

A Comparison of Generalized Linear Discriminant Analysis Algorithms

Cheong Hee Park¹ and Haesun Park^{2*}

Dept. of Computer Science and Engineering¹
Chungnam National University
220 Gung-dong, Yuseong-gu
Daejeon, 305-763, Korea
cheonghee@cnu.ac.kr

College of Computing²
Georgia Institute of Technology
801 Atlantic Drive, Atlanta, GA, 30332, USA
hpark@cc.gatech.edu

January 28, 2006

Abstract

Linear Discriminant Analysis (LDA) is a dimension reduction method which finds an optimal linear transformation that maximizes the class separability. However, in undersampled problems where the number of data samples is smaller than the dimension of data space, it is difficult to apply the LDA due to the singularity of scatter matrices caused by high dimensionality. In order to make the LDA applicable, several generalizations of the LDA have been proposed recently. In this paper, we present theoretical and algorithmic relationships among several generalized LDA algorithms and compare their computational complexities and performances in text classification and face

*This work was supported in part by the National Science Foundation grants CCR-0204109 and ACI-0305543. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF). The work of Haesun Park has been performed while serving as a program director at the NSF and was partly supported by IR/D from the NSF.

recognition. Towards a practical dimension reduction method for high dimensional data, an efficient algorithm is proposed, which reduces the computational complexity greatly while achieving competitive prediction accuracies. We also present nonlinear extensions of these LDA algorithms based on kernel methods. It is shown that a generalized eigenvalue problem can be formulated in the kernel-based feature space, and generalized LDA algorithms are applied to solve the generalized eigenvalue problem, resulting in nonlinear discriminant analysis. Performances of these linear and nonlinear discriminant analysis algorithms are compared extensively.

Keywords: Dimension reduction, Feature extraction, Generalized Linear Discriminant Analysis, Kernel methods, Nonlinear Discriminant Analysis, Undersampled problems.

1 Introduction

Linear Discriminant Analysis (LDA) seeks an optimal linear transformation by which the original data is transformed to a much lower dimensional space. The goal of LDA is to find a linear transformation that maximizes class separability in the reduced dimensional space. Hence the criteria for dimension reduction in LDA are formulated to maximize the between-class scatter and minimize the within-class scatter. The scatters are measured by using scatter matrices such as the between-class scatter matrix (S_b), within-class scatter matrix (S_w) and total scatter matrix (S_t). Let us denote a data set A as

$$A = [a_1, \dots, a_n] = [A_1, A_2, \dots, A_r] \in \mathbb{R}^{m \times n}, \quad (1)$$

where a collection of data items in the class i ($1 \leq i \leq r$) is represented as a block matrix $A_i \in \mathbb{R}^{m \times n_i}$ and N_i is the index set of data items in the class i . Each class i has n_i elements and the total number of data is $n = \sum_{i=1}^r n_i$. The between-class scatter matrix S_b , within-

class scatter matrix S_w and total scatter matrix S_t are defined as

$$\begin{aligned} S_b &= \sum_{i=1}^r n_i (c_i - c)(c_i - c)^T, & S_w &= \sum_{i=1}^r \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T, \\ S_t &= \sum_{j=1}^n (a_j - c)(a_j - c)^T \end{aligned}$$

where $c_i = \frac{1}{n_i} \sum_{j \in N_i} a_j$ and $c = \frac{1}{n} \sum_{j=1}^n a_j$ are class centroids and the global centroid, respectively.

The optimal dimension reducing transformation $G^T \in \mathbb{R}^{l \times m}$ ($l < m$) for LDA is the one that maximizes the between-class scatter and minimizes the within-class scatter in a reduced dimensional space. Common optimization criteria for LDA are formulated as the maximization problem of objective functions

$$J_1(G) = \frac{\text{trace}(G^T S_b G)}{\text{trace}(G^T S_w G)}, \quad J_2(G) = \text{trace}((G^T S_w G)^{-1} (G^T S_b G)), \quad J_3(G) = \frac{|G^T S_b G|}{|G^T S_w G|} \quad (2)$$

where $\tilde{S}_i = G^T S_i G$ for $i = b, w$ are scatter matrices in the space transformed by G^T . It is well known [1, 2] that when S_w is nonsingular, the transformation matrix G is obtained by the eigenvectors corresponding to the $r - 1$ largest eigenvalues of

$$S_w^{-1} S_b g = \lambda g. \quad (3)$$

However, for undersampled problems such as text classification and face recognition where the number of data items is smaller than the data dimension, scatter matrices become singular and their inverses are not defined. In order to overcome the problems caused by the singularity of the scatter matrices, several methods have been proposed [3, 4, 5, 6, 7, 8]. In this paper, we present theoretical relationships among several generalized LDA algorithms and compare computational complexities and performances of them.

While linear dimension reduction has been used in many application areas due to its simple concept and easiness in computation, it is difficult to capture a nonlinear relationship in the data by a linear function. Recently kernel methods have been widely used for

nonlinear extension of linear algorithms [9]. The original data space is transformed to a feature space by an implicit nonlinear mapping through kernel methods. As long as an algorithm can be formulated with inner product computations, without knowing the explicit representation of a nonlinear mapping we can apply the algorithm in the transformed feature space, obtaining nonlinear extension of the original algorithm. We present nonlinear extensions of generalized LDA algorithms through the formulation of a generalized eigenvalue problem in the kernel-based feature space.

The rest of the paper is organized as follows. In Section 2, a theoretical comparison of generalized LDA algorithms is presented. We study theoretical and algorithmic relationships among several generalized LDA algorithms and compare their computational complexities and performances. Computationally efficient algorithm is also proposed which computes the exactly same solution as that in [4, 10] but saves computational complexities greatly. In Section 3, nonlinear extensions of these generalized LDA algorithms are presented. A generalized eigenvalue problem is formulated in the nonlinearly transformed feature space for which all the generalized LDA algorithms can be applied resulting in nonlinear dimension reduction methods. Extensive comparisons of these linear and nonlinear discriminant analysis algorithms are conducted. Conclusion follows in Section 4.

2 A Comparison of Generalized LDA Algorithms for Undersampled Problems

2.1 Regularized LDA

In the regularized LDA (RLDA), when S_w is singular or ill-conditioned, a diagonal matrix αI with $\alpha > 0$ is added to S_w . Since S_w is symmetric positive semidefinite, $S_w + \alpha I$ is nonsingular with any $\alpha > 0$. Therefore we can apply the algorithm for the classical LDA

to solve the eigenvalue problem

$$S_b g = \lambda(S_w + \alpha I)g. \quad (4)$$

Two-Class Problem

We now consider a simple case when the data set has two classes, since in that case a comparison of generalized LDA algorithms is easy to illustrate. The two-class problem in LDA is known as Fisher Discriminant Analysis (FDA) [2]. In a two-class case, S_b can be expressed as

$$S_b = \frac{n_1 n_2}{n} (c_1 - c_2)(c_1 - c_2)^T, \quad (5)$$

and the eigenvalue problem (3) is simplified to

$$S_w^{-1} (c_1 - c_2)(c_1 - c_2)^T g = \lambda g \quad (6)$$

when S_w is nonsingular. The solution for (6) is a nonzero multiple of $g = S_w^{-1}(c_1 - c_2)$, and the 1-dimensional representation of any data item $z \in \mathbb{R}^{m \times 1}$ by LDA is obtained as

$$g^T z = (c_1 - c_2)^T S_w^{-1} z = (c_1 - c_2)^T U_w \Sigma_w^{-1} U_w^T z$$

where $S_w = U_w \Sigma_w U_w^T$ is the Eigenvalue Decomposition (EVD) of S_w . Since $S_w + \alpha I = U_w (\Sigma_w + \alpha I) U_w^T$, the regularized LDA gives the solution

$$g^T z = (c_1 - c_2)^T U_w (\Sigma_w + \alpha I)^{-1} U_w^T z,$$

and the regularization parameter α affects the scales of the principal components of S_w .

In the regularized LDA, the parameter α is to be optimized experimentally since no theoretical procedure for choosing an optimal parameter is easily available. Recently, a generalization of LDA through simultaneous diagonalization of S_b and S_w using the generalized singular value decomposition (GSVD) has been developed [4]. This LDA/GSVD, summarized in the next section, does not require any parameter optimization.

2.2 LDA based on the Generalized Singular Value Decomposition

Howland et al. [4, 10] applied the Generalized Singular Value Decomposition (GSVD) due to Paige and Saunders [11] to overcome the limitation of the classical LDA. When the GSVD is applied to two matrices Z_1 and Z_2 with the same number of columns, p , we obtain

$$U_1^T Z_1 X = \underbrace{[\Gamma_1]}_{\gamma} \underbrace{[0]}_{p-\gamma} \quad \text{and} \quad U_2^T Z_2 X = \underbrace{[\Gamma_2]}_{\gamma} \underbrace{[0]}_{p-\gamma} \quad \text{for} \quad \gamma = \text{rank} \left(\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix} \right)$$

where U_1 and U_2 are orthogonal and X is nonsingular, $\Gamma_1^T \Gamma_1 + \Gamma_2^T \Gamma_2 = I_\gamma$ and $\Gamma_1^T \Gamma_1$ and $\Gamma_2^T \Gamma_2$ are diagonal matrices with nonincreasing and nondecreasing diagonal components respectively.

