46.0606	44.2424	41.8182	41.8182	46.6667	35.7576	34.5455
	w_3	43.0303	41.2121	43.6364	47.8788	46.0606	45.4545	43.6364	33.9394
	w_4	42.4242	43.6364	46.0606	41.2121	38.7879	44.8485	38.1818	35.1515
	w_5	44.8485	37.5758	43.0303	43.6364	40.6061	44.8485	39.3939	33.9394
date set 4	w_0	44.6667	55.3333	52.0000	48.6667	50.6667	47.3333	45.3333	42.6667
	w_1	46.6667	47.3333	52.6667	44.6667	47.3333	43.3333	44.6667	40.6667
	w_2	46.6667	47.3333	52.6667	44.6667	47.3333	43.3333	44.6667	40.6667
	w_3	51.3333	48.6667	49.3333	42.0000	52.0000	50.0000	46.0000	41.3333
	w_4	48.0000	47.3333	51.3333	45.3333	46.6667	50.0000	45.3333	42.6667
	w_5	94.0000	92.0000	93.3333	92.0000	92.0000	96.0000	86.6667	84.0000
date set 5	w_0	56.6667	64.0000	57.3333	58.6667	54.6667	59.3333	52.0000	38.6667
	w_1	59.3333	57.3333	58.0000	56.0000	64.6667	57.3333	55.3333	37.3333
	w_2	59.3333	57.3333	58.0000	56.0000	64.6667	57.3333	55.3333	37.3333
	w_3	59.3333	56.6667	61.3333	58.0000	60.6667	58.6667	52.6667	42.0000
	w_4	62.0000	56.0000	60.0000	57.3333	62.6667	59.3333	52.0000	38.6667
	w_5	96.0000	96.0000	92.6667	91.3333	90.6667	92.0000	93.3333	87.3333

duces the best classification accuracy for data set 1; None of the accuracies is higher than matrix M_4 for data set 2; Matrix M_4 produces the best classification accuracy among PILDA, but is not higher than RSLDA for data set 3; Matrix M_1 produces the best classification accuracy among PILDA, but is not higher than RSLDA for data set 4; Matrix M_5 produces the best classification accuracy among PILDA, but is not higher than NLDA and RSLDA for data set 5.

4.2 Affect of weighting functions

In this subsection, we study the effect of five weighting functions given in subsection 3.2 to classification accuracy on five data sets. The experiment results are listed in Table 1.

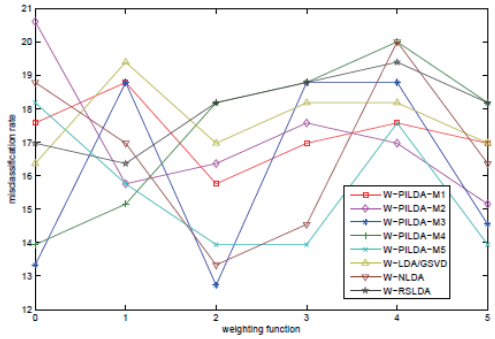

Figure 2: The misclassification rates (%) for Data set 1.

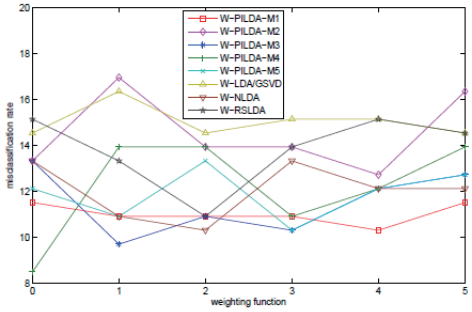

Figure 3: The misclassification rates (%) for Data set 2.

From Figure 2, we can see that the weighting function w_2 is better than other weighting functions for data set 1. Especially, we get the best classification accuracy 87.2727 for M_3 and w_2. Figure 3 shows that the weighting function w_3 produces good overall results for PILDA, the weighting function w_2 produces good overall results for LDA/GSVD, NLDA and RSLDA. For data set 3, the weighting function w_4 produces good overall results for PILDA and LDA/GSVD, the weighting function w_2 produces good overall result for NLDA , the weighting functions w_3 and w_5 produce the best results for RSLDA. For data sets 4 and 5, the weighting function w_5 can't be applied, the weighting functions w_1 and w_2 produce same results. For data set 4 and M_1, M_2, the best weighting functions for LDA/GSVD, NLDA and RSLDA are w_1 and w_2. The weighting function w_3 produces the best result for M_3 and M_4. The best weighting function for M_5 is w_4. For data set 5 and M_1, M_3, M_4, the best weighting functions for LDA/GSVD and RSLDA are w_1 and w_2. The weighting function w_3 produces the best results for M_2 and M_5 and NLDA.

