Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-048

A Comparison of Generalized LDA Algorithms for Undersampled Problems

Cheonghee Park and Haesun Park

December 11, 2003

A Comparison of Generalized LDA Algorithms for Undersampled Problems

Cheong Hee Park and Haesun Park*

Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455
{chpark, hpark}@cs.umn.edu

Abstract

Linear Discriminant Analysis (LDA) is a dimension reduction method which finds an optimal linear transformation that maximizes the between-class scatter and minimizes the within-class scatter. In undersampled problems where the number of 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 for undersampled problems, several generalizations of the LDA have been proposed recently. In this paper, we present the theoretical and algorithmic relationships among several generalized LDA algorithms and compare their computational complexities and performances in text classification and face recognition. Towards a practical dimension reduction method for high dimensional data, an efficient algorithm is also proposed.

Keywords: Dimension reduction, Facial recognition, Feature extraction, Generalized Linear Discriminant Analysis, Generalized Singular Value Decomposition, Linear Discriminant Analysis, Text classification, Undersampled problem.

*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).

1 Introduction

Linear Discriminant Analysis (LDA) and Principal Component Analysis (PCA) have been two of the most commonly used linear dimension reduction methods. Given high dimensional data, linear dimension reduction methods seek an optimal linear transformation by which the original data is transformed to a much lower dimensional space. PCA aims to minimize information loss in representing the original data in a lower dimensional space. Hence, in PCA, optimal dimension reduction is performed by searching for the directions along which variance in the data is greatest. While the class structure in the data is not considered in finding a transformation in PCA, the goal of LDA is an optimal dimension reduction that preserves class structure in the data [1, 2]. Hence the criteria for dimension reduction in LDA are formulated to optimize class separability.

The separability among classes is measured by scatter matrices. The scatters between classes, within each class, and among all data items are represented as the between-class scatter matrix (S_b), within-class scatter matrix (S_w) and mixture scatter matrix (S_m), respectively. In the classical LDA, the optimal transformation is obtained based on the eigenvectors corresponding to the largest eigenvalues of $S_2^{-1}S_1$ where S_1 is S_b and S_2 is S_w or S_m [2]. However, for undersampled problems 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 problem caused by the singularity of the scatter matrices in undersampled problems, several methods have been proposed [3, 4, 5, 6]. Regularized LDA has been widely used, where a positive diagonal identity matrix is added to S_w to make it nonsingular when S_w is singular or ill-conditioned [3, 7]. Recently, a generalization of LDA based on the generalized singular value decomposition, called LDA/GSVD, has been developed to make the LDA applicable when any scatter matrix is singular [4]. In [5, 6], methods that combine two linear transformations have been introduced, where the attention is restricted to either S_b or S_w in the first stage. Chen et al. [5] proposed a method which projects the original space to the null space of S_w, and then the between-class scatter in the projected space is maximized. In [6], the original space is transformed by using a basis of range(S_b), and then in the transformed space minimization of within-class

scatter is pursued.

In this paper, we compare these generalized LDA algorithms and present theoretical and algorithmic relationships among them. The computational complexities and performances are compared in text classification and face recognition. We also propose an efficient algorithm for the generalized LDA which produces the same solution as the LDA/GSVD presented in [4], while saving computational complexity greatly.

The rest of this paper is organized as follows. In Section 2, the classical LDA is presented. Several generalizations of LDA and the theoretical comparison among them are given in Section 3. A new efficient algorithm for LDA/GSVD is proposed and analyzed in Section 4, and the computational complexities of these methods are compared in Section 5. Extensive experimental results for undersampled problems in text classification and face recognition are given in Section 6.

2 Linear Discriminant Analysis

Let

$$A = [a_1, \cdots, a_n] = [A_1, A_2, \cdots, A_r] \in \mathbb{R}^{m \times n}$$

be a vector space representation of a data set with r classes, where each data item is represented as a column vector a_i in an m-dimensional space. Each class A_i consists of n_i columns and $\sum_{i=1}^{r} n_i = n$. Let N_i $(1 \leq i \leq r)$ denote the index set for column vectors which belong to the i-th class. Utilizing the class centroids $c_i = \frac{1}{n_i} \sum_{j \in N_i} a_j$ and the global centroid $c = \frac{1}{n} \sum_{j=1}^{n} a_j$, the between-class scatter matrix S_b, within-class scatter matrix S_w, and mixture scatter matrix S_m are defined as

$$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^T = H_b H_b^T \quad \in \mathbb{R}^{m \times m}, \tag{1}$$

$$S_w = \sum_{i=1}^{r} \sum_{j \in N_i} (a_j - c_i)(a_j - c_i)^T = H_w H_w^T \quad \in \mathbb{R}^{m \times m}, \tag{2}$$

$$S_m = \sum_{j=1}^{n} (a_j - c)(a_j - c)^T = H_m H_m^T \quad \in \mathbb{R}^{m \times m}, \tag{3}$$

where

$$H_b = [\sqrt{n_1}(c_1 - c), \cdots, \sqrt{n_r}(c_r - c)] \quad \in \mathbb{R}^{m \times r}, \tag{4}$$

$$H_w = [A_1 - c_1 e_1, \cdots, A_r - c_r e_r] \quad \in \mathbb{R}^{m \times n}, \tag{5}$$

$$H_m = [a_1 - c, \cdots, a_n - c] \quad \in \mathbb{R}^{m \times n}, \tag{6}$$

$$e_i = [1, \cdots, 1] \in \mathbb{R}^{1 \times n_i}.$$

The quality of clustered structure in a data set can be measured using these scatter matrices. By a linear transformation $G^T \in \mathbb{R}^{l \times m}$, the data $A \in \mathbb{R}^{m \times n}$ is transformed to $G^T A \in \mathbb{R}^{l \times n}$ and scatter matrices of $G^T A$ can be computed as

$$\tilde{S}_i = G^T S_i G \quad \text{for} \quad i = b, w \text{ and } m.$$

The optimal dimension reducing transformation for LDA is the one that maximizes the scatter between classes and minimizes the scatter within each class in a reduced dimensional space. The common optimization criteria in LDA are to find a linear transformation G^T which maximizes

$$J_1(G) = \text{trace}((G^T S_2 G)^{-1}(G^T S_1 G)) \quad \text{or} \quad J_2(G) = \frac{\det(G^T S_1 G)}{\det(G^T S_2 G)}, \tag{7}$$

where $(S_1, S_2) = (S_b, S_w)$ or $(S_1, S_2) = (S_b, S_m)$.

Throughout this section, we use the pair $(S_1, S_2) = (S_b, S_w)$ for the review of the classical LDA. It is well known [1] that both criteria in (7) are maximized when the columns of $G \in \mathbb{R}^{m \times (r-1)}$ are the eigenvectors g's corresponding to the $r - 1$ largest eigenvalue λ's of

$$S_b g = \lambda S_w g. \tag{8}$$

In the classical LDA, S_w is assumed to be nonsingular. With this assumption, the Cholesky decomposition of S_w,

$$S_w = LL^T$$

where $L \in \mathbb{R}^{m \times m}$ is a nonsingular lower triangular matrix, can be used to solve Eq. (8), since

$$S_b g = \lambda S_w g \quad \Leftrightarrow \quad L^{-1} S_b g = \lambda L^T g \quad \Leftrightarrow \quad (L^{-1} S_b L^{-T}) L^T g = \lambda L^T g. \tag{9}$$

An eigenvector x of the symmetric matrix $L^{-1}S_bL^{-T}$ gives the generalized eigenvector

$$g = L^{-T}x$$

of (8). Note that $L^{-1}S_bL^{-T}$ and $L^{-T}x$ can be computed by solving linear systems with a triangular coefficient matrix without directly computing the inverse of L.