The method in [4] utilized the representations of the scatter matrices

$$S_b = H_b H_b^T, \quad S_w = H_w H_w^T, \quad \text{and} \quad S_t = H_t H_t^T, \quad \text{where} \quad (7)$$

$$H_b = [\sqrt{n_1}(c_1 - c), \dots, \sqrt{n_r}(c_r - c)] \in \mathbb{R}^{m \times r}, \quad (8)$$

$$H_w = [A_1 - c_1 e_1, \dots, A_r - c_r e_r] \in \mathbb{R}^{m \times n}, \quad (9)$$

$$H_t = [a_1 - c, \dots, a_n - c] \in \mathbb{R}^{m \times n}, \quad (10)$$

and $e_i = [1, \dots, 1] \in \mathbb{R}^{1 \times n_i}$. Suppose the GSVD is applied to the matrix pair (H_b^T, H_w^T) and we obtain

$$U_b^T H_b^T X = [\Gamma_b \quad 0] \quad \text{and} \quad U_w^T H_w^T X = [\Gamma_w \quad 0]. \quad (11)$$

From (11) and $\Gamma_b^T \Gamma_b + \Gamma_w^T \Gamma_w = I_s$ for $s = \text{rank} \left(\begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \right)$,

$$X^T S_b X = \begin{bmatrix} \Gamma_b^T \Gamma_b & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} I_\mu & & & \\ & D_\tau & & \\ & & 0_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix} \quad \text{and} \quad (12)$$

$$X^T S_w X = \begin{bmatrix} \Gamma_w^T \Gamma_w & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} 0_\mu & & & \\ & E_\tau & & \\ & & I_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix}, \quad (13)$$

where the subscripts in I and 0 denote the size of square identity and zero matrices. Denoting the diagonal elements in $\Gamma_b^T \Gamma_b$ as η_i 's and the diagonal elements in $\Gamma_w^T \Gamma_w$ as ζ_i 's, we have

$$\zeta_i S_b x_i = \eta_i S_w x_i \quad i = 1, \dots, m, \quad (14)$$

where x_i is the column vectors of X . Note that $x_i, i = s+1, \dots, m$, belong to $\text{null}(S_b) \cap \text{null}(S_w)$. Hence η_i and ζ_i for $i = s+1, \dots, m$ in Eq. (14) can be any arbitrary numbers.

By partitioning X in (12 - 13) as

$$X = [\underbrace{X_1}_{\mu} \underbrace{X_2}_{\tau} \underbrace{X_3}_{s-\mu-\tau} \underbrace{X_4}_{m-s}], \quad (15)$$

the generalized eigenvalues and eigenvectors obtained by the GSVD can be classified as shown in Table 1. For the last $m - s$ vectors x belonging to $\text{null}(S_w) \cap \text{null}(S_b)$,

$$\begin{aligned} 0 &= x^T S_b x = (x^T H_b)(H_b^T x) = \|x^T H_b\|^2 = \sum_{i=1}^r n_i |x^T c_i - x^T c|^2 \quad \text{and} \\ 0 &= x^T S_w x = \sum_{j=1}^n |x^T a_j - x^T c_i|^2 \quad \text{where } a_j \text{ belongs to a class } i. \end{aligned}$$

Hence

$$\begin{cases} x^T c_i = x^T c & \text{for } i = 1, \dots, r \\ x^T a_j = x^T c_i & \text{for all } a_j \text{ in a class } i, \end{cases} \quad (16)$$

therefore

$$X_4^T z = X_4^T c \quad (17)$$

for any given data item $z = a_i$. This implies that the vectors $x_i, i = s+1, \dots, m$, belonging to $\text{null}(S_b) \cap \text{null}(S_w)$ do not convey discriminative information among the classes, even though the corresponding eigenvalues are not necessarily zeros. Since $\text{rank}(S_b) \leq r - 1$, from Eqs. (12-13) we have

$$x_i^T S_b x_i = 0 \quad \text{and} \quad x_i^T S_w x_i = 1 \quad \text{for } r \leq i \leq s,$$

	η_i	ζ_i	$\lambda_i = \frac{\eta_i}{\zeta_i}$	x_i belongs to
$1 \leq i \leq \mu$	1	0	∞	$\text{null}(S_w) \cap \text{null}(S_b)^c$
$\mu + 1 \leq i \leq \mu + \tau$	$1 > \eta_i > 0$	$0 < \zeta_i < 1$	$\infty > \lambda_i > 0$	$\text{null}(S_w)^c \cap \text{null}(S_b)^c$
$\mu + \tau + 1 \leq i \leq s$	0	1	0	$\text{null}(S_w)^c \cap \text{null}(S_b)$
$s + 1 \leq i \leq m$	any value	any value	any value	$\text{null}(S_w) \cap \text{null}(S_b)$

Table 1: Generalized eigenvalues λ_i 's and eigenvectors x_i 's from the GSVD. The superscript c denotes the complement.

and the between-class scatter becomes zero by the projection onto the vector x_i . Hence $r - 1$ leftmost columns of X gives an optimal transformation G_h^T for LDA. This method is called LDA/GSVD.

An Efficient Algorithm for LDA/GSVD

The algorithm to compute the GSVD for the pair (H_b^T, H_w^T) was presented in [4] as follows.

1. Compute the Singular Value Decomposition (SVD) of $Z = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \in \mathbb{R}^{(r+n) \times m}$:
 $Z = P \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} U^T$ where $s = \text{rank}(Z)$ and $P \in \mathbb{R}^{(r+n) \times (r+n)}$ and $U \in \mathbb{R}^{m \times m}$ are orthogonal and the diagonal components of $\Lambda \in \mathbb{R}^{s \times s}$ is nonincreasing.
2. Compute V from the SVD of $P(1 : r, 1 : s)^1$, which is $P(1 : r, 1 : s) = WTV^T$.
3. Compute the first $r - 1$ columns of $X = U \begin{bmatrix} \Lambda^{-1}V & 0 \\ 0 & I \end{bmatrix}$, and assign them to the transformation matrix G_h .

Now we show that this algorithm can be computed rather simply, producing an efficient and intuitive approach for LDA/GSVD. Since $\Gamma_b^T \Gamma_b + \Gamma_w^T \Gamma_w = I_s$, from (12-13), we have

$$X^T S_t X = X^T S_b X + X^T S_w X = \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} \quad (18)$$

¹The notation $P(1 : r, 1 : s)$ which may appear as a MATLAB shorthand denotes a submatrix of P composed of the components from the first to the r -th row and from the first to s -th column.

Algorithm 1 An efficient algorithm for 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 $\tilde{S}_b \equiv \Sigma_1^{-1/2} U_1^T S_b U_1 \Sigma_1^{-1/2}$: $\tilde{S}_b = V \Gamma_b^T \Gamma_b V^T$.
 3. Assign the first $r - 1$ columns of $U_1 \Sigma_1^{-1/2} V$ to G_h .
-

where $s = \text{rank}(Z)$. Eq. (18) implies $s = \text{rank}(S_t)$ and from the step 3 in the LDA/GSVD algorithm

$$S_t = X^{-T} \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} X^{-1} = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} U^T, \quad \Sigma_1 = \Lambda^T \Lambda \quad (19)$$

which results in the EVD of S_t . Partitioning U as $U = \begin{bmatrix} \underbrace{U_1}_s & \underbrace{U_2}_{m-s} \end{bmatrix}$, we have

$$X = U \begin{bmatrix} \Lambda^{-1} V & 0 \\ 0 & I \end{bmatrix} = \begin{bmatrix} U_1 \Sigma_1^{-1/2} V & U_2 \end{bmatrix}. \quad (20)$$

By substituting X in (12) with Eq. (20),

$$\Sigma_1^{-1/2} U_1^T S_b U_1 \Sigma_1^{-1/2} = V \Gamma_b^T \Gamma_b V^T. \quad (21)$$

Note that the optimal transformation matrix G_h by LDA/GSVD is obtained by the leftmost $r - 1$ columns of X , which are the leftmost $r - 1$ columns of $U_1 \Sigma_1^{-1/2} V$. Eqs. (19) and (21) show that U_1 and Σ_1 can be computed from the EVD of S_t and V from the EVD of $\Sigma_1^{-1/2} U_1^T S_b U_1 \Sigma_1^{-1/2}$. This new approach for LDA/GSVD is summarized in Algorithm 1.

In Algorithm 1, the matrices U_1 and Σ_1 in the EVD of $S_t \in \mathbb{R}^{m \times m}$ can be obtained by the EVD of $H_t^T H_t \in \mathbb{R}^{n \times n}$ instead of $H_t H_t^T \in \mathbb{R}^{m \times m}$ [1] by which computational complexity can be reduced from $O(m^3)$ to $O(n^3)$. Especially when m is much bigger than n , computational savings become great. Let the EVD of $H_t^T H_t$ be

$$H_t^T H_t = \begin{bmatrix} \underbrace{J_1}_s & \underbrace{J_2}_{n-s} \end{bmatrix} \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J_1^T \\ J_2^T \end{bmatrix}, \quad (22)$$

where $s = \text{rank}(H_t) = \text{rank}(S_t)$. From (22)

$$S_t(H_t J_1) = H_t(H_t^T H_t)J_1 = (H_t J_1)D_1,$$

and therefore the columns in $H_t J_1$ are eigenvectors of S_t corresponding to nonzero eigenvalues in the diagonal of D_1 . Since $(H_t J_1)^T (H_t J_1) = D_1$, we obtain the orthonormal eigenvectors and corresponding nonzero eigenvalues of S_t by $H_t J_1 D_1^{-1/2}$ and D_1 , which are U_1 and Σ_1 respectively. In this new approach, we just need to compute the EVD of a much smaller $n \times n$ matrix $H_t^T H_t$ instead of $m \times m$ matrix $S_t = H_t H_t^T$ when $m \gg n$. However, in the regularized LDA or the method by Chen et al. which is presented next, we can not resort to this approach. The regularized LDA needs the entire m eigenvectors of S_w and the method based on the projection to $\text{null}(S_w)$ needs to compute a basis of $\text{null}(S_w)$ which are eigenvectors corresponding to zero eigenvalues.