5 Conclusion

In this paper, based on Pseudo-inverse LDA, LDA/GSVD, null space LDA and range space LDA, we propose weighted Pseudo-inverse LDA, weighted LDA/GSVD, weighted null space LDA and weighted range space LDA. Not only can these methods deal with the singularity problem caused by the undersampled problem, they can also make the criterions directly related to classification errors. In order to explain the effective of the proposed methods and illustrate the affect of weighting functions and any nonsingular matrix M in PILDA and weighted PILDA to classification accuracy, we conduct a series of experiments on 5 different data sets from the AT&T face database and Yale face database. Results show that different weighting functions and different nonsingular matrix M affect the classification accuracy of the proposed methods.

Acknowledgements: This work is supported by National Natural Science Foundation of China (61273251).

References:

[1] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed. *Academic Press*, 1990.

[2] W. Berry, S. T. Dumais and G. W. O'Brie, Using linear algebra for intelligent information retrieval, *SIAM Review* 37, 1995, pp. 573–595.

[3] Deerwester, S. T. Dumais, G.W. Furnas, T.K. Landauer and R. Harshman, Indexing by latent semantic analysis, *Journal of the Society for Information Scienc* 41, 1990, pp. 391–407.

[4] 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* 19 (7), 1997, pp. 711–720.

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

[6] A. Turk and A. P. Pentland, Face recognition using Eigenfaces, *In IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[7] 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* 97 (457), 2002, pp. 77–87.

[8] Cao, Shen, Sun, Yang, Chen, Feature selection in a kernel space, *In International conference on machine learning (ICML) Oregon*, USA, June 20-24, pp. 121–128.

[9] K. Fukunaga, Introduction to Statistical Pattern Classification, *Academic Press*, San Diego, California, USA, 1990.

[10] 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* 26 (8), 2004b, pp. 982–994.

[11] J. P. Ye, Characterization of a family of algorithms for generalized discriminant analysis on undersampled problems, *Journal of Machine Learning Research* 6, 2005, pp. 483–502.

[12] C. H. Park, H. Park, A comparision of generalized linear discriminant analysis algorithms, *Pattern Recognition* 41, 2008, pp. 1083–1097.

[13] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 26 (8), 2004, pp. 995–1006.

[14] P. Howland, M. Jeon, H. Park, Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition, *SIAM J. Matrix Anal. Appl.* 25 (1), 2003, pp. 165–179.

[15] L. Chen, H. M. Liao, M. Ko, J. Lin, G. Yu, A new LDA-based face recognition system which can solve the small sample size problem, *Pattern Recognition* 33, 2000, pp. 1713–1726.

[16] H. Yu and J. Yang, A direct LDA algorithm for high-dimensional datałwith application to face recognition, *Pattern Recognition* 34, 2001, pp. 2067–2070.

[17] J. P. Ye and T. Xiong, Computational and theoretical analysis of null space and orthogonal linear discriminant analysis, *Journal of Machine Learning Research* 7, 2006, pp. 1183–1204.

[18] Ye, R. Janardan, Q. Li and H. Park, Feature extraction via generalized uncorrelated linear discriminant analysis, *In Proc. International Conference on Machine Learning*, 2004a, pp. 895–902.

[19] J. H. Friedman, Regularized discriminant analysis, *Journal of American Statistical Association* 84 (405), 1989, pp. 165–175.

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

[21] Y. Liang et al. Uncorrelated linear discriminant analysis based on weighted pairwise Fisher criterion, *Pattern Recognition* 40, 2007, pp. 3606–3615

[22] R. O. Duda, P. E. Hart and D. Stork, *Pattern Classification*, Wiley, 2000.

[23] G. H. Golub and C. F. VanLoan, Matrix computations, third ed. *John Hopkins Univ. Press*, 1996.

[24] P. Howland, H. Park, Equivalence of several two-stage methods for linear discriminant analysis, *in: Proceedings of the Fourth SIAM International Conference on Data Mining*, April 2004, pp. 69-77.

[25] M. Loog, R. P. W. Duin, R. Haeb-Umbach, Multiclass linear dimension reduction by weighted pairwise Fisher criteria, *IEEE Trans. Pattern Anal. Mach. Intell.* 23 (7), 2001, pp. 762–766.

[26] Jian-qiang Gao, Li-ya Fan, Li-zhong Xu, Median null(S_w)-based method for face feature recognition, *Applied Mathematics and Computation* 219, 2013, pp. 6410-6419.

[27] Lishan Qiao, Songcan Chen, Xiaoyang Tan, Sparsity preserving projections with applications to face Recognition, *Pattern Recognition* 43 (1), 2010, pp. 331-341.