Another approach to solve (8) is by two-step diagonalizations. First, the symmetric eigenvalue decomposition (EVD) of S_w is computed as

$$S_w = U\Sigma_w U^T, \tag{10}$$

where $U^TU = I$ and Σ_w is a diagonal matrix with positive non-increasing diagonal elements. Denoting $Z = U\Sigma_w^{-1/2}$, we have

$$Z^TS_wZ = I. \tag{11}$$

Next, the symmetric EVD of Z^TS_bZ is computed as

$$Z^TS_bZ = W\Sigma_bW^T, \tag{12}$$

where $W^TW = I$ and Σ_b is a diagonal matrix with non-increasing diagonals. From (11) and (12), a simultaneous diagonalization of S_b and S_w is obtained as

$$W^TZ^TS_bZW = \Sigma_b \quad \text{and} \quad W^TZ^TS_wZW = I, \tag{13}$$

and from (13), we have

$$S_b(ZW) = S_w(ZW)\Sigma_b.$$

Hence

$$S_bg_i = \lambda_iS_wg_i \quad \text{where} \quad ZW = [g_1, \cdots, g_m] \quad \text{and} \quad \Sigma_b = \text{diag}(\lambda_1, \cdots, \lambda_m).$$

Since rank$(S_b) \leq r-1$, the columns of the transformation matrix G for LDA are obtained from the first $r-1$ columns of ZW.

When the number of data items is smaller than the dimension of data space (i.e., $n < m$) as in document retrieval [8] and face recognition [9], all of the scatter matrices become singular. Hence the above solution techniques for the classical LDA fail. For these undersampled problems, several methods for the generalization of LDA have been proposed [5, 6], including regularized LDA [3] and the method based on the generalized singular value decomposition, LDA/GSVD [4]. In the next section we review several generalization methods of LDA and present algebraic and theoretical relationships among them.

3 Generalizations of Linear Discriminant Analysis

To make matrix notations consistent among the methods compared in this section, we use subscripts, whenever distinction is needed based on the notations used in various sections.

3.1 Regularized LDA

In the regularized LDA, 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,

$$\tilde{S}_w = S_w + \alpha I$$

is nonsingular with any $\alpha > 0$. Therefore we can apply the algorithms for the classical LDA to solve the eigenvalue problem

$$S_b g = \lambda \tilde{S}_w g. \tag{14}$$

Explicit computation of $\tilde{S}_w^{-1}$ for solving (14) should be avoided especially when dealing with high dimensional data. Instead, the diagonalization of $\tilde{S}_w$ can be obtained from the EVD of $S_w \in \mathbb{R}^{m\times m}$ which can be computed by using the smaller matrix $H_w \in \mathbb{R}^{m\times n}$ as follows. Let the singular value decomposition (SVD) of H_w be

$$H_w = UDY^T, \tag{15}$$

where $U \in \mathbb{R}^{m \times m}$ and $Y \in \mathbb{R}^{n \times n}$ are orthogonal and $D \in \mathbb{R}^{m \times n}$ is a diagonal matrix. Then the EVD of S_w is obtained as

$$S_w = H_w H_w^T = U \Sigma_w U^T \quad \text{where} \quad \Sigma_w = DD^T \tag{16}$$

and the EVD of $\tilde{S}_w$ as

$$\tilde{S}_w = U \tilde{\Sigma}_w U^T \tag{17}$$

where $\tilde{\Sigma}_w = \Sigma_w + \alpha I$. Now, from the EVD of $\tilde{\Sigma}_w^{-1/2} U^T \, S_b \, U \, \tilde{\Sigma}_w^{-1/2}$,

$$\tilde{\Sigma}_w^{-1/2} U^T \, S_b \, U \, \tilde{\Sigma}_w^{-1/2} = W_f \Sigma_{fb} W_f^T,$$

the transformation matrix G_f is obtained as $r-1$ columns of $U \tilde{\Sigma}_w^{-1/2} W_f$.

Two – Class Problem

We now consider the simple case when the data set has two classes, since in that case the effect of the regularization parameter α to the solution is easy to illustrate. Two-class problem in LDA is known as Fisher Discriminant Analysis (FDA) [1]. In two-class case,

$$S_b = \frac{n_1 n_2}{n} (c_1 - c_2)(c_1 - c_2)^T, \tag{18}$$

and the eigenvalue problem (8) is simplified to

$$S_w^{-1} (c_1 - c_2)(c_1 - c_2)^T g = \lambda g \tag{19}$$

when S_w is nonsingular. The solution for (19) is a nonzero multiple of

$$g = S_w^{-1} (c_1 - c_2),$$

and the 1-dimensional representation of any data item $a \in \mathbb{R}^{m \times 1}$ by LDA is obtained as

$$g^T a = (c_1 - c_2)^T S_w^{-1} a = (c_1 - c_2)^T U \Sigma_w^{-1} U^T a.$$

Similarly, the regularized LDA by αI gives the solution

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

where 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.

3.2 LDA based on Generalized Singular Value Decomposition (LDA/GSVD)

Howland et al. [4] applied the Generalized Singular Value Decomposition (GSVD) due to Paige and Saunders [10] to overcome the limitation of the classical LDA and made the LDA applicable even when any scatter matrix is singular. For our later discussion, we introduce the GSVD briefly. The proof is given in Appendix.

THEOREM 1 (GENERALIZED SINGULAR VALUE DECOMPOSITION [10]) *Suppose two matrices with the same number of columns, $K_b \in \mathbb{R}^{p_1 \times q}$ and $K_w \in \mathbb{R}^{p_2 \times q}$, are given. Then for*

$$K = \begin{bmatrix} K_b \\ K_w \end{bmatrix} \quad \textit{and} \quad t = \textit{rank}(K),$$

there exist orthogonal matrices $U_{hb} \in \mathbb{R}^{p_1 \times p_1}$ and $U_{hw} \in \mathbb{R}^{p_2 \times p_2}$, and a nonsingular matrix $X \in \mathbb{R}^{q \times q}$ such that

$$U_{hb}^T K_b X = [\Sigma_{hb} \quad 0] \quad \textit{and} \quad U_{hw}^T K_w X = [\Sigma_{hw} \quad 0], \tag{20}$$

where

$$\Sigma_{hb} = \underbrace{\begin{bmatrix} I_{hb} & & \\ & D_{hb} & \\ & & 0_{hb} \end{bmatrix}}_{\mu \quad \tau \quad t-\mu-\tau} \begin{matrix} \}\mu \\ \}\tau \\ \}p_1 - \mu - \tau \end{matrix} \quad \textit{and} \quad \Sigma_{hw} = \underbrace{\begin{bmatrix} 0_{hw} & & \\ & D_{hw} & \\ & & I_{hw} \end{bmatrix}}_{\mu \quad \tau \quad t-\mu-\tau} \begin{matrix} \}p_2 - t + \mu \\ \}\tau \\ \}t - \mu - \tau \end{matrix} \tag{21}$$

Here μ and τ are associated with

$$\mu = t - \textit{rank}(K_w) \quad \textit{and} \quad \tau = \textit{rank}(K_b) + \textit{rank}(K_w) - t \tag{22}$$

Algorithm 1 LDA/GSVD [4]

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with r classes, it computes the dimension reducing transformation matrix $G_h \in \mathbb{R}^{m\times(r-1)}$. For any input vector $a \in \mathbb{R}^{m\times 1}$, its $r-1$ dimensional representation is given by $G_h^T a$.