Two-Class Problem

Now we consider the two-class problem in LDA/GSVD. By Eq. (5), we have

$$\begin{aligned} \Sigma_1^{-1/2} U_1^T S_b U_1 \Sigma_1^{-1/2} &= \Sigma_1^{-1/2} U_1^T \rho (c_1 - c_2)(c_1 - c_2)^T U_1 \Sigma_1^{-1/2} \\ &= \left(\frac{w}{\|w\|_2} \right) \rho \|w\|_2^2 \left(\frac{w}{\|w\|_2} \right)^T, \end{aligned}$$

where $\rho = n_1 n_2 / n$ and $w = \Sigma_1^{-1/2} U_1^T (c_1 - c_2)$. Hence the transformation matrix $g \in \mathbb{R}^{m \times 1}$ is given by

$$g = \nu U_1 \Sigma_1^{-1/2} w = \nu U_1 \Sigma_1^{-1} U_1^T (c_1 - c_2)$$

for some scalar ν , and the dimension reduced representation of any data item z is given by

$$g^T z = \nu (c_1 - c_2)^T U_1 \Sigma_1^{-1} U_1^T z = \nu (c_1 - c_2)^T S_t^+ z,$$

where S_t^+ denotes the pseudoinverse of S_t . When S_w is nonsingular, by applying the Sherman-Morrison formula [12] to $S_t = S_w + S_b$, we have

$$S_t^{-1} = (S_w + \rho (c_1 - c_2)(c_1 - c_2)^T)^{-1} = S_w^{-1} - \frac{S_w^{-1} \rho (c_1 - c_2)(c_1 - c_2)^T S_w^{-1}}{1 + \rho (c_1 - c_2)^T S_w^{-1} (c_1 - c_2)}$$

and

$$g^T z = \nu(c_1 - c_2)^T S_t^{-1} z = \nu_1(c_1 - c_2)^T S_w^{-1} z \quad (23)$$

for a scalar $\nu_1 = \nu / (1 + \rho(c_1 - c_2)^T S_w^{-1} (c_1 - c_2))$. Eq. (23) shows that LDA/GSVD is equal to the classical LDA when S_w is nonsingular.

In face recognition, in the efforts to overcome the singularity of scatter matrices caused by high dimensionality, some methods have been proposed [5, 6]. The basic principle of the algorithms proposed in [5, 6] is that the transformation using a basis of either $\text{range}(S_b)$ or $\text{null}(S_w)$ is performed in the first stage and then in the transformed space the second projective directions are searched. These methods are summarized in the next two sections where we also present their algebraic relationships.

2.3 A Method based on the Projection onto $\text{null}(S_w)$

Chen et al. [5] proposed a generalized method of LDA which solves undersampled problems and applied it for face recognition. The method projects the original space onto the null space of S_w using an orthonormal basis of $\text{null}(S_w)$, and then in the projected space, a transformation that maximizes the between-class scatter is computed.

Consider the SVD of $S_w \in \mathbb{R}^{m \times m}$,

$$S_w = U_w \Sigma_w U_w^T.$$

Partitioning U_w as $U_w = [\underbrace{U_{w1}}_{s_1} \underbrace{U_{w2}}_{m-s_1}]$ where $s_1 = \text{rank}(S_w)$,

$$\text{null}(S_w) = \text{span}(U_{w2}). \quad (24)$$

First, the transformation by $U_{w2} U_{w2}^T$ projects the original data to $\text{null}(S_w)$. Then, the eigenvectors corresponding to the largest eigenvalues of the between-class scatter matrix \tilde{S}_b in the projected space are found. Let the EVD of $\tilde{S}_b \equiv U_{w2} U_{w2}^T S_b U_{w2} U_{w2}^T$ be

$$\tilde{S}_b = \tilde{U}_b \tilde{\Sigma}_b \tilde{U}_b^T = [\underbrace{\tilde{U}_{b1}}_{s_2} \underbrace{\tilde{U}_{b2}}_{m-s_2}] \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix}, \quad (25)$$

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

$$G_e = U_{w2} U_{w2}^T \tilde{U}_{b1}. \quad (26)$$

Let us call this method *To-N*(S_w) as an abbreviation.

Two-Class Problem

In the two-class problem, S_b is expressed as in (5) and

$$\tilde{S}_b = U_{w2} U_{w2}^T \rho (c_1 - c_2)(c_1 - c_2)^T U_{w2} U_{w2}^T = \left(\frac{w}{\|w\|_2} \right) \rho \|w\|_2^2 \left(\frac{w}{\|w\|_2} \right)^T$$

where $\rho = n_1 n_2 / n$ and $w = U_{w2} U_{w2}^T (c_1 - c_2) \in \mathbb{R}^{m \times 1}$. Hence the transformation matrix $g \in \mathbb{R}^{m \times 1}$ is obtained by

$$g = U_{w2} U_{w2}^T \frac{w}{\|w\|_2} = \nu U_{w2} U_{w2}^T (c_1 - c_2)$$

with $\nu = 1/\|w\|_2$. For any data item $z \in \mathbb{R}^{m \times 1}$, the dimension reduced representation is given by

$$g^T z = \nu (c_1 - c_2)^T U_{w2} U_{w2}^T z.$$

Relationship with LDA/GSVD

From (25), we have

$$\begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix} U_{w2} U_{w2}^T S_b U_{w2} U_{w2}^T [\tilde{U}_{b1} \tilde{U}_{b2}] = \begin{bmatrix} \tilde{\Sigma}_{b1} & 0 \\ 0 & 0 \end{bmatrix}, \quad (27)$$

$$\begin{bmatrix} \tilde{U}_{b1}^T \\ \tilde{U}_{b2}^T \end{bmatrix} U_{w2} U_{w2}^T S_w U_{w2} U_{w2}^T [\tilde{U}_{b1} \tilde{U}_{b2}] = 0. \quad (28)$$

The second equation holds due to (24). Eqs. in (27-28) imply that the column vectors of G_e given in (26) belong to $\text{null}(S_w) \cap \text{null}(S_b)^c$ and they are discriminative vectors, since the transformation by these vectors minimizes the within-class scatter to zero and increases the between-class scatter. The top row of Table 1 shows that the LDA/GSVD solution also

includes the vectors from $\text{null}(S_w) \cap \text{null}(S_b)^c$. Based on this observation, this method $To-N(S_w)$ can be compared with LDA/GSVD. By denoting X in LDA/GSVD as

$$X = [\underbrace{X_1}_{\mu} \underbrace{X_2}_{\tau} \underbrace{X_3}_{s-\mu-\tau} \underbrace{X_4}_{m-s}], \quad (29)$$

we find a relationship between X_1 and $G_e = U_{w2}U_{w2}^T\tilde{U}_{b1}$.

Eq. (13) implies that $[X_1 \ X_4]$ is a basis of $\text{null}(S_w)$. Hence any vector in $\text{null}(S_w)$ can be represented as a linear combination of column vectors in $[X_1 \ X_4]$. The following Theorem shows the condition for any vector in $\text{null}(S_w)$ to belong to $\text{null}(S_w) \cap \text{null}(S_b)^c$.

THEOREM 1 *Any vector x belongs to $\text{null}(S_w) \cap \text{null}(S_b)^c$ if and only if x is represented as $X_1h + X_4k$ where $h \neq 0 \in \mathbb{R}^{\mu \times 1}$ and $k \in \mathbb{R}^{(m-s) \times 1}$.*

Proof. Let $x \in \text{null}(S_w) \cap \text{null}(S_b)^c$. Since $[X_1 \ X_4]$ is a basis of $\text{null}(S_w)$, $x = X_1h + X_4k$ for some $h \in \mathbb{R}^{\mu \times 1}$ and $k \in \mathbb{R}^{(m-s) \times 1}$. Suppose $h = 0$. Then $x = X_4k \in \text{null}(S_w) \cap \text{null}(S_b)$, which contradicts to $x \in \text{null}(S_w) \cap \text{null}(S_b)^c$. Hence $h \neq 0$.

