PROD. TYPE: COM
PP: 1-15 (col.fig.: nil)
PR 3000 MODE+
ED: Praveen
PAGN: Suresh – SCAN: ICC
ARTICLE IN PRESS
Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition ▮▮▮ (▮▮▮▮) ▮▮▮–▮▮▮
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/pr

A comparison of generalized linear discriminant analysis algorithms

Cheong Hee Park[a,*,1], Haesun Park[b2]

[a]Department of Computer Science and Engineering, Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon 305 763, Republic of Korea
[b]College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332, USA

Received 26 April 2006; received in revised form 27 June 2007; accepted 30 July 2007

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 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

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

where a collection of data items in the class i ($1 \leqslant i \leqslant r$) 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

$$S_b = \sum_{i=1}^{r} n_i (c_i - c)(c_i - c)^{\mathrm{T}},$$

$$S_w = \sum_{i=1}^{r} \sum_{j\in N_i} (a_j - c_i)(a_j - c_i)^{\mathrm{T}},$$

$$S_t = \sum_{j=1}^{n} (a_j - c)(a_j - c)^{\mathrm{T}},$$

* Corresponding author. Tel.: +82 42 821 6293.
E-mail addresses: cheonghee@cnu.ac.kr (C.H. Park), hpark@cc.gatech.edu (H. Park).
[1] This study was financially supported by research fund of Chungnam National University in 2005.
[2] This work was supported in part by the National Science Foundation grant CCF-0621889. 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).

0031-3203/$30.00 © 2007 Published by Elsevier Ltd on behalf of Pattern Recognition Society.
doi:10.1016/j.patcog.2007.07.022

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

The optimal dimension reducing transformation $G^{\mathrm{T}} \in \mathbb{R}^{l\times m}$ $(l < m)$ 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

$$J_1(G) = \frac{\mathrm{trace}(G^{\mathrm{T}} S_b G)}{\mathrm{trace}(G^{\mathrm{T}} S_w G)},$$

$$J_2(G) = \mathrm{trace}((G^{\mathrm{T}} S_w G)^{-1}(G^{\mathrm{T}} S_b G)),$$

$$J_3(G) = \frac{|G^{\mathrm{T}} S_b G|}{|G^{\mathrm{T}} S_w G|}, \tag{2}$$

where $\tilde{S}_i = G^{\mathrm{T}} S_i G$ 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

$$S_w^{-1} S_b g = \lambda g. \tag{3}$$

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 [3–8]. 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 Refs. [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.

For convenience, important notations used throughout the rest of the paper are listed in Table 1.

Table 1
Summary of the notations used

Notations	Description
m	Data dimension
n	Number of data items
r	Number of classes
n_i	Number of data items in class i
c, c_i	The global and class centroids
A	Data matrix of size $m \times n$
S_b, S_w, S_t	Scatter matrices of size $m \times m$
H_b	The matrix of size $m \times r$ such that $S_b = H_b H_b^{\mathrm{T}}$
H_w	The matrix of size $m \times n$ such that $S_w = H_w H_w^{\mathrm{T}}$
H_t	The matrix of size $m \times n$ such that $S_t = H_t H_t^{\mathrm{T}}$
s	Rank of the matrix $[H_b\ H_w]$
I_τ, 0_τ	Identity and zero matrices of size $\tau \times \tau$

2. A comparison of generalized LDA algorithms for undersampled problems

2.1. Regularized LDA

In the regularized LDA (RLDA) [3], 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

$$S_b g = \lambda(S_w + \alpha I) g. \tag{4}$$

2.1.1. 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

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

and the eigenvalue problem (3) is simplified to

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

when S_w is nonsingular. The solution for Eq. (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

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

where $S_w = U_w \Sigma_w U_w^{\mathrm{T}}$ is the eigenvalue decomposition (EVD) of S_w. Since $S_w + \alpha I = U_w(\Sigma_w + \alpha I) U_w^{\mathrm{T}}$, the regularized LDA

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

gives the solution

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

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 GSVD

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

$$U_1^{\mathrm{T}} Z_1 X = [\underbrace{\Gamma_1}_{\gamma} \underbrace{0}_{p-\gamma}] \quad \text{and} \quad U_2^{\mathrm{T}} Z_2 X = [\underbrace{\Gamma_2}_{\gamma} \underbrace{0}_{p-\gamma}]$$

$$\text{for} \quad \gamma = \operatorname{rank}\left(\begin{bmatrix} Z_1 \\ Z_2 \end{bmatrix}\right),$$