1. Compute the SVD or the complete orthogonal decomposition of $K = P \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} Q^T \in \mathbb{R}^{(r+n)\times m}$, which is $K = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Q^T$ and $t = \text{rank}(K)$.
2. Compute W_h from the SVD of $P(1:r, 1:t)$, which is $P(1:r,1:t) = U_{hb}\Sigma_{hb}W_h^T$.
3. Compute the first $r-1$ columns of $X = Q \begin{bmatrix} R^{-1}W_h & 0 \\ 0 & I \end{bmatrix}$, and assign them to G_h.

and

$$D_{hb} = diag(\eta_{\mu+1}, \ldots, \eta_{\mu+\tau}) \quad \textit{for} \quad 1 > \eta_{\mu+1} \geq \cdots \geq \eta_{\mu+\tau} > 0,$$
$$D_{hw} = diag(\zeta_{\mu+1}, \ldots, \zeta_{\mu+\tau}) \quad \textit{for} \quad 0 < \zeta_{\mu+1} \leq \cdots \leq \zeta_{\mu+\tau} < 1,$$
$$\eta_i^2 + \zeta_i^2 = 1 \quad \textit{for } i = \mu+1, \ldots, \mu+\tau. \qquad \square$$

Suppose GSVD is applied to the matrix pair $(H_b^T,\ H_w^T)$, where H_b and H_w are defined in (4) and (5) respectively, and we obtain

$$U_{hb}^T H_b^T X = [\Sigma_{hb} \quad 0] \quad \text{and} \quad U_{hw}^T H_w^T X = [\Sigma_{hw} \quad 0] \tag{23}$$

by Theorem 1. Then from (23),

$$X^T S_b X = X^T H_b H_b^T X = \begin{bmatrix} \Sigma_{hb}^T\Sigma_{hb} & & \\ & 0_{m-t} & \end{bmatrix} = \begin{bmatrix} I_\mu & & & \\ & D_{hb}^T D_{hb} & & \\ & & 0_{t-\mu-\tau} & \\ & & & 0_{m-t} \end{bmatrix} \text{ and} \tag{24}$$

$$X^T S_w X = X^T H_w H_w^T X = \begin{bmatrix} \Sigma_{hw}^T\Sigma_{hw} & & \\ & 0_{m-t} & \end{bmatrix} = \begin{bmatrix} 0_\mu & & & \\ & D_{hw}^T D_{hw} & & \\ & & I_{t-\mu-\tau} & \\ & & & 0_{m-t} \end{bmatrix}, \tag{25}$$

where the subscripts in I and 0 denote the order of square identity and zero matrices.

Denoting the diagonal elements in (24) as η_i^2's and the diagonal elements in (25) as ζ_i^2's, we have

$$\zeta_i^2 S_b x_i = \eta_i^2 S_w x_i \quad i = 1, \cdots, m, \tag{26}$$

where x_i is the column vectors of X. Since $\Sigma_{hb}^T \Sigma_{hb}$ has nonincreasing diagonal elements and $\Sigma_{hw}^T \Sigma_{hw}$ has nondecreasing diagonal elements, $r-1$ leftmost columns of X gives an optimal transformation G_h^T for LDA. This method is called LDA/GSVD and the algorithm is summarized in Algorithm 1.

From (24 - 26), the generalized eigenvalues and eigenvectors obtained by GSVD are classified as shown in Table 1. The last $m-t$ eigenvectors in Table 1 belong to null$(S_w) \cap$ null(S_b). For any $x \in$ null$(S_w) \cap$ 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

$$\begin{cases} x^T c_i = x^T c \quad \text{for} \quad i = 1, \cdots, r \\ x^T a_j = x^T c_i \quad \text{for all} \;\; a_j \text{ in a class } i \end{cases} \tag{27}$$

and these imply that all data items are transformed to one point by x^T. Therefore, the vectors in null$(S_b) \cap$ null(S_w) do not convey any discriminative information among the classes, even though the corresponding eigenvalues are not necessarily zeros.

Both regularization method and LDA/GSVD achieve the solution of LDA through the simultaneous diagonalization of scatter matrices, while the within-class scatter matrix handled in regularized LDA is modified to be 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 range(S_b) or 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.

	η_i	ζ_i	$\lambda_i = \frac{\eta_i^2}{\zeta_i^2}$	x_i belongs to
$1 \leq i \leq \mu$	1	0	∞	$\text{null}(S_w) \cap \text{null}(S_b)^c$
$\mu + 1 \leq i \leq \mu + \tau$	$1 > \eta_i > 0$	$0 < \zeta_i < 1$	$\infty > \lambda_i > 0$	$\text{null}(S_w)^c \cap \text{null}(S_b)^c$
$\mu + \tau + 1 \leq i \leq t$	0	1	0	$\text{null}(S_w)^c \cap \text{null}(S_b)$
$t + 1 \leq i \leq m$	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.

3.3 A Method based on Projection onto 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 null(S_w), and then in the projected space, a transformation that maximizes the between-class scatter is computed.

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

$$H_w = UDY^T.$$

Partitioning U as

$$U = [\underbrace{U_{e1}}_{s_1} \underbrace{U_{e2}}_{m-s_1}]$$

where $s_1 = \text{rank}(H_w) = \text{rank}(S_w)$. Then

$$\text{null}(S_w) = \text{span}(U_{e2}). \tag{28}$$

First, the transformation by $U_{e2}U_{e2}^T$ projects the original data to 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 $\tilde{S}_b = U_{e2}U_{e2}^T S_b U_{e2}U_{e2}^T$ be

$$\tilde{S}_b = W_e \Sigma_{eb} W_e^T = [\underbrace{W_{e1}}_{s_2} \underbrace{W_{e2}}_{m-s_2}] \begin{bmatrix} D_{eb} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} W_{e1}^T \\ W_{e2}^T \end{bmatrix}, \tag{29}$$

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

$$G_e = U_{e2} U_{e2}^T W_{e1}. \tag{30}$$

Two – Class Problem

In two-class problem, S_b is given as in (18) and

$$\tilde{S}_b = U_{e2} U_{e2}^T \rho (c_1 - c_2)(c_1 - c_2)^T U_{e2} U_{e2}^T = \left(\frac{w}{\|w\|_2} \right) \rho \|w\|_2^2 \left(\frac{w}{\|w\|_2} \right)^T, \tag{31}$$

where $\rho = n_1 n_2 / n$ and $w = U_{e2} U_{e2}^T (c_1 - c_2) \in \mathbb{R}^{m \times 1}$. Hence the transformation matrix $g \in \mathbb{R}^{m \times 1}$ is obtained by

$$g = U_{e2} U_{e2}^T \frac{w}{\|w\|_2} = \nu U_{e2} U_{e2}^T (c_1 - c_2)$$

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

$$g^T a = \nu (c_1 - c_2)^T U_{e2} U_{e2}^T a.$$

Relationship with LDA/GSVD

From (29), we have

$$(W_{e1}^T U_{e2} U_{e2}^T) S_b (U_{e2} U_{e2}^T W_{e1}) = D_{eb} \quad \text{and} \quad (W_{e1}^T U_{e2} U_{e2}^T) S_w (U_{e2} U_{e2}^T W_{e1}) = 0. \tag{32}$$

The second equation holds due to (28). Eqs. in (32) imply that the column vectors g_i of G_e given in (30) belong to $\text{null}(S_w) \cap \text{null}(S_b)^c$ and they are discriminative vectors, since transformation by these vectors minimizes the within-class scatter to zero and increases the between-class scatter. Based on this observation, this method due to Chen et al. can be compared with LDA/GSVD. The top row of Table 1 shows that the LDA/GSVD solution also includes the vectors from $\text{null}(S_w) \cap \text{null}(S_b)^c$. By denoting X in step 3 of Algorithm 1 LDA/GSVD as

$$X = [\underbrace{X_1}_{\mu} \underbrace{X_2}_{\tau} \underbrace{X_3}_{t-\mu-\tau} \underbrace{X_4}_{m-t}], \tag{33}$$

we find a relationship between X_1 and $U_{e2}U_{e2}^T W_{e1}$.