Now let us prove that if $h \neq 0$ then $x = X_1h + X_4k$ belongs to $\text{null}(S_w) \cap \text{null}(S_b)^c$. Since $x = X_1h + X_4k \in \text{null}(S_w)$, it is enough to show $x \notin \text{null}(S_b)$. From (12),

$$x^T S_b x = (X_1h)^T S_b (X_1h) = h^T (X_1^T S_b X_1) h = h^T I_\mu h = \|h\|_2^2 \neq 0. \quad \square$$

By Theorem 1,

$$U_{w2}U_{w2}^T\tilde{U}_{b1} = X_1H + X_4K$$

for some matrices $H \in \mathbb{R}^{\mu \times s_2}$ and $K \in \mathbb{R}^{(m-s) \times s_2}$ with $s_2 = \text{rank}(\tilde{S}_b)$, where each column of H is nonzero. Hence for any data item $z \in \mathbb{R}^{m \times 1}$, the reduced dimensional representation by $G_e = U_{w2}U_{w2}^T\tilde{U}_{b1}$ is given as

$$G_e^T z = H^T X_1^T z + K^T X_4^T z. \quad (30)$$

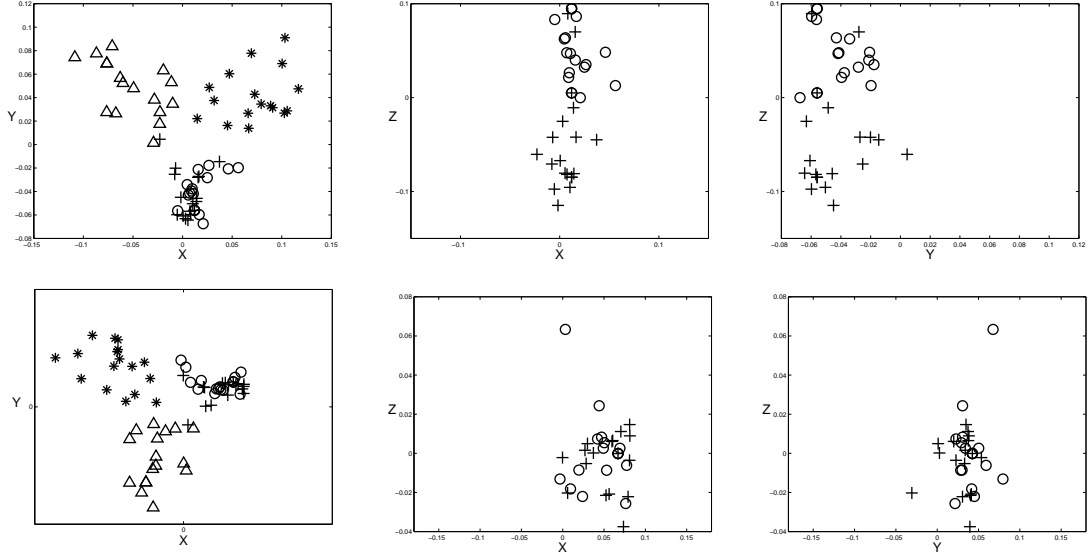


Figure 1: The visualization of the data in the reduced dimensional spaces by LDA/GSVD (figures in the first row) and the method $To-N(S_w)$ (figures in the second row).

As explained in (16) of Section 2.2, since all data items are transformed to one point by x^T for $x \in \text{null}(S_w) \cap \text{null}(S_b)$, the second part $K^T X_4^T z$ in (30) corresponds to the translation which does not affect the classification performance.

While the transformation matrix $G_e = U_{w2} U_{w2}^T \tilde{U}_{b1}$ by the method $To-N(S_w)$ is related to X_1 of LDA/GSVD as in (30), the main difference between the two methods is due to the eigenvectors in $\text{null}(S_w)^c \cap \text{null}(S_b)^c$, which correspond to the second row in Table 1. The projection to $\text{null}(S_w)$ by $U_{w2} U_{w2}^T$ excludes vectors in $\text{null}(S_w)^c$, and therefore $\text{null}(S_w)^c \cap \text{null}(S_b)^c$. When

$$\text{rank}(\tilde{S}_b) < \text{rank}(S_b) \leq r - 1$$

where r is the number of classes, the reduced dimension by $G_e = U_{w2} U_{w2}^T \tilde{U}_{b1}$ is $\text{rank}(\tilde{S}_b)$, therefore less than $r - 1$, while LDA/GSVD includes $r - 1$ vectors from both $\text{null}(S_w) \cap \text{null}(S_b)^c$ and $\text{null}(S_w)^c \cap \text{null}(S_b)^c$. In order to demonstrate this case, we conducted an experiment using data in text classification, of which characteristics will be discussed in detail in the section for experiments. The data was collected from Reuters-21578 database

and contains 4 classes. Each class has 80 samples and the data dimension is 2412. After splitting the dataset randomly to training data and test data with a ratio of 4:1, the linear transformations by LDA/GSVD and the method $To-N(S_w)$ were computed by using training data. While the rank of S_b was 3, the rank of \tilde{S}_b was 2 in this dataset. Hence the reduced dimension by the method $To-N(S_w)$ due to Chen et al. was 2. On the other hand, LDA/GSVD produced two eigenvectors from $\text{null}(S_w) \cap \text{null}(S_b)^c$ and one eigenvector from $\text{null}(S_w)^c \cap \text{null}(S_b)^c$, resulting in the reduced dimension 3. Figure 1 illustrates the reduced dimensional spaces by both methods. The top three figures were generated by LDA/GSVD. For the visualization, the data reduced to 3-dimensional space by LDA/GSVD was projected to 2-dimensional spaces, x - y , x - z and y - z spaces, respectively. In x - y space, two classes (\triangle and $*$) are well separated, while two other classes (O and $+$) are mixed together. However, as shown in the second and third figures, two classes mixed in x - y space are separated in x - z and y - z spaces along z axis. This shows the third eigenvector from $\text{null}(S_w)^c \cap \text{null}(S_b)^c$ improves the separation of classes. The bottom three figures were generated by the method based on the projection to $\text{null}(S_w)$. Since $\text{rank}(\tilde{S}_b)=2$, the reduced dimension by that method was 2 and the first figure illustrates the reduced dimensional space. The second and third figures show that adding one more column vector from $U_{w2}U_{w2}^T\tilde{U}_{b2}$ and increasing the reduced dimension to 3 does not improve the separation of classes mixed in x - y space, since the one extra dimension comes from $\text{null}(S_w) \cap \text{null}(S_b)$. On the other hand, when

$$\text{rank}(\tilde{S}_b) = \text{rank}(S_b) = r - 1,$$

both LDA/GSVD and the method $To-N(S_w)$ obtain transformation matrices G_h and G_e from $\text{null}(S_w) \cap \text{null}(S_b)^c$. Then the difference between two methods comes from the diagonal components of I_{r-1} and $\tilde{\Sigma}_{b1}$ in

$$G_h^T S_b G_h = I_{r-1} \quad \text{and} \quad G_e^T S_b G_e = \tilde{\Sigma}_{b1}$$

where $\tilde{\Sigma}_{b1}$ has nonincreasing diagonal components. As shown in the experimental results of Section 2.7, the effects of different scaling in the diagonal components may depend on the characteristics of data.

2.4 A Method based on the Transformation by a Basis of $\text{range}(S_b)$

In this section, we review another two-step approach by Yu and Yang [6] proposed to handle undersampled problems, and illustrate its relationship to other methods. Contrary to the method discussed in Section 2.3, the method presented in this section first transforms the original space by using a basis of $\text{range}(S_b)$, and then in the transformed space the minimization of within-class scatter is pursued.

Consider the EVD of S_b ,

$$S_b = U_b \Sigma_b U_b^T = \underbrace{[U_{b1}]}_{s_1} \underbrace{[U_{b2}]}_{m-s_1} \begin{bmatrix} \Sigma_{b1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{b1}^T \\ U_{b2}^T \end{bmatrix},$$

where U_b is orthogonal, $\text{rank}(S_b) = s_1$ and Σ_{b1} is a diagonal matrix with nonincreasing positive diagonal components. Then $\text{range}(S_b) = \text{span}(U_{b1})$. In the method by Yu and Yang, the original data is first transformed to an s_1 -dimensional space by $V_y = U_{b1} \Sigma_{b1}^{-1/2}$.

Then the between-class scatter matrix \tilde{S}_b in the transformed space becomes

$$\tilde{S}_b \equiv V_y^T S_b V_y = I_{s_1}.$$

Now consider the EVD of $\tilde{S}_w \equiv V_y^T S_w V_y$,

$$\tilde{S}_w = \tilde{U}_w \tilde{\Sigma}_w \tilde{U}_w^T, \tag{31}$$

where $\tilde{U}_w \in \mathbb{R}^{s_1 \times s_1}$ is orthogonal and $\tilde{\Sigma}_w \in \mathbb{R}^{s_1 \times s_1}$ is a diagonal matrix. Then

$$\tilde{U}_w^T V_y^T S_b V_y \tilde{U}_w = I_{s_1} \quad \text{and} \quad \tilde{U}_w^T V_y^T S_w V_y \tilde{U}_w = \tilde{\Sigma}_w. \tag{32}$$

In most applications, $\text{rank}(S_w)$ is greater than $\text{rank}(S_b)$, and $\tilde{\Sigma}_w$ is nonsingular since

$$\text{rank}(\tilde{U}_w^T V_y^T S_w V_y \tilde{U}_w) = \text{rank}(S_w) \geq \text{rank}(S_b) = \text{rank}(\tilde{U}_w^T V_y^T S_b V_y \tilde{U}_w) = s_1.$$

Face data	Transformation matrix		
	$G_y = V_y$	$G_y = V_y \tilde{U}_w$	$G_y = V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}$
AT&T	94.3	94.3	99.0
Yale	80.6	80.6	89.7

Table 2: The prediction accuracies(%).

Scaling (32) by $\tilde{\Sigma}_w^{-1/2}$, we have

$$(\tilde{\Sigma}_w^{-1/2} \tilde{U}_w^T V_y^T) S_b (V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}) = \tilde{\Sigma}_w^{-1}, \quad (\tilde{\Sigma}_w^{-1/2} \tilde{U}_w^T V_y^T) S_w (V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}) = I_{s_1}. \quad (33)$$

The authors in [6] proposed the transformation matrix

$$G_y = V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}.$$

Eqs. in (33) imply that each column of G_y belongs to $\text{null}(S_w)^c \cap \text{null}(S_b)^c$. We call this method *To-R*(S_b) for short.