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

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

$$S_b = H_b H_b^{\mathrm{T}}, \quad S_w = H_w H_w^{\mathrm{T}}, \quad \text{and} \quad S_t = H_t H_t^{\mathrm{T}}. \tag{7}$$

where

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

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

$$H_t = [a_1 - c, \ldots, a_n - c] \in \mathbb{R}^{m \times n}, \tag{10}$$

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

$$U_b^{\mathrm{T}} H_b^{\mathrm{T}} X = [\Gamma_b \ \ 0] \quad \text{and} \quad U_w^{\mathrm{T}} H_w^{\mathrm{T}} X = [\Gamma_w \ \ 0], \tag{11}$$

where $U_b \in \mathbb{R}^{r \times r}$ and $U_w \in \mathbb{R}^{n \times n}$ are orthogonal, $X \in \mathbb{R}^{m \times m}$ is nonsingular, and

$$\Gamma_b^{\mathrm{T}}\Gamma_b + \Gamma_w^{\mathrm{T}}\Gamma_w = I_s \quad \text{for } s = \operatorname{rank}\left(\begin{bmatrix} H_b^{\mathrm{T}} \\ H_w^{\mathrm{T}} \end{bmatrix}\right).$$

Then Eqs. in (11) give

$$X^{\mathrm{T}} S_b X = X^{\mathrm{T}}(H_b H_b^{\mathrm{T}})X = (X^{\mathrm{T}} H_b U_b)(U_b^{\mathrm{T}} H_b^{\mathrm{T}} X)$$
$$= [\Gamma_b \ \ 0]^{\mathrm{T}}[\Gamma_b \ \ 0],$$

$$X^{\mathrm{T}} S_w X = X^{\mathrm{T}}(H_w H_w^{\mathrm{T}})X = (X^{\mathrm{T}} H_w U_w)(U_w^{\mathrm{T}} H_w^{\mathrm{T}} X)$$
$$= [\Gamma_w \ \ 0]^{\mathrm{T}}[\Gamma_w \ \ 0]. \tag{12}$$

From (12) and $\Gamma_b^{\mathrm{T}}\Gamma_b + \Gamma_w^{\mathrm{T}}\Gamma_w = I_s$, we have

$$X^{\mathrm{T}} S_b X = \begin{bmatrix} \Gamma_b^{\mathrm{T}}\Gamma_b & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} I_\mu & & & \\ & D_\tau & & \\ & & 0_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix} \tag{13}$$

and

$$X^{\mathrm{T}} S_w X = \begin{bmatrix} \Gamma_w^{\mathrm{T}}\Gamma_w & \\ & 0_{m-s} \end{bmatrix} \equiv \begin{bmatrix} 0_\mu & & & \\ & E_\tau & & \\ & & I_{s-\mu-\tau} & \\ & & & 0_{m-s} \end{bmatrix}, \tag{14}$$

where $D_\tau + E_\tau = I_\tau$ and the subscripts in I and 0 denote the size of square identity and zero matrices. Denoting the diagonal elements in $\Gamma_b^{\mathrm{T}}\Gamma_b$ as η_i's and the diagonal elements in $\Gamma_w^{\mathrm{T}}\Gamma_w$ as ζ_i's, we have

$$\zeta_i S_b x_i = \eta_i S_w x_i, \quad i = 1, \ldots, m, \tag{15}$$

where x_i is the column vectors of X. Note that x_i, $i = s+1, \ldots, m$, belong to $\operatorname{null}(S_b) \cap \operatorname{null}(S_w)$. Hence η_i and ζ_i for $i = s+1, \ldots, m$ in Eq. (15) can be any arbitrary numbers.

By partitioning X in Eqs. (13)–(14) as

$$X = [\underbrace{X_1}_{\mu} \underbrace{X_2}_{\tau} \underbrace{X_3}_{s-\mu-\tau} \underbrace{X_4}_{m-s}] \in \mathbb{R}^{m \times m}, \tag{16}$$

the generalized eigenvalues and eigenvectors obtained by the GSVD can be classified as shown in Table 2. For the last $m-s$ vectors x belonging to $\operatorname{null}(S_w) \cap \operatorname{null}(S_b)$,

$$0 = x^{\mathrm{T}} S_b x = (x^{\mathrm{T}} H_b)(H_b^{\mathrm{T}} x) = \|x^{\mathrm{T}} H_b\|^2 = \sum_{i=1}^{r} n_i |x^{\mathrm{T}} c_i - x^{\mathrm{T}} c|^2$$

and

$$0 = x^{\mathrm{T}} S_w x = \sum_{j=1}^{n} |x^{\mathrm{T}} a_j - x^{\mathrm{T}} c_i|^2,$$

where a_j belongs to a class i.