Eq. (25) implies that $[X_1 \ X_4]$ is a basis of null(S_w). Hence any vector in null(S_w) can be represented as a linear combination of column vectors in $[X_1 \ X_4]$. Note that null$(S_w) \cap$ null$(S_b)^c$ is not a vector space and X_1 is not a basis of null$(S_w) \cap$ null$(S_b)^c$. The following Theorem shows the condition for any vector in null(S_w) to belong to null$(S_w) \cap$ null$(S_b)^c$.

THEOREM 2 *Any vector x belongs to null$(S_w) \cap$ null$(S_b)^c$ if and only if x is represented as $X_1 h + X_4 k$ where $h \neq 0 \in \mathbb{R}^{\mu \times 1}$ and $k \in \mathbb{R}^{(m-t)\times 1}$.*

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

Now let us prove that $x = X_1 h + X_4 k$ belongs to null$(S_w) \cap$ null$(S_b)^c$ if $h \neq 0$. Since $x = X_1 h + X_4 k \in$ null(S_w), it is enough to show $x \notin$ null(S_b). From (24),

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

By Theorem 2,

$$U_{e2}U_{e2}^T W_{e1} = X_1 H + X_4 K \tag{34}$$

for some matrices $H \in \mathbb{R}^{\mu \times s_2}$ and $K \in \mathbb{R}^{(m-t)\times s_2}$ with $s_2 = \text{rank}(\tilde{S}_b)$, where each column of H is nonzero. Hence for any data item $a \in \mathbb{R}^{m\times 1}$, the reduced dimensional representation by $G_e = U_{e2}U_{e2}^T W_{e1}$ is given as

$$G_e^T a = H^T X_1^T a + K^T X_4^T a. \tag{35}$$

As explained in (27) of Section 3.2, since all data items are transformed to one point by x^T for $x \in$ null$(S_w) \cap$ null(S_b), the second part $K^T X_4^T a$ in (35) corresponds to the translation which does not affect the classification performance.

While the transformation matrix $G_e = U_{e2}U_{e2}^T W_{e1}$ by the Chen et al.'s method is related to X_1 of LDA/GSVD as in (35), the main difference between the two methods is due to the eigenvectors

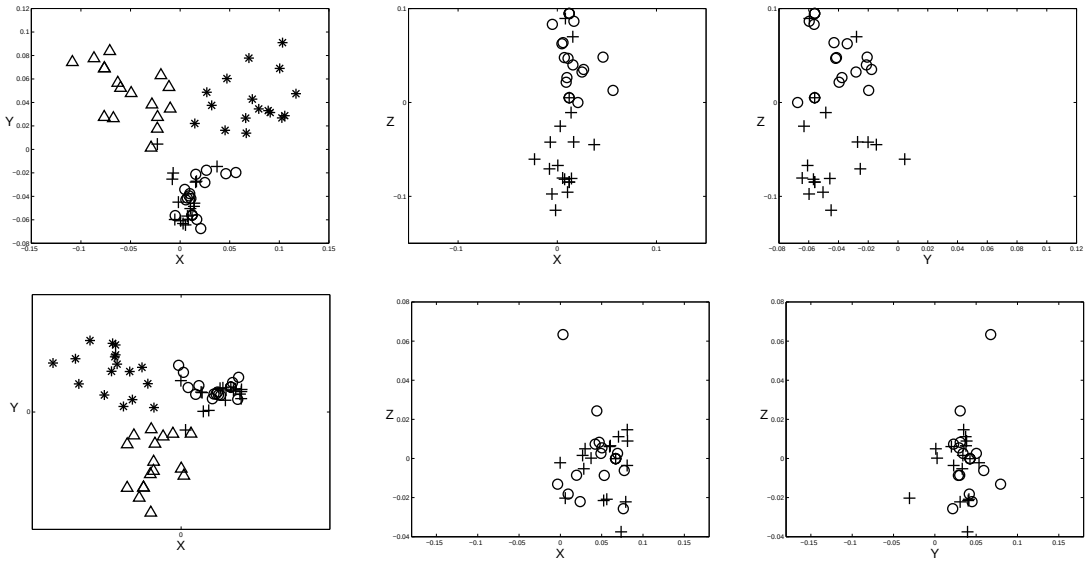


Figure 1: The visualization of data in the reduced dimensional spaces by LDA/GSVD (figures in the first row) and the method based on the projection to null(S_w) (figures in the second row).

in null$(S_w)^c \cap$null$(S_b)^c$, which correspond to the second row in Table 1. The projection to null(S_w) by $U_{e2}U_{e2}^T$ excludes vectors in null$(S_w)^c$, and therefore null$(S_w)^c \cap$ 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 $G_e = U_{e2}U_{e2}^T W_{e1}$ is less than $r - 1$, while LDA/GSVD includes vectors from both null$(S_w) \cap$ null$(S_b)^c$ and null$(S_w)^c \cap$ 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 [11]. 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 based on the projection to null(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 due to Chen et al. was 2. On the other hand, LDA/GSVD produced two eigenvectors from null$(S_w) \cap$ null$(S_b)^c$ and one eigenvector from null$(S_w)^c \cap$ 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 null$(S_w)^c \cap$ null$(S_b)^c$ improves the separation of classes. The bottom three figures were generated by the method based on the projection to null(S_w). Since rank$(\tilde{S}_b)$=2, the reduced dimension by that method was 2 and the first figure shows the reduced dimensional space. The second and third figures show that adding one more column vector from $U_{e2}U_{e2}^T W_{e2}$ 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 null$(S_w) \cap$ null(S_b).

As mentioned in [5], the method due to Chen et al. can not be applied when S_w is nonsingular since null$(S_w) = \{0\}$. Hence Chen et al.'s method does not give the classical LDA as a special case when S_w is nonsingular. However, LDA/GSVD gives the solution which is the same as that of the classical LDA when S_w is nonsingular.

3.4 A Method based on the Transformation by a Basis of 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 3.3, the method presented in this section first transforms the original space by using a basis of range(S_b), and then in the transformed space the minimization of within-class scatter is pursued.

Consider the EVD of S_b,

$$S_b = V\Sigma_{yb}V^T = [\underbrace{V_1}_{s}\underbrace{V_2}_{m-s}] \begin{bmatrix} D_{yb} & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} V_1^T \\ V_2^T \end{bmatrix},$$

where V is orthogonal, rank$(S_b) = s$ and D_{yb} is a diagonal matrix with $\lambda_1 \geq \cdots \geq \lambda_s > 0$. Then

$$\text{range}(S_b) = \text{span}(V_1).$$

In the method by Yu and Yang, the original data is first transformed to an s-dimensional space by

$V_y = V_1\, D_{yb}^{-1/2}$. Then the between-class scatter matrix $\tilde{S}_b$ in the transformed space becomes

$$\tilde{S}_b = V_y^T S_b V_y = I_s. \tag{36}$$

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

$$V_y^T S_w V_y = W_y D_{yw} W_y^T, \tag{37}$$

where $W_y \in \mathbb{R}^{s\times s}$ is orthogonal and $D_{yw} \in \mathbb{R}^{s\times s}$ is a diagonal matrix. Then

$$W_y^T V_y^T S_b V_y W_y = I_s \quad \text{and} \quad W_y^T V_y^T S_w V_y W_y = D_{yw}. \tag{38}$$

In most applications, rank(S_w) is greater than rank(S_b), and D_{yw} is nonsingular since

$$\text{rank}(W_y^T V_y^T S_w V_y W_y) = \text{rank}(S_w) \geq \text{rank}(S_b) = \text{rank}(W_y^T V_y^T S_b V_y W_y) = s.$$

Scaling (38) by $D_{yw}^{-1/2}$, we have

$$(D_{yw}^{-1/2} W_y^T V_y^T) S_b (V_y W_y D_{yw}^{-1/2}) = D_{yw}^{-1} \quad \text{and} \quad (D_{yw}^{-1/2} W_y^T V_y^T) S_w (V_y W_y D_{yw}^{-1/2}) = I_s. \tag{39}$$

The authors in [6] proposed the transformation matrix

$$G_y = V_y W_y D_{yw}^{-1/2}.$$

Eqs. in (39) imply that each column of G_y belongs to null$(S_w)^c \cap$ null$(S_b)^c$.