Two-Class Problem

In a two-class problem, since

$$S_b = \rho(c_1 - c_2)(c_1 - c_2)^T = \left(\frac{c_1 - c_2}{\|c_1 - c_2\|_2} \right) \rho \|c_1 - c_2\|_2^2 \left(\frac{c_1 - c_2}{\|c_1 - c_2\|_2} \right)^T$$

where $\rho = n_1 n_2 / n$, a data item is transformed to the 1-dimensional space by $g = \frac{c_1 - c_2}{\sqrt{\rho} \|c_1 - c_2\|_2}$.

The dimension reduced representation of any data item z is given by $g^T z = \nu(c_1 - c_2)^T z$ for some scalar ν . Note that no minimization of within-class scatter in the transformed space is possible.

The optimization criteria by J_2 and J_3 in (2) are invariant under any nonsingular linear transformation, i.e. for any nonsingular matrix F whose order is the same as that of the column dimension of G ,

$$J_i(G) = J_i(GF), \quad i = 2, 3, \quad (34)$$

while the objective function J_1 is not. Hence in the transformation matrix $G_y = V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}$ obtained by the method $To-R(S_b)$, none of the components $\tilde{\Sigma}_w^{-1/2}$ and $\tilde{U}_w \tilde{\Sigma}_w^{-1/2}$ involved in the second step (those in (31-33)) improves the optimization criteria by J_2 and J_3 . However, the following experimental results show that the scaling by $\tilde{\Sigma}_w^{-1/2}$ can make dramatic effects on the classification performances. Postponing the detailed explanation on the data sets and experimental setting until Section 2.7, experimental results on the face recognition data sets are shown in Table 2. After dimension reduction, 1-NN classifier was used in the reduced dimensional space.

2.5 A Method of PCA plus Transformations to $\text{range}(S_w)$ and $\text{null}(S_w)$

As shown in the analysis of the compared methods, they search for discriminative vectors in $\text{null}(S_w) \cap \text{null}(S_b)^c$ and $\text{null}(S_w)^c \cap \text{null}(S_b)^c$. The method $To-N(S_w)$ by Chen et al. finds solution vectors in $\text{null}(S_w) \cap \text{null}(S_b)^c$ and $To-R(S_b)$ by Yu et al. restricts the search space to $\text{null}(S_w)^c \cap \text{null}(S_b)^c$. LDA/GSVD by Howland et al. finds solution from both spaces, however the number of possible discriminative vectors can not be greater than $\text{rank}(S_b)$, possibly resulting in solution vectors only from $\text{null}(S_w) \cap \text{null}(S_b)^c$ in the case of high dimensional data. Recently Yang et al. [7] have proposed a method to obtain solution vectors in both spaces, which we will call $To-NR(S_w)$.

In the method by Yang et al., first, the transformation by the orthonormal basis of $\text{range}(S_t)$, as in PCA, is performed. Let the SVD of S_t be

$$S_t = U_t \Sigma_t U_t^T = [\underbrace{U_{t1}}_s \underbrace{U_{t2}}_{m-s}] \begin{bmatrix} \Sigma_{t1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix}$$

where $s = \text{rank}(S_t)$. In the transformed space by U_{t1} , let the within-scatter matrix be $\tilde{S}_w = U_{t1}^T S_w U_{t1}$. Then the basis of $\text{null}(\tilde{S}_w)$ and $\text{range}(\tilde{S}_w)$ can be found by the EVD of \tilde{S}_w as

$$\tilde{S}_w = \tilde{U}_w \tilde{\Sigma}_w \tilde{U}_w^T = [\tilde{U}_{w1} \tilde{U}_{w2}] \begin{bmatrix} \tilde{\Sigma}_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{w1}^T \\ \tilde{U}_{w2}^T \end{bmatrix}. \quad (35)$$

In the transformed space by the basis \tilde{U}_{w2} of $\text{null}(\tilde{S}_w)$, let Y be the matrix whose columns are the eigenvectors corresponding to nonzero eigenvalues of

$$\bar{S}_b \equiv \tilde{U}_{w2}^T U_{t1}^T S_b U_{t1} \tilde{U}_{w2}. \quad (36)$$

On the other hand, in the transformed space by the basis \tilde{U}_{w1} of $\text{range}(\tilde{S}_w)$, let Z be the matrix whose columns are the eigenvectors² with the k largest nonzero eigenvalues of $\hat{S}_t^{-1} \hat{S}_b$ where $\hat{S}_b \equiv \tilde{U}_{w1}^T U_{t1}^T S_b U_{t1} \tilde{U}_{w1}$ and $\hat{S}_t \equiv \tilde{U}_{w1}^T U_{t1}^T S_t U_{t1} \tilde{U}_{w1}$. Then the transformation matrix by the method *To-NR*(S_w) is constructed as

$$G_d = [U_{t1} \tilde{U}_{w2} Y \quad U_{t1} \tilde{U}_{w1} Z]. \quad (37)$$

When two parts $U_{t1} \tilde{U}_{w2} Y$ and $U_{t1} \tilde{U}_{w1} Z$ are used for transformation matrix G_d , it will be better to normalize the columns in $U_{t1} \tilde{U}_{w1} Z$ so that effects of both parts can be balanced.

Relationship with the method *To-N*(S_w)

Recall from Section 2.3 that the method *To-N*(S_w) projects the original space onto the null space of S_w using an orthonormal basis of $\text{null}(S_w)$, and then in the projected space, a transformation that maximizes the between-class scatter is computed.

Since U_{t2} is a basis of $\text{null}(S_t)$ and $\text{null}(S_t) \subset \text{null}(S_w)$, from (35)

$$\begin{bmatrix} U_{t1}^T \\ U_{t2}^T \end{bmatrix} S_w \begin{bmatrix} U_{t1} & U_{t2} \end{bmatrix} = \begin{bmatrix} \tilde{U}_w^T \tilde{\Sigma}_w \tilde{U}_w^T & 0 \\ 0 & 0 \end{bmatrix}. \quad (38)$$

By Eq. (38), we can obtain the EVD of S_w as

$$\begin{aligned} S_w &= \begin{bmatrix} U_{t1} \tilde{U}_w & U_{t2} \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_w & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_w^T U_{t1}^T \\ U_{t2}^T \end{bmatrix} \\ &= \begin{bmatrix} U_{t1} \tilde{U}_{w1} & U_{t1} \tilde{U}_{w2} & U_{t2} \end{bmatrix} \begin{bmatrix} \tilde{\Sigma}_{w1} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \tilde{U}_{w1}^T U_{t1}^T \\ \tilde{U}_{w2}^T U_{t1}^T \\ U_{t2}^T \end{bmatrix}. \end{aligned} \quad (39)$$

²In [7], it was claimed that the orthonormal eigenvectors of $\hat{S}_t^{-1} \hat{S}_b$ should be used. However, $\hat{S}_t^{-1} \hat{S}_b$ may not be symmetric therefore it is not guaranteed that there exist orthonormal eigenvectors of $\hat{S}_t^{-1} \hat{S}_b$.

Eq. (39) shows that the columns of $V \equiv [U_{t1}\tilde{U}_{w2} \ U_{t2}]$ is an orthonormal basis of $\text{null}(S_w)$. Hence the transformation by VV^T gives the projection onto the null space of S_w .

Now by the notation (36) and $\text{span}(U_{t2}) = \text{null}(S_t) \subset \text{null}(S_b)$,

$$[U_{t1}\tilde{U}_{w2} \ U_{t2}] \begin{bmatrix} (U_{t1}\tilde{U}_{w2})^T \\ U_{t2}^T \end{bmatrix} S_b [U_{t1}\tilde{U}_{w2} \ U_{t2}] \begin{bmatrix} (U_{t1}\tilde{U}_{w2})^T \\ U_{t2}^T \end{bmatrix} = U_{t1}\tilde{U}_{w2}\bar{S}_b\tilde{U}_{w2}^T U_{t1}^T$$

which is the between-class scatter matrix in the projected space by VV^T . Let the EVD of \bar{S}_b be

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

Then we have the transformation matrix G_e by the method $To-N(S_w)$ as

$$G_e = [U_{t1}\tilde{U}_{w2} \ U_{t2}] \begin{bmatrix} (U_{t1}\tilde{U}_{w2})^T \\ U_{t2}^T \end{bmatrix} U_{t1}\tilde{U}_{w2}\bar{U}_{b1} = U_{t1}\tilde{U}_{w2}\bar{U}_{b1} \quad (40)$$

which is exactly same as $U_{t1}\tilde{U}_{w2}Y$ in G_d of (37).

2.6 Other Approaches for generalized LDA

2.6.1 PCA plus LDA

Using PCA as a preprocessing step before applying LDA has been a traditional technique for undersampled problems and successfully applied for face recognition [13]. In this approach, data dimension is reduced by PCA so that in the reduced dimensional space the within-class scatter matrix becomes nonsingular and classical LDA can be performed. However, choosing optimal dimensions reduced by PCA is not easy and experimental process for it can be expensive.