Hence

$$\begin{cases} x^{\mathrm{T}} c_i = x^{\mathrm{T}} c & \text{for } i = 1, \ldots, r, \\ x^{\mathrm{T}} a_j = x^{\mathrm{T}} c_i & \text{for all } a_j \text{ in a class } i, \end{cases} \tag{17}$$

therefore

$$X_4^{\mathrm{T}} z = X_4^{\mathrm{T}} c \tag{18}$$

for any given data item $z = a_i$. This implies that the vectors x_i, $i = s+1, \ldots, m$, belonging to $\operatorname{null}(S_b) \cap \operatorname{null}(S_w)$ do not convey discriminative information among the classes, even though

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

Table 2
Generalized eigenvalues λ_i's and eigenvectors x_i's from the GSVD

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

The superscript c denotes the complement.

the corresponding eigenvalues are not necessarily zeros. Since $\text{rank}(S_b) \leqslant r-1$, from Eqs. (13)–(14), we have

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

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.

2.2.1. An efficient algorithm for LDA/GSVD

The algorithm to compute the GSVD for the pair $(H_b^{\mathrm{T}}, H_w^{\mathrm{T}})$ was presented in Ref. [4] as follows:

(1) Compute the singular value decomposition (SVD) of

$$Z = \begin{bmatrix} H_b^{\mathrm{T}} \\ H_w^{\mathrm{T}} \end{bmatrix} \in \mathbb{R}^{(r+n)\times m} : Z = P \begin{bmatrix} \Lambda & 0 \\ 0 & 0 \end{bmatrix} U^{\mathrm{T}}$$

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 $P(1:r, 1:s)$,[3] which is $P(1:r, 1:s) = W \Gamma V^{\mathrm{T}}$.

(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 $\Gamma_b^{\mathrm{T}} \Gamma_b + \Gamma_w^{\mathrm{T}} \Gamma_w = I_s$, from Eqs. (13)–(14), we have

$$X^{\mathrm{T}} S_t X = X^{\mathrm{T}} S_b X + X^{\mathrm{T}} S_w X = \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix}, \tag{19}$$

where $s = \text{rank}(Z)$. Eq. (19) implies $s = \text{rank}(S_t)$ and from step 3 in the LDA/GSVD algorithm

$$S_t = X^{-T} \begin{bmatrix} I_s & 0 \\ 0 & 0 \end{bmatrix} X^{-1} = U \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} U^{\mathrm{T}}, \quad \Sigma_1 = \Lambda^{\mathrm{T}} \Lambda, \tag{20}$$

which results in the EVD of S_t. Partitioning U as $U = [\underbrace{U_1}_{s} \underbrace{U_2}_{m-s}]$, we have

$$X = U \begin{bmatrix} \Lambda^{-1} V & 0 \\ 0 & I \end{bmatrix} = [U_1 \Sigma_1^{-1/2} V \;\; U_2]. \tag{21}$$

By substituting X in Eq. (13) with Eq. (21),

$$\Sigma_1^{-1/2} U_1^{\mathrm{T}} S_b U_1 \Sigma_1^{-1/2} = V \Gamma_b^{\mathrm{T}} \Gamma_b V^{\mathrm{T}}. \tag{22}$$

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. (20) and (22) show that U_1 and Σ_1 can be computed from the EVD of S_t and V from the EVD of $\Sigma_1^{-1/2} U_1^{\mathrm{T}} S_b U_1 \Sigma_1^{-1/2}$. This new approach for LDA/GSVD is summarized in Algorithm 1.

Algorithm 1. An efficient algorithm for LDA/GSVD.

(1) Compute the EVD of S_t: $S_t = [U_1 \;\; U_2] \begin{bmatrix} \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} U_1^{\mathrm{T}} \\ U_2^{\mathrm{T}} \end{bmatrix}$.

(2) Compute V from the EVD of $\tilde{S}_b \equiv \Sigma_1^{-1/2} U_1^{\mathrm{T}} S_b U_1 \Sigma_1^{-1/2}$: $\tilde{S}_b = V \Gamma_b^{\mathrm{T}} \Gamma_b V^{\mathrm{T}}$.

(3) Assign the first $r-1$ columns of $U_1 \Sigma_1^{-1/2} V$ to G_h.