Two – Class Problem

In a two-class problem, since

$$S_b = \rho(c_1 - c_2)(c_1 - c_2)^T = \left(\frac{c_1 - c_2}{\|c_1 - c_2\|_2}\right) \rho \|c_1 - c_2\|_2^2 \left(\frac{c_1 - c_2}{\|c_1 - c_2\|_2}\right)^T \tag{40}$$

where $\rho = n_1 n_2 / n$, a data item is transformed to the 1-dimensional space by

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

The dimension reduced representation of any data item a is given by

$$g^T a = \nu(c_1 - c_2)^T a$$

Data	Transformation matrix	Reduced dim.	kNN		
			$k=1$	$k=5$	$k=9$
AT&T database	$G_y = V_y W_y$	39	94.3	90.5	89.3
$2576\times$ 400, 40 classes	$G_y = V_y W_y D_{yw}^{-1/2}$	39	99.0	98.3	97.8
Yale database	$G_y = V_y W_y$	14	80.6	78.8	80.6
8586×165, 15 classes	$G_y = V_y W_y D_{yw}^{-1/2}$	14	89.7	94.6	91.5

Table 2: The prediction accuracies(%)

for some scalar ν. Note that no minimization of within-class scatter in the transformed space is possible.

The optimization criteria in (7) is 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,

$$J_1(G) = J_1(GF) \quad \text{and} \quad J_2(G) = J_2(GF). \tag{41}$$

In the transformation matrix

$$G_y = V_y W_y D_{yw}^{-1/2}$$

obtained by this method, $D_{yw}^{-1/2}$ or $W_y D_{yw}^{-1/2}$ do not influence the value of the optimization criteria, since

$$J_i(V_y) = J_i(V_y W_y) = J_i(V_y W_y D_{yw}^{-1/2}), \quad i = 1, 2. \tag{42}$$

Therefore, none of the components involved in the second step (those in (37-39) improves the optimization criteria. However, the following experimental results show that the scaling by $D_{yw}^{-1/2}$ can make dramatic effects on the classification performances. Postponing the detailed explanation on the datasets and experimental setting until Section 6, experimental results on the face recognition datasets are shown in Table 2. After dimension reduction by above method, kNN classifiers were used in the reduced dimensional space.

4 A New Efficient Algorithm for LDA/GSVD

As demonstrated by our experiments in Section 6, LDA/GSVD gives highly accurate prediction results. However, the high dimensionality makes the computational complexity of LDA/GSVD a major problem. Now, we propose an efficient algorithm for LDA/GSVD and compare the complexities of methods discussed.

Recall that in Algorithm 1 LDA/GSVD, the SVD of the matrix K,

$$K \equiv \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Q^T,$$

and the SVD of $P(1:r, 1:t)$,

$$P(1:r, 1:t) = U_{hb} \Sigma_{hb} W_h^T,$$

are computed, where R is a diagonal matrix with positive and nonincreasing diagonal elements. The optimal transformation is given by $r-1$ leftmost columns of

$$X = Q \begin{bmatrix} R^{-1} W_h & 0 \\ 0 & I \end{bmatrix}$$

with which S_b and S_w are simultaneously diagonalized as

$$X^T S_b X = \begin{bmatrix} \Sigma_{hb}^T \Sigma_{hb} & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad X^T S_w X = \begin{bmatrix} \Sigma_{hw}^T \Sigma_{hw} & 0 \\ 0 & 0 \end{bmatrix}. \tag{43}$$

Since

$$\Sigma_{hb}^T \Sigma_{hb} + \Sigma_{hw}^T \Sigma_{hw} = I_t$$

by Theorem 1, we have

$$X^T S_m X = X^T S_b X + X^T S_w X = \begin{bmatrix} I_t & 0 \\ 0 & 0 \end{bmatrix} \tag{44}$$

where $t = \text{rank}(K) = \text{rank}\left(\begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \right)$. Eq. (44) shows that $t = \text{rank}(S_m)$ and

$$S_m = X^{-T} \begin{bmatrix} I_t & 0 \\ 0 & 0 \end{bmatrix} X^{-1} = Q \begin{bmatrix} R^2 & 0 \\ 0 & 0 \end{bmatrix} Q^T \tag{45}$$

which results in the EVD of S_m. Partitioning Q as

$$Q = [\underbrace{Q_1}_{t} \underbrace{Q_2}_{m-t}],$$

we have

$$X = Q \begin{bmatrix} R^{-1}W_h & 0 \\ 0 & I_{m-t} \end{bmatrix} = \begin{bmatrix} Q_1 R^{-1} W_h & Q_2 \end{bmatrix} \tag{46}$$

and from the first equation in (43),

$$R^{-1}Q_1^T S_b Q_1 R^{-1} = W_h \Sigma_{hb}^T \Sigma_{hb} W_h^T. \tag{47}$$

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 $Q_1 R^{-1} W_h$.

Now, (45) and (47) show that the solution to LDA/GSVD can be obtained by computing

(1) Q_1 and R from the EVD of S_m,

(2) W_h from the EVD of $R^{-1}Q_1^T S_b Q_1 R^{-1}$.

The matrices Q_1 and R in the EVD of $S_m \in \mathbb{R}^{m \times m}$ can be obtained by the SVD of $H_m \in \mathbb{R}^{m \times n}$ as in the case of S_w and H_w in (15-16). However, computing the SVD of H_m can be expensive when the dimension m is very large. Instead, Q_1 and R can be obtained by the EVD of $n \times n$ matrix $H_m^T H_m$ [2]. Let the EVD of $H_m^T H_m$ be

$$H_m^T H_m = [\underbrace{J_1}_{t} \underbrace{J_2}_{n-t}] \begin{bmatrix} D_t & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J_1^T \\ J_2^T \end{bmatrix}, \tag{48}$$

where $t = \text{rank}(H_m) = \text{rank}(S_m)$. Then the columns in $H_m J_1$ are eigenvectors of S_m corresponding to nonzero eigenvalues in the diagonal of D_t. Since

$$(H_m J_1)^T (H_m J_1) = D_t$$

from (48), we obtain the orthonormal eigenvectors and corresponding nonzero eigenvalues of S_m by $H_m J_1 D_t^{-1/2}$ and D_t, which are Q_1 and R^2 respectively. Then, W_h is obtained by the EVD of $R^{-1}Q_1^T S_b Q_1 R^{-1}$,

$$R^{-1}Q_1^T S_b Q_1 R^{-1} = W_h \Sigma W_h^T,$$

Algorithm 2 A new algorithm for LDA/GSVD

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with r classes, it computes the dimension reducing transformation $G_h \in \mathbb{R}^{m \times (r-1)}$. For any data item $a \in \mathbb{R}^m$, $r-1$ dimensional representation is given by $G_h^T a$.

1. Compute the EVD of $H_m^T H_m$, which is

$$H_m^T H_m = [\underbrace{J_1}_{t} \; \underbrace{J_2}_{n-t}] \begin{bmatrix} D_t & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J_1^T \\ J_2^T \end{bmatrix}$$

   where $t = \text{rank}(H_m)$. Let $Q_1 = H_m J_1 D_t^{-1/2}$ and $R = D_t^{1/2}$.