2.6.2 GSLDA

Zheng et al. claimed that the most discriminant vectors for LDA can be chosen from

$$\text{null}(S_t)^\perp \cap \text{null}(S_w) \quad (41)$$

where $\text{null}(S_t)^\perp$ denotes the orthogonal complement of $\text{null}(S_t)$ [8]. They also proposed a computationally efficient method called GSLDA [14] which uses the modified Gram-Schmidt Orthogonalization (MGS) in order to obtain an orthogonal basis of $\text{null}(S_t)^\perp \cap \text{null}(S_w)$. In [14], under the assumption that the given data items are independent, MGS is applied to

$$[H_w^*, H_b^*] \quad (42)$$

obtaining an orthogonal basis Q of (42), where H_w^* is constructed by deleting one column from each subblock $A_i - c_i e_1$, $1 \leq i \leq r$, in H_w and $H_b^* = [c_1 - c, \dots, c_{r-1} - c]$. Then the last $r - 1$ columns of Q give an orthogonal basis of (41). When applying L_2 -norm as a similarity measure, using any orthogonal basis of $\text{null}(S_t)^\perp \cap \text{null}(S_w)$ as a transformation matrix gives the same classification performances [14].

In Section 2.5, it was shown that a transformation matrix G_e by the method $To-N(S_w)$ is same as the first part $U_{t1}\tilde{U}_{w2}Y$ in the transformation matrix G_d by the method $To-NR(S_w)$. In fact, it is not difficult to prove that under the assumption of the independence of data items, $U_{t1}\tilde{U}_{w2}Y$ is an orthogonal basis of (41), and therefore prediction accuracies by the method $To-N(S_w)$ and GSLDA should be same.

2.6.3 Uncorrelated Linear discriminant analysis

Instead of the orthogonality of the columns $\{g_i\}$ in the transformation matrix G , i.e., $g_i^T g_j = 0$ for $i \neq j$, uncorrelated LDA (ULDA) imposes the S_t -orthogonal constraint, $g_i^T S_t g_j = 0$ for $i \neq j$ [15]. In [16], it was shown that discriminant vectors obtained by the LDA/GSVD solve the S_t -orthogonal constraint. Hence the proposed algorithm 1 can also give solutions for ULDA more efficiently.

Data	Re	Tr1	Tr2	Tr3	Tr4	Tr5	AT&T	Yale
Dim.	3094	5896	5825	8104	7362	8175	2576	8586
no. data	490	210	187	841	757	575	400	165
classes	5	7	4	4	5	6	40	15

Table 3: The description of data sets

2.7 Experimental Comparisons of Generalized LDA Algorithms

In order to compare the discussed methods, we conducted extensive experiments using two types of data sets in text classification and face recognition.

Text classification is a task to assign a class label to a new document based on the information from pre-classified documents. A collection of documents are assumed to be represented as a term-document matrix, where each document is represented as a column vector and the components of the column vector denote frequencies of words appeared in the document. The term-document matrix is obtained after preprocessing with common words and rare term removal, stemming, term frequency and inverse term frequency weighting and normalization [17]. The term-document matrix representation often makes the high dimensionality inevitable.

For all text data sets³, they were randomly split to the training set and the test set with the ratio of 4 : 1. Experiments are repeated 10 times to obtain mean prediction accuracies and standard deviation as a performance measure. Detailed description of text data sets is given in Table 3. After computing a transformation matrix using training data, both training data and test data were represented in the reduced dimensional space. In the transformed space, the nearest neighbor classifier was applied to compute the prediction accuracies for classification. For each data item in test set, it finds the nearest neighbor from the training data set and predicts a class label for the test data according to the class label of the nearest

³The text data sets were downloaded and preprocessed from <http://www-users.cs.umn.edu/~karypis/cluto/download.html>, which were collected from Reuter-21578 and TREC-5, TREC-6, TREC-7 database.

neighbor. Table 4 reports the mean prediction accuracies from 10 times random splitting to training and test sets.

The second experiment, face recognition, is a task to identify a person based on given face images with different facial expressions, illumination and poses. Since the number of pictures for each subject is limited and the data dimension is the number of pixels of a face image, face recognition data sets are typically severely undersampled.

Our experiments used two data sets, AT&T (formerly ORL) face database and Yale face database. The AT&T database has 400 images, which consists of 10 images of 40 subjects. All the images were taken against a dark homogeneous background, with slightly varying lighting, facial expressions (open/closed eyes, smiling/non-smiling), and facial details (glasses/no-glasses). The subjects are in up-right, frontal positions with tolerance for some side movement [18]. For the manageable data sizes, the images have been downsampled from the size 92×112 to 46×56 by averaging the grey level values on 2×2 blocks. Yale face database contains 165 images, 11 images of 15 subjects. The 11 images per subject were taken under various facial expressions or configurations: center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised, and wink [19]. In our experiment, each image has been downsampled from 320×243 to 106×81 by averaging the grey values on 3×3 blocks. Detailed description of face data sets is also given in Table 3. Since the number of images for each subject is small, leave-one-out method was performed where it takes one image for test set and the remaining images are used as a training set. Each image serves as a test datum by turns and the ratio of the number of correctly classified cases and the total number of data is considered as a prediction accuracy.

Table 4 summarizes the prediction accuracies from both experiments. For the regularized LDA, we report the best among the accuracies obtained with the regularization parameter $\alpha = 0.5, 1, 1.5$. The method based on the transformation to $\text{range}(S_b)$, $To-R(S_b)$, gives

Data	RLDA	LDA/GSVD	$To-N(S_w)$	$To-R(S_b)$	$To-NR(S_w)$
Text Classification					
Re	95.8	95.1	94.5	94.2	94.7
Tr1	95.7	98.3	98.1	96.7	97.6
Tr2	87.9	90.3	91.5	88.2	91.8
Tr3	98.6	98.4	98.6	97.7	98.7
Tr4	98.0	97.3	97.0	96.3	97.1
Tr5	93.6	93.3	94.2	94.1	94.4
Face Recognition					
AT&T	98.0	93.5	98.0	99.0	98.8
Yale	97.6	98.8	97.6	89.7	98.2

Table 4: Prediction accuracies (%). For RLDA, the best accuracy among $\alpha = 0.5, 1, 1.5$ is reported. For each dataset, the best prediction accuracy is shown in boldface.

relatively low prediction accuracies compared with the methods utilizing the null space of the within-class scatter matrix S_w . While no single methods works the best in all situations, computational complexities can be dramatically different among the compared methods as we will discuss in the next section.

2.8 Analysis of Computational Complexities

In this section we analyze computational complexities for the discussed methods. The computational complexity for the SVD decomposition depends on what parts need to be explicitly computed. We use flop counts for the analysis of computational complexities where one flop (floating point operation) represents roughly what is required to do one addition/subtraction or one multiplication/division [12]. For the SVD of a matrix $H \in \mathbb{R}^{p \times q}$ when $p \gg q$,

$$H = U\Sigma V^T = \underbrace{[U_1 \ U_2]}_{\substack{q \quad p-q}} \Sigma V^T,$$

where $U \in \mathbb{R}^{p \times p}$, $\Sigma \in \mathbb{R}^{p \times q}$ and $V \in \mathbb{R}^{q \times q}$, the complexities (flops) can be roughly estimated as follows [12, pp.254].

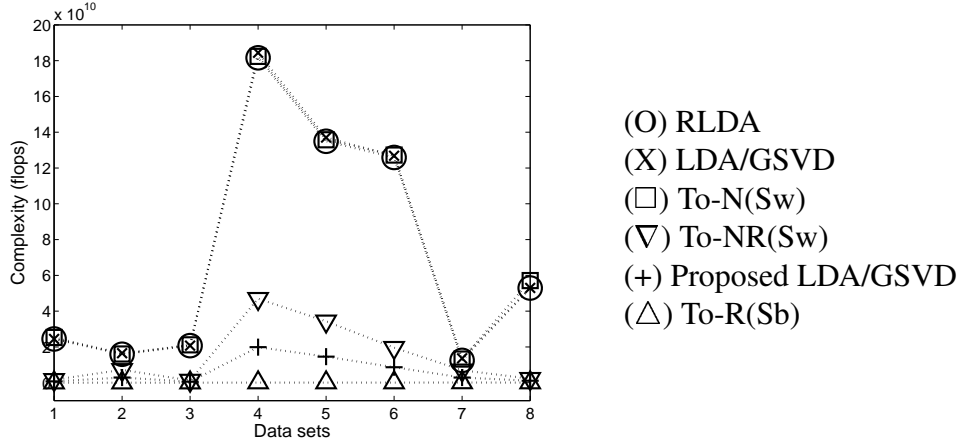


Figure 2: Comparison of computational complexities of the generalized LDA methods using the sizes of training data used in experiments. From the left on x-axis, the data sets, Tr1, Re, Tr2, Tr3, Tr4, Tr5, AT&T and Yale, are corresponded.

Need to be computed explicitly	Complexities
U_1, Σ	$6pq^2 + 11q^3$
U, Σ	$4p^2q + 13q^3$
U, Σ, V	$4p^2q + 22q^3$

For the multiplication of the $p_1 \times p_2$ matrix and the $p_2 \times p_3$ matrix, $2p_1p_2p_3$ flops can be counted.

For simplicity, cost for constructing $H_b \in \mathbb{R}^{m \times r}$, $H_w \in \mathbb{R}^{m \times n}$ and $H_t \in \mathbb{R}^{m \times n}$ in (8-10) was not included for the comparison, since the construction of scatter matrices is required in all the methods. For $H \in \mathbb{R}^{p \times q}$ and $p \gg q$, when only eigenvectors corresponding to the nonzero eigenvalues of $HH^T \in \mathbb{R}^{p \times p}$ are needed, the approach of computing the EVD of H^TH instead of HH^T as explained in Section 2.2 was utilized.