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 $H_t^{\mathrm{T}} H_t \in \mathbb{R}^{n\times n}$ instead of $H_t H_t^{\mathrm{T}} \in \mathbb{R}^{m\times m}$ [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 $H_t^{\mathrm{T}} H_t$ be

$$H_t^{\mathrm{T}} H_t = [\underbrace{J_1}_{s} \underbrace{J_2}_{n-s}] \begin{bmatrix} D_1 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} J_1^{\mathrm{T}} \\ J_2^{\mathrm{T}} \end{bmatrix}, \tag{23}$$

where $s = \text{rank}(H_t) = \text{rank}(S_t)$. From Eq. (23)

$$S_t (H_t J_1) = H_t (H_t^{\mathrm{T}} H_t) J_1 = (H_t J_1) D_1,$$

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 $(H_t J_1)^{\mathrm{T}} (H_t J_1) = D_1$, 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 $H_t^{\mathrm{T}} H_t$ instead of $m \times m$ matrix $S_t = H_t H_t^{\mathrm{T}}$ when $m \gg n$. However, in the regularized LDA or the method by Chen et al. which is presented next, we cannot resort to this approach.

[3] The 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 rth row and from the first to sth column.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

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.

2.2.2. Two-class problem

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

$$\Sigma_1^{-1/2}U_1^{\mathrm{T}}S_bU_1\Sigma_1^{-1/2} = \Sigma_1^{-1/2}U_1^{\mathrm{T}}\rho(c_1-c_2)(c_1-c_2)^{\mathrm{T}}U_1\Sigma_1^{-1/2} = \left(\frac{w}{\|w\|_2}\right)\rho\|w\|_2^2\left(\frac{w}{\|w\|_2}\right)^{\mathrm{T}},$$

where $\rho = n_1n_2/n$ and $w = \Sigma_1^{-1/2}U_1^{\mathrm{T}}(c_1 - c_2)$. Hence the transformation matrix $g \in \mathbb{R}^{m\times 1}$ is given by

$$g = vU_1\Sigma_1^{-1/2}w = vU_1\Sigma_1^{-1}U_1^{\mathrm{T}}(c_1-c_2)$$

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

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

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

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

and

$$g^{\mathrm{T}}z = v(c_1-c_2)^{\mathrm{T}}S_t^{-1}z = v_1(c_1-c_2)^{\mathrm{T}}S_w^{-1}z \tag{24}$$

for a scalar $v_1 = v/(1+\rho(c_1-c_2)^{\mathrm{T}}S_w^{-1}(c_1-c_2))$. Eq. (24) shows that LDA/GSVD is equal to the classical LDA when S_w is nonsingular.

2.3. A method based on the projection onto null(S_w)

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 Refs. [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 this and next section where we also present their algebraic relationships.

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 $S_w \in \mathbb{R}^{m\times m}$,

$$S_w = U_w\Sigma_wU_w^{\mathrm{T}}.$$

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

$$\mathrm{null}(S_w) = \mathrm{span}(U_{w2}). \tag{25}$$

First, the transformation by $U_{w2}U_{w2}^{\mathrm{T}}$ 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 $\tilde{S}_b \equiv U_{w2}U_{w2}^{\mathrm{T}}S_bU_{w2}U_{w2}^{\mathrm{T}}$ be

$$\tilde{S}_b = \tilde{U}_b\tilde{\Sigma}_b\tilde{U}_b^{\mathrm{T}} = [\underbrace{\tilde{U}_{b1}}_{s_2}\ \underbrace{\tilde{U}_{b2}}_{m-s_2}]\begin{bmatrix}\tilde{\Sigma}_{b1} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tilde{U}_{b1}^{\mathrm{T}}\\ \tilde{U}_{b2}^{\mathrm{T}}\end{bmatrix}, \tag{26}$$

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

$$G_e = U_{w2}U_{w2}^{\mathrm{T}}\tilde{U}_{b1}. \tag{27}$$

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

2.3.1. Two-class problem

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

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

where $\rho = n_1n_2/n$ and $w = U_{w2}U_{w2}^{\mathrm{T}}(c_1-c_2) \in \mathbb{R}^{m\times 1}$. Hence the transformation matrix $g \in \mathbb{R}^{m\times 1}$ is obtained by

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

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

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

2.3.2. Relationship with LDA/GSVD

From (26), we have

$$\begin{bmatrix}\tilde{U}_{b1}^{\mathrm{T}}\\ \tilde{U}_{b2}^{\mathrm{T}}\end{bmatrix}U_{w2}U_{w2}^{\mathrm{T}}S_bU_{w2}U_{w2}^{\mathrm{T}}[\tilde{U}_{b1}\tilde{U}_{b2}] = \begin{bmatrix}\tilde{\Sigma}_{b1} & 0\\ 0 & 0\end{bmatrix}, \tag{28}$$

$$\begin{bmatrix}\tilde{U}_{b1}^{\mathrm{T}}\\ \tilde{U}_{b2}^{\mathrm{T}}\end{bmatrix}U_{w2}U_{w2}^{\mathrm