2. Compute W_h from the EVD of $Z_h = R^{-1} Q_1^T S_b Q_1 R^{-1}$, which is $Z_h = W_h \Sigma W_h^T$.

3. Compute the first $r-1$ columns of $Q_1 R^{-1} W_h$ and assign them to G_h.

where $W_h^T W_h = I$ and Σ is a diagonal matrix. The new algorithm for LDA/GSVD is summarized in Algorithm 2. In the algorithm proposed, we just need to compute the EVD of a much smaller $n \times n$ matrix $H_m^T H_m$ instead of $m \times m$ matrix $S_m = H_m H_m^T$ when $m >> n$. However, in the regularized LDA or the method by Chen et al., 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 null(S_w) needs to compute a basis of null(S_w) which are eigenvectors corresponding to zero eigenvalues.

Two – Class Problem

Now we consider two-class problem in LDA/GSVD. With the representation of S_b in (18), we have

$$R^{-1} Q_1^T S_b Q_1 R^{-1} = R^{-1} Q_1^T \rho (c_1 - c_2)(c_1 - c_2)^T Q_1 R^{-1} = \left(\frac{w}{\|w\|_2} \right) \rho \|w\|_2^2 \left(\frac{w}{\|w\|_2} \right)^T,$$

where $w = R^{-1} Q_1^T (c_1 - c_2)$. Hence the transformation matrix $g \in \mathbb{R}^{m \times 1}$ is given by

$$g = \nu Q_1 R^{-1} w = \nu Q_1 R^{-2} Q_1^T (c_1 - c_2)$$

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

$$g^T a = \nu (c_1 - c_2)^T Q_1 R^{-2} Q_1^T a = \nu (c_1 - c_2)^T S_m^+ a,$$

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

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

and

$$g^T a = \nu(c_1 - c_2)^T S_m^{-1} a = \nu_1(c_1 - c_2)^T S_w^{-1} a$$

for a scalar $\nu_1 = \frac{\nu}{1+\rho(c_1-c_2)^T S_w^{-1}(c_1-c_2)}$.

5 Computational Complexities

Algorithms and computational complexities for the methods discussed in this paper are compared in Table 3 and Figure 2. Assuming that $H_b \in \mathbb{R}^{m\times r}$, $H_w \in \mathbb{R}^{m\times n}$ and $H_m \in \mathbb{R}^{m\times n}$ in (4-6) have already been computed, each algorithm in Table 3 computes the transformation matrix G. The table also shows the main decompositions and multiplications which are needed to compute the transformation matrix. For $S \in \mathbb{R}^{p\times q}$ and $p >> q$, when only eigenvectors of $SS^T \in \mathbb{R}^{p\times p}$ corresponding to the nonzero eigenvalues are needed, the algorithms in Table 3 utilize the approach of computing the EVD of $S^T S$ instead of SS^T as explained in Section 4.

Figure 2 illustrates the total complexities corresponding to the algorithms in Table 3 for specific sizes of training datasets used in the experiments. One flop (floating point operation) represents roughly what is required to do one addition/subtraction or one multiplication/division [12]. The computational complexity for the SVD decomposition depends on what parts need to be explicitly computed. For the SVD of a matrix $S \in \mathbb{R}^{p\times q}$ when $p >> q$,

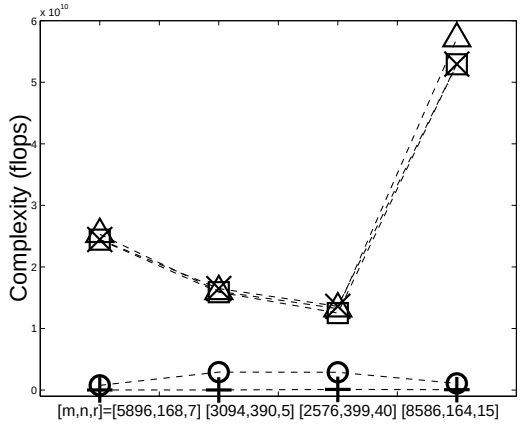
$$S = Y\Sigma Z^T = [\underbrace{Y_1}_{q}\ \underbrace{Y_2}_{p-q}]\Sigma Z^T,$$

where $Y \in \mathbb{R}^{p\times p}$, $\Sigma \in \mathbb{R}^{p\times q}$ and $Z \in \mathbb{R}^{q\times q}$, $6pq^2 + 11q^3$ flops were counted for computing Y_1 and Σ, $4p^2q + 13q^3$ for Y and Σ, and $4p^2q + 22q^3$ for Y and Σ and Z [12]. For the multiplication of the $p_1 \times p_2$ matrix and the $p_2 \times p_3$ matrix, $2p_1p_2p_3$ flops were counted. For the simplicity, the rank

Method	Complexity
Regularized LDA	
Compute the SVD of H_w : $H_w = UDY^T$.	SVD($m \times n$)
Let $\mathbf{E_f} = U(DD^T + \alpha I)^{-1/2}$ and $\mathbf{S_f} = E_f^T H_b$.	
Compute the EVD of $\mathbf{S_f^T S_f}$: $S_f^T S_f = Z_f \Sigma_f Z_f^T$.	EVD($r \times r$)
Let $\hat{\mathbf{W}}_\mathbf{f} = S_f(Z_f(:, 1:r-1)\Sigma_f(1:r-1, 1:r-1)^{-1/2})$.	
$\mathbf{G_f} = E_f \hat{W}_f$.	
LDA/GSVD	
Compute the SVD of $K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix}$: $K = P \begin{bmatrix} R \\ 0 \end{bmatrix} Q^T$, where $t = \text{rank}(K)$ and $R \in \mathbb{R}^{t \times t}$.	SVD($m \times (r+n)$)
Compute the EVD of $\mathbf{P(1:r, 1:t)}\,\mathbf{P(1:r, 1:t)^T}$: $P(1:r, 1:t)\,P(1:r, 1:t)^T = Z_h \Sigma_h\, Z_h^T$.	EVD($r \times r$)
Let $\hat{\mathbf{W}}_\mathbf{h} = P(1:r, 1:t)^T\,(Z_h(:, 1:r-1)\,\Sigma_h(1:r-1, 1:r-1)^{-1/2})$.	
$\mathbf{G_h} = Q(:, 1:t)\,(R^{-1}\hat{W}_h)$.	
Proposed LDA/GSVD	
Compute the EVD of $\mathbf{H_m^T H_m}$: $H_m^T H_m = J D_p J^T$, where $t = \text{rank}(H_m)$.	EVD($n \times n$)
Let $\mathbf{E_p} = H_m(J(:, 1:t)D_p(1:t, 1:t)^{-1})$ and $\mathbf{S_p} = E_p^T H_b$.	
Compute the EVD of $\mathbf{S_p^T S_p}$: $S_p^T S_p = Z_p \Sigma_p Z_p^T$.	EVD($r \times r$)
Let $\hat{\mathbf{W}}_\mathbf{h} = S_p(Z_p(:, 1:r-1)\Sigma_p(1:r-1, 1:r-1)^{-1/2})$.	
$\mathbf{G_h} = E_p \hat{W}_h$.	
Projection to null(S_w)	
Compute the SVD of H_w : $H_w = UDY^T$.	SVD($m \times n$)
Let $\mathbf{S_e} = U(:, s_1 + 1:m)(U(:, s_1 + 1:m)^T H_b)$, where $s_1 = \text{rank}(H_w)$.	