Figure 2 compares computational complexities of the discussed methods by using specific sizes of training data sets used in the experiments. As shown in Figure 2, regularized LDA, LDA/GSVD [4] and the method $To-N(S_w)$ [5] have high computational complexities overall. The method $To-R(S_b)$ [6] obtained the lowest computational costs compared with other methods while its performance can not be ranked highly. The proposed algo-

rithm for LDA/GSVD reduced the complexity of the original algorithm dramatically while it achieves competitive prediction accuracies as shown in Section 2.7. This new algorithm can save computational complexities even more when the number of terms is much greater than the number of documents.

3 Nonlinear Discriminant Analysis based on Kernel Methods

Linear dimension reduction is conceptually simple and has been used in many application areas. However, it has a limitation for the data which is not linearly separable since it is difficult to capture a nonlinear relationship with a linear mapping. In order to overcome such a limitation, nonlinear extensions of linear dimension reduction methods using kernel methods have been proposed [20, 21, 22, 23, 24, 25]. The main idea of kernel methods is that without knowing the nonlinear feature mapping or the mapped feature space explicitly, we can work on the nonlinearly transformed feature space through kernel functions. It is based on the fact that for any kernel function κ satisfying Mercer's condition, there exists a reproducing kernel Hilbert space H and a feature map Φ such that

$$\kappa(x, y) = \langle \Phi(x), \Phi(y) \rangle \quad (43)$$

where \langle, \rangle is an inner product in H [26, 9, 27].

Suppose that given a kernel function κ original data space is mapped to a feature space (possibly an infinite dimensional space) through a nonlinear feature mapping $\Phi : \mathcal{A} \subset \mathbb{R}^m \rightarrow \mathcal{F} \subset \mathbb{R}^N$ satisfying (43). As long as the problem formulation depends only on the inner products between data points in \mathcal{F} and not on the data points themselves, without explicit representation of the feature mapping Φ or the feature space \mathcal{F} , we can work on the feature space \mathcal{F} through the relation (43). As positive definite kernel functions satisfying

Mercer's condition, polynomial kernel and Gaussian kernel

$$\kappa(x, y) = (\gamma_1(x \cdot y) + \gamma_2)^d, d > 0 \text{ and } \gamma_1, \gamma_2 \in \mathbb{R},$$

$$\kappa(x, y) = \exp(-\|x - y\|^2/2\sigma^2), \sigma \in \mathbb{R}$$

are in wide use.

In this section, we present the formulation of a generalized eigenvalue problem in the kernel-based feature space and apply the generalized LDA algorithms, obtaining nonlinear discriminant analysis. Given a kernel function κ , let \mathcal{S}_b and \mathcal{S}_w be the between-class and within-class scatter matrices in the feature space $\mathcal{F} \subset \mathbb{R}^N$ which has been transformed by a mapping Φ satisfying (43). Then the LDA in \mathcal{F} finds a linear transformation $\mathcal{G} = [\varphi_1, \dots, \varphi_l] \in \mathbb{R}^{N \times l}$, where the columns of \mathcal{G} are the generalized eigenvectors corresponding to the l largest eigenvalues of

$$\mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi. \quad (44)$$

As in (7), \mathcal{S}_b and \mathcal{S}_w can be expressed as

$$\mathcal{S}_b = \mathcal{H}_b \mathcal{H}_b^T \quad \text{and} \quad \mathcal{S}_w = \mathcal{H}_w \mathcal{H}_w^T \quad \text{where}$$

$$\mathcal{H}_b = [\sqrt{n_1}(\tilde{c}_1 - \tilde{c}), \dots, \sqrt{n_r}(\tilde{c}_r - \tilde{c})] \in \mathbb{R}^{N \times r}, \quad (45)$$

$$\mathcal{H}_w = [\Phi(A_1) - \tilde{c}_1 e_1, \dots, \Phi(A_r) - \tilde{c}_r e_r] \in \mathbb{R}^{N \times n}, \quad (46)$$

$$\tilde{c}_i = \frac{1}{n_i} \sum_{j \in N_i} \Phi(a_j), \quad \tilde{c} = \frac{1}{n} \sum_{i=1}^n \Phi(a_i) \quad \text{and} \quad e_i = [1, \dots, 1] \in \mathbb{R}^{1 \times n_i}.$$

The notation $\Phi(A_i)$ is used to denote $\Phi(A_i) = \Phi([a_j, \dots, a_k]) = [\Phi(a_j), \dots, \Phi(a_k)]$.

Let φ be represented as a linear combination of $\Phi(a_i)$'s such as $\varphi = \sum_{i=1}^n u_i \Phi(a_i)$, and define

$$u = [u_1, \dots, u_n]^T, \quad \mathcal{K}_b = [b_{ij}]_{(1 \leq i \leq n, 1 \leq j \leq r)}, \quad b_{ij} = \sqrt{n_j} \left(\frac{1}{n_j} \sum_{p \in N_j} \kappa(a_i, a_p) - \frac{1}{n} \sum_{p=1}^n \kappa(a_i, a_p) \right). \quad (47)$$

Then we have

$$\mathcal{H}_b^T \varphi = \mathcal{K}_b^T u, \quad (48)$$

since

$$\begin{aligned} \mathcal{H}_b^T \varphi &= \begin{bmatrix} \sqrt{n_1}(\tilde{c}_1 - \tilde{c})^T \\ \vdots \\ \sqrt{n_r}(\tilde{c}_r - \tilde{c})^T \end{bmatrix} \left(\sum_{i=1}^n u_i \Phi(a_i) \right) \\ &= \begin{bmatrix} \sqrt{n_1}(\frac{1}{n_1} \sum_{p \in N_1} \Phi(a_p) - \frac{1}{n} \sum_{p=1}^n \Phi(a_p))^T \\ \vdots \\ \sqrt{n_r}(\frac{1}{n_r} \sum_{p \in N_r} \Phi(a_p) - \frac{1}{n} \sum_{p=1}^n \Phi(a_p))^T \end{bmatrix} \begin{bmatrix} \Phi(a_1), & \dots, & \Phi(a_n) \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} \\ &= \mathcal{K}_b^T u. \end{aligned}$$

Similarly, we can obtain

$$\mathcal{H}_w^T \varphi = \mathcal{K}_w^T u \quad \text{where} \quad (49)$$

$$\mathcal{K}_w = [w_{ij}]_{(1 \leq i \leq n, 1 \leq j \leq n)}, \quad (50)$$

$$w_{ij} = \kappa(a_i, a_j) - \frac{1}{n_\delta} \sum_{p \in N_\delta} \kappa(a_i, a_p) \text{ when } a_j \text{ belongs to the class } \delta.$$

From (48) and (49), for any $\varphi = \sum_{i=1}^n u_i \Phi(a_i)$ and $\psi = \sum_{i=1}^n v_i \Phi(a_i)$ we have

$$\mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi \quad \Leftrightarrow \quad \psi^T \mathcal{H}_b \mathcal{H}_b^T \varphi = \lambda \psi^T \mathcal{H}_w \mathcal{H}_w^T \varphi \quad (51)$$

$$\Leftrightarrow \quad v^T \mathcal{K}_b \mathcal{K}_b^T u = \lambda v^T \mathcal{K}_w \mathcal{K}_w^T u$$

$$\text{for } u = [u_1, \dots, u_n]^T, \quad v = [v_1, \dots, v_n]^T$$

$$\Leftrightarrow \quad \mathcal{K}_b \mathcal{K}_b^T u = \lambda \mathcal{K}_w \mathcal{K}_w^T u.$$

Therefore, the generalized eigenvalue problem $\mathcal{S}_b \varphi = \lambda \mathcal{S}_w \varphi$ becomes

$$\mathcal{K}_b \mathcal{K}_b^T u = \lambda \mathcal{K}_w \mathcal{K}_w^T u. \quad (52)$$

Note that $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ can be viewed as the between-class scatter matrix and within-class scatter matrix of the kernel matrix

$$K = [\kappa(a_i, a_j)]_{(1 \leq i \leq n, 1 \leq j \leq n)} \quad (53)$$

Algorithm 2 Nonlinear Discriminant Analysis

Given a data matrix $A = [a_1, \dots, a_n] \in \mathbb{R}^{m \times n}$ with r classes and a kernel function κ , it computes the l dimensional representation of any input vector $z \in \mathbb{R}^{m \times 1}$ by applying the generalized LDA algorithm in the kernel-based feature space composed of the columns of $K = [\kappa(a_i, a_j)]_{(1 \leq i \leq n, 1 \leq j \leq n)}$.

1. Compute $\mathcal{K}_b \in \mathbb{R}^{n \times r}$, $\mathcal{K}_w \in \mathbb{R}^{n \times n}$ and $\mathcal{K}_t \in \mathbb{R}^{n \times n}$ according to Eqs. (47), (50) and (54).
 2. Compute transformation matrix \mathcal{G} by applying the generalized LDA algorithms discussed in Section 2.
 3. For any input vector $z \in \mathbb{R}^{m \times 1}$, a dimension reduced representation is computed by Eq. (56).
-

when each column $[\kappa(a_1, a_j), \dots, \kappa(a_n, a_j)]^T$ in K is considered as a data point in the n -dimensional space. It can be observed by comparing the structures of \mathcal{K}_b and \mathcal{K}_w with those of \mathcal{H}_b and \mathcal{H}_w in (45-46). As in \mathcal{K}_b and \mathcal{K}_w of (47) and (50), \mathcal{K}_t can be computed as

$$\mathcal{K}_t = [t_{ij}]_{(1 \leq i \leq n, 1 \leq j \leq n)}, \quad t_{ij} = \kappa(a_i, a_j) - \frac{1}{n} \sum_{p=1}^n \kappa(a_i, a_p). \quad (54)$$

Since $\mathcal{K}_b \mathcal{K}_b^T$ and $\mathcal{K}_w \mathcal{K}_w^T$ are both singular in the feature space, the classical LDA can not be applied for the generalized eigenvalue problem (52). Now we apply the generalized LDA algorithms discussed in Section 2 to solve (52), obtaining nonlinear discriminant analysis. Let

$$\mathcal{G} = [u^{(1)}, \dots, u^{(l)}] \in \mathbb{R}^{n \times l} \quad (55)$$

be the transformation matrix obtained by applying any generalized LDA algorithm in the feature space. Then the dimension reduced representation of any data item $z \in \mathbb{R}^{m \times 1}$ is given by

$$\mathcal{G}^T \begin{bmatrix} \kappa(a_1, z) \\ \vdots \\ \kappa(a_n, z) \end{bmatrix} \in \mathbb{R}^{l \times 1}. \quad (56)$$

Algorithm 2 summarizes nonlinear extension of generalized LDA algorithms by kernel methods.