Compute the EVD of $\mathbf{S_e^T S_e}$: $S_e^T S_e = Z_e \Sigma_e Z_e^T$.	EVD($r \times r$)
Let $\hat{\mathbf{W}}_\mathbf{e} = S_e(Z_e(:, 1:s_2)\Sigma_e(1:s_2, 1:s_2)^{-1/2})$, where $s_2 = \text{rank}(S_e)$.	
$\mathbf{G_e} = U(:, s_1+1:m)(U(:, s_1+1:m)^T \hat{W}_e)$.	
Transformation to range(S_b)	
Compute the EVD of $\mathbf{H_b^T H_b}$: $H_b^T H_b = F D_y F^T$, where $s = \text{rank}(H_b)$.	EVD($r \times r$)
Let $\mathbf{V_y} = H_b(F(:, 1:s)D_y(1:s, 1:s)^{-1})$ and $\mathbf{S_y} = V_y^T H_w$.	
Compute the EVD of $\mathbf{S_y S_y^T}$: $S_y S_y^T = W_y \Sigma_y W_y^T$,	EVD($r \times r$)
$\mathbf{G_y} = V_y(W_y \Sigma_y^{-1/2})$.	

Table 3: Comparison of main decompositions and multiplications for each algorithm. Multiplications needed are shown in boldface. Assuming that $H_b \in \mathbb{R}^{m \times r}$, $H_w \in \mathbb{R}^{m \times n}$ and $H_m \in \mathbb{R}^{m \times n}$ according to (4), (5) and (6) are given, the linear transformation matrix $G \in \mathbb{R}^{m \times l}$ is computed in each algorithm. For any data item $a \in \mathbb{R}^{m \times 1}$, the reduced dimensional representation is $G^T a \in \mathbb{R}^{l \times 1}$. ($m$: dimension of data, n: number of training data, r: number of classes)



(□) Regularized LDA
(X) LDA/GSVD
(△) Projection to null(S_w)
(O) Proposed LDA/GSVD
(+) Transformation to range(S_b)

Figure 2: Comparison of computational complexities. The figure compares the flops required according to the algorithms given in Table 3 for specific problem sizes. From the left, the sizes of training data used in experiments are [5896,168,7], [3094,390,5], [2576,399,40] and [8586,164,15], where $[m, n, r]$ denotes [dimension, no. of data, no. of classes].

was estimated as the number of data samples, n, or the number of classes, r, since it was close to either of two in most cases. Figure 2 shows that the proposed LDA/GSVD algorithm reduced the complexity of the original LDA/GSVD algorithm dramatically.

In the next section, we present extensive experimental results of the algorithms discussed in this paper using datasets in text classification and face recognition in order to compare their performances.

6 Experimental Results

6.1 Text Classification

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. 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 [8, 13]. The term-document matrix representation often makes the high dimensionality inevitable.

In our experiment, two text data sets were used [11]. Dataset 1 was collected from the TREC database. It has 7 classes with 5896 terms and 210 documents, resulting in the term-document matrix of size 5896×210. Dataset 2 was from Reuters-21578 database, which has a term-document matrix of 3094×490 with 5 classes. For all datasets, they are randomly split to the training set and the test set with the ratio of $4:1$ and experiments are repeated 10 times to reduce the possible bias caused from splitting data to training and test sets. In the reduced dimensional spaces, k-nearest neighbor classifier was used. Table 4 reports the mean prediction accuracies and standard deviations. The method based on the transformation to range(S_b) gives low prediction accuracies in both datasets compared with other methods, even though the computational complexity is low. The proposed fast algorithm for LDA/GSVD achieves competitive prediction accuracies while it requires low computational complexity. This new algorithm can save computational complexities even more when the number of terms is much greater than the number of documents.

Data	Method	Dim.	kNN 1	kNN 15	kNN 29
Data 1 5896×210 7 classes	Original	5896	91.7 (4.2)	92.9 (3.7)	91.0 (4.5)
	Regularized LDA ($\lambda = 1.0$)	6	95.7 (2.9)	97.4 (2.1)	96.7 (2.0)
	LDA/GSVD	6	**98.3** (2.3)	**98.3** (2.3)	**98.3** (2.3)
	Projection to null(S_w)	6	98.1 (1.9)	98.1 (1.9)	98.1 (1.9)
	Transformation to range(S_b)	6	96.7 (2.0)	96.0 (2.5)	96.2 (2.0)
Data 2 3094×490 5 classes	Original	3094	88.0 (3.2)	91.1 (1.7)	90.1 (1.9)
	Regularized LDA ($\lambda = 1.0$)	4	**95.5** (1.6)	**95.2** (1.2)	94.9 (1.3)
	LDA/GSVD	4	95.1 (1.4)	95.1 (1.4)	**95.1** (1.4)
	Projection to null(S_w)	4	94.5 (1.8)	94.5 (1.8)	94.5 (1.8)
	Transformation to range(S_b)	4	94.2 (2.5)	94.9 (2.0)	94.1 (2.0)

Table 4: Prediction accuracies(%). Mean and standard deviation from 10 runnings. For each k in kNN, the best prediction accuracy is shown in boldface. Standard deviation are shown in the parenthesis.

6.2 Face Recognition

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 datasets, 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

Figure 3: The first row shows face images in one class of AT&T database. The last two rows show the face images in one class of Yale face database.

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 [14]. 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 [15]. In our experiment, each image has been downsampled from 320×243 to 106×81 by averaging the grey values on 3×3 blocks. Examples of face images in each database are shown in Figure 3.

Since the number of images for each subject is small, the experimental condition to set up training data and test data can effect the performance greatly. The experiments are performed using two different strategies: random splitting and leave-one-out method. By randomly splitting face images of each subject to training and test data with the ratio 1:1 and repeating the experiments 100 times, the prediction accuracies were measured. Figure 4 shows the boxplots for each methods with different k values in kNN classifiers, where the box has lines at the lower quartile, median, and upper quartile values in prediction accuracies from 100 runnings, and the whiskers are lines extending from each end of the box to show the extent of the rest. The high variances in 100

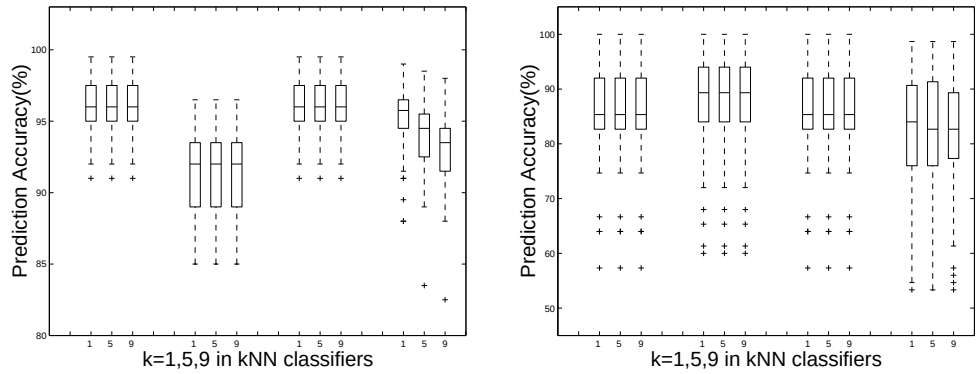


Figure 4: Prediction accuracies by random splitting. The left figure was obtained by using AT&T database and the right figure by Yale database. In each figure, from the left, box plots are shown for the methods of Regularized LDA, LDA/GSVD, Projection to null(S_w), Transformation to range(S_b), for k=1,5,9 in kNN classifiers.

runnings show that the performance comparisons among different methods should be interpreted carefully regarding various splitting methods for training and test sets.

Leave-one-out method is performed by taking one image for test set and the remaining images are used as a training set. Each image serves as a test datum and the ratio of the number of correctly classified cases and the total number of data is considered as a prediction accuracy. After training data and test data are represented in the reduced dimensional space, kNN classifiers are applied for classification. Table 5 compares the experimental results.

The experiment using AT&T database shows somewhat different results from other experiments in text classification and face recognition using Yale database. The best prediction accuracy was achieved by the method of transformation to range(S_b), while LDA/GSVD obtained lower prediction accuracy. We could not find theoretical reasoning about the poor performance of LDA/GSVD in this test. We conjecture there might be overfitting caused by the projection onto vectors from null(S_w) $\cap$ null(S_w), considering that the method by the transformation to range(S_b) obtains all the projection vectors from null$(S_b)^c \cap$ null$(S_w)^c$.

Data	Method	Dim.	kNN 1	kNN 5	kNN 9
AT&T database $2576\times$ 400 40 classes	Original	2576	97.8	91.8	81.8
	Regularized LDA ($\lambda = 1.0$)	39	98.0	98.0	**98.0**
	LDA/GSVD	39	93.5	93.5	93.5
	Projection to null(S_w)	39	98.0	98.0	**98.0**
	Transformation to range(S_b)	39	**99.0**	**98.3**	97.8
Yale database 8586×165 15 classes	Original	8586	79.4	76.4	72.1
	Regularized LDA ($\lambda = 1.0$)	14	97.6	97.6	97.6
	LDA/GSVD	14	**98.8**	**98.8**	**98.8**
	Projection to null(S_w)	14	97.6	97.6	97.6
	Transformation to range(S_b)	14	89.7	94.6	91.5

Table 5: Prediction accuracies(%) by leave-one-out strategy. For each k in kNN, the best prediction accuracy is shown in boldface.

7 Conclusions/Discussions

In this paper, we presented the relationships among the generalized Linear Discriminant Analysis algorithms developed for handling undersampled problems and compared their computational complexities and performances. The comparison of computational complexities and classification performances in text classification and face recognition show that the performances may depend on the properties of datasets and no single method works the best in all situations. It also demonstrates the necessity of practical algorithm to solve undersampled problems.

The LDA/GSVD showed competitive performances throughout the experiments, but the com-

putational 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. Combining the proposed efficient algorithm for LDA/GSVD and the nonlinear generalization by kernel method [16], it is expected that high performance and efficiency can be achieved in many applications.

References

[1] R.O. Duda, P.E. Hart, and D.G. Stork. *Pattern Classification*. Wiley-interscience, New York, 2001.

[2] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Acadamic Press, second edition, 1990.

[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] P.C. Hansen. *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. SIAM, Philadelphia, PA, 1997.

[8] G. Salton and M.J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, 1983.

[9] 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.

[10] C.C. Paige and M.A. Saunders. Towards a generalized singular value decomposition. *SIAM Journal on Numerical Analysis*, 18:398–405, 1981.

[11] J. Ye, R. Janardan, C.H. Park, and H. Park. A new optimization criterion for generalized discriminant analysis on undersampled problems. The proceedings of the 3rd IEEE international conference on Data Mining. 419–426,2003.

[12] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition, 1996.

[13] G. Kowalski. *Information Retrieval System: Theory and Implementation*. Kluwer Academic Publishers, 1997.

[14] http://www.uk.research.att.com/facedatabase.html.

[15] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

[16] C.H. Park and H. Park. Kernel discriminant analysis based on generalized singular value decomposition. Technical Reports 03-017, Department of Computer Science and Engineering, University of Minnesota, Twin Cities, 2003.

A Generalized Singular Value Decomposition

THEOREM 3 (GENERALIZED SINGULAR VALUE DECOMPOSITION [10]) *Suppose two matrices with the same number of columns, $K_b \in \mathbb{R}^{p_1 \times q}$ and $K_w \in \mathbb{R}^{p_2 \times q}$ are given. Then for*

$$K = \begin{bmatrix} K_b \\ K_w \end{bmatrix} \quad \textit{and} \quad t = \textit{rank}(K),$$

there exist orthogonal matrices $U_b \in \mathbb{R}^{p_1 \times p_1}$ *and* $U_w \in \mathbb{R}^{p_2 \times p_2}$, *and a nonsingular matrix* $X \in \mathbb{R}^{q \times q}$ *such that*

$$U_b^T K_b X = [\Sigma_b \quad 0] \quad \textit{and} \quad U_w^T K_w X = [\Sigma_w \quad 0], \tag{49}$$

where

$$\Sigma_b = \underbrace{\begin{bmatrix} I_b & & \\ & D_b & \\ & & 0_b \end{bmatrix}}_{\mu \quad \tau \quad t-\mu-\tau} \begin{matrix} \}\mu \\ \}\tau \\ \}p_1 - \mu - \tau \end{matrix} \quad \textit{and} \quad \Sigma_w = \underbrace{\begin{bmatrix} 0_w & & \\ & D_w & \\ & & I_w \end{bmatrix}}_{\mu \quad \tau \quad t-\mu-\tau} \begin{matrix} \}p_2 - t + \mu \\ \}\tau \\ \}t - \mu - \tau \end{matrix}. \tag{50}$$

Here μ *and* τ *are associated with*

$$\mu = t - \textit{rank}(K_w) \quad \textit{and} \quad \tau = \textit{rank}(K_b) + \textit{rank}(K_w) - t \tag{51}$$

and

$$\begin{aligned} &D_b = diag(\eta_{\mu+1}, \ldots, \eta_{\mu+\tau}) \quad \textit{for} \quad 1 > \eta_{\mu+1} \geq \cdots \geq \eta_{\mu+\tau} > 0, \\ &D_w = diag(\zeta_{\mu+1}, \ldots, \zeta_{\mu+\tau}) \quad \textit{for} \quad 0 < \zeta_{\mu+1} \leq \cdots \leq \zeta_{\mu+\tau} < 1, \\ &\eta_i^2 + \zeta_i^2 = 1 \quad \textit{for } i = \mu+1, \ldots, \mu+\tau. \qquad \square \end{aligned}$$

Proof. (See [10] for the details.) For the matrix K, there exist orthogonal matrices $P \in \mathbb{R}^{(p_1+p_2)\times(p_1+p_2)}$ and $Q \in \mathbb{R}^{q \times q}$, and a nonsingular matrix $R \in \mathbb{R}^{t \times t}$ such that

$$\begin{bmatrix} K_b \\ K_w \end{bmatrix} = P \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix} Q^T, \tag{52}$$

according to the SVD (or a complete orthogonal decomposition) [12]. Partition the first t columns of P as

$$P(:, 1:t) = \begin{bmatrix} P_{p_1 \times t} \\ P_{p_2 \times t} \end{bmatrix}.$$

Then it can be shown that the SVD of $P_{p_1 \times t}$ is given by

$$P_{p_1 \times t} = U_b \Sigma_b W^T$$

where U_b and W are orthogonal matrices and Σ_b is the matrix in (21), since $\|P_{p_1\times t}\|_2 \leq 1$. Now $P_{p_2\times t}W$ can be decomposed as U_wL, where $U_w \in \mathbb{R}^{p_2\times p_2}$ is orthogonal and $L \in \mathbb{R}^{p_2\times t}$ is a lower triangular. Then $\begin{bmatrix} \Sigma_b \\ L \end{bmatrix}$ has orthonormal columns and therefore $L = \Sigma_w$ as in (50). Hence

$$\begin{bmatrix} K_b \\ K_w \end{bmatrix} Q = \begin{bmatrix} U_b\Sigma_bW^TR & 0 \\ U_w\Sigma_wW^TR & 0 \end{bmatrix} = \begin{bmatrix} U_b & 0 \\ 0 & U_w \end{bmatrix} \begin{bmatrix} \Sigma_b & 0 \\ \Sigma_w & 0 \end{bmatrix} \begin{bmatrix} W^TR & 0 \\ 0 & I_{q-t} \end{bmatrix}. \tag{53}$$

Defining

$$X = Q \begin{bmatrix} R^{-1}W & 0 \\ 0 & I_{q-t} \end{bmatrix}, \tag{54}$$

from (53), we have

$$U_b^TK_bX = [\Sigma_b \quad 0] \quad \text{and} \quad U_w^TK_wX = [\Sigma_w \quad 0]. \quad \square$$