Data	dim.	no.data	classes	LDA	In the feature space				
					RLDA	LDA/GSVD	$To-N(S_w)$	$To-R(S_b)$	$To-NR(S_w)$
Musk	166	6599	2	91.2	97.6	99.4	99.4	89.2	99.3
Isolet	617	7797	26	93.9	95.8	96.8	97.0	89.7	97.1
Car	6	1728	4	88.2	94.7	94.1	94.9	84.5	95.2
Mfeature	649	2000	10	–	94.4	98.1	98.3	94.0	98.3
Bcancer	9	699	2	95.3	95.2	96.4	93.5	92.8	94.3
Bscale	4	625	3	87.0	94.1	86.5	86.5	86.5	86.1

Table 5: Prediction accuracies(%) by the classical LDA in the original space and the generalized LDA algorithms in the nonlinearly transformed feature space. In the Mfeature dataset, the classical LDA was not applicable due to the singularity of the within-class scatter matrix.

3.1 Experimental Comparisons of Nonlinear Discriminant Analysis Algorithms

For this experiment, six data sets from UCI Machine Learning Repository were used. By randomly splitting the data to the training and test set of equal size and repeating it 10 times, ten pairs of training and test sets were constructed for each data. For the Bcancer and Bscale data sets, the ratio of training and test set was set as 4:1. Using the training set of the first pair among ten pairs and the nearest-neighbor classifier, 5 cross-validation was used in order to determine the optimal value for σ in the Gaussian kernel function $\kappa(x, y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right)$. After finding the optimal σ values, mean prediction accuracies from ten pairs of training and test sets were calculated and they are reported in Table 5. In the regularization method, while the regularization parameter was set as 1, the optimal σ value was searched by the cross-validation. Table 5 also reports the prediction accuracies by the classical LDA in the original data space and it demonstrates that nonlinear discriminant analysis can improve prediction accuracies compared with linear discriminant analysis.

Figure 3 illustrates the computational complexities using the specific sizes of the training data used in Table 5. As in the comparison of the generalized LDA algorithms, the method $To-R(S_b)$ [5] gives the lowest computational complexities among the compared

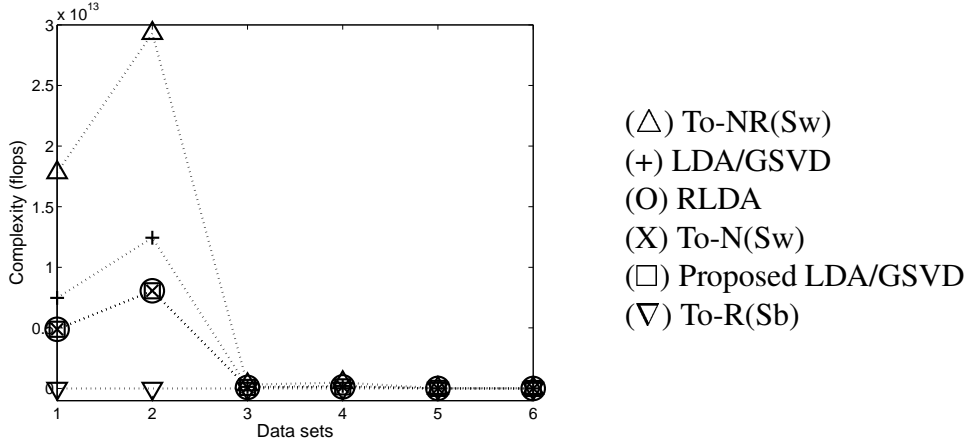


Figure 3: The figures compare complexities required for the generalized LDA algorithms in the feature space for specific problem sizes of training data used in Table 5. From the left on x-axis, the data sets, Musk, Isolet, Car, Mfeature, Bcancer and Bscale are corresponded.

methods. However, combining $To-R(S_b)$ with kernel methods does not make effective nonlinear dimension reduction method as shown in Table 5. In the generalized eigenvalue problem,

$$\mathcal{K}_b \mathcal{K}_b^T u = \lambda \mathcal{K}_w \mathcal{K}_w^T u \quad \text{where } \mathcal{K}_b \mathcal{K}_b^T, \mathcal{K}_w \mathcal{K}_w^T \in \mathbb{R}^{n \times n},$$

the data dimension is equal to the number of data and the rank of $\mathcal{K}_w \mathcal{K}_w^T$ is not severely smaller than the data dimension. However, poor performances by $To-R(S_b)$ demonstrate that the null space of $\mathcal{K}_w \mathcal{K}_w^T$ contains discriminative information. Figures 2 and 3 show that the proposed LDA/GSVD method can reduce greatly the computational cost of the original LDA/GSVD in both the original space and the feature space.

4 Conclusions/Discussions

We presented the relationships among several generalized Linear Discriminant Analysis algorithms developed for handling undersampled problems and compared their computational complexities and performances. As discussed in the theoretical comparison, many algorithms are closely related, and experimental results indicate that computational com-

plexities are important issues in addition to classification performances. The LDA/GSVD showed competitive performances throughout the experiments, but the computational complexities can be expensive especially for high dimensional data. An efficient algorithm has been proposed, which produces the same solution as LDA/GSVD. The computational savings are remarkable especially for high dimensional data.

Nonlinear extensions of the generalized LDA algorithms by the formulation of generalized eigenvalue problem in the kernel-based feature space were presented. Experimental results using data sets from UCI database demonstrate that nonlinear discriminant analysis can improve prediction accuracies compared with linear discriminant analysis.

References

- [1] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.
- [2] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-interscience, New York, 2001.
- [3] J.H. Friedman. Regularized discriminant analysis. *Journal of the American statistical association*, 84(405):165–175, 1989.
- [4] P. Howland, M. Jeon, and H. Park. Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, 25(1):165–179, 2003.
- [5] L. Chen, H.M. Liao, M. Ko, J. Lin, and G. Yu. A new LDA-based face recognition system which can solve the small sample size problem. *pattern recognition*, 33:1713–1726, 2000.

- [6] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data- with application to face recognition. *pattern recognition*, 34:2067–2070, 2001.
- [7] J. Yang and J.-Y. Yang. Why can LDA be performed in PCA transformed space? *Pattern Recognition*, 36:563–566, 2003.
- [8] W. Zheng, L. Zhao, and C. Zou. An efficient algorithm to solve the small sample size problem for lda. *Pattern Recognition*, 37:1077–1079, 2004.
- [9] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge, 2000.
- [10] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE transactions on pattern analysis and machine intelligence*, 26(8):995–1006, 2004.
- [11] C.C. Paige and M.A. Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18:398–405, 1981.
- [12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.
- [13] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces v.s. fisherfaces: Recognition using class specific linear projection. *IEEE transactions on pattern analysis and machine learning*, 19(7):711–720, 1997.
- [14] W. Zheng, C. Zou, and L. Zhao. *Real-time Face Recognition Using Gram-Schmidt Orthogonalization for LDA*. In the Proceedings of the 17th International Conference on Pattern Recognition, 2004.
- [15] Z. Jin, J.-Y. Yang, Z.-S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34:1405–1416, 2001.

- [16] J. Ye, R. Janardan, Q. Li and H. Park. *Feature extraction via generalized uncorrelated linear discriminant analysis*. In the proceedings of the 21st international conference on machine learning, 2004.
- [17] T.G. Kolda and D.P. O’Leary. A semidiscrete matrix decomposition for latent semantic indexing in information retrieval. *ACM transactions on Information Systems*, 16(4):322–346, 1998.
- [18] <http://www.uk.research.att.com/facedatabase.html>.
- [19] <http://cvc.yale.edu/projects/yalefaces/yalefaces.html>.
- [20] B. Scholkopf, A.J. Smola, and K.-R. Muller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10:1299–1319, 1998.
- [21] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller. Fisher discriminant analysis with kernels. In E.Wilson J.Larsen and S.Douglas, editors, *Neural networks for signal processing IX*, pages 41–48. IEEE, 1999.
- [22] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural computation*, 12:2385–2404, 2000.
- [23] V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. *Advances in neural information processing systems*, 12:568–574, 2000.
- [24] S.A. Billings and K.L. Lee. Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural networks*, 15(2):263–270, 2002.
- [25] C.H. Park and H. Park. Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition. *SIAM journal on matrix analysis and applications*, 27(1):87–102, 2005.

- [26] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, and A.J. Smola. Input space versus feature space in kernel-based methods. *IEEE transactions on neural networks*, 10(5):1000–1017, September 1999.
- [27] C.J.C. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167, 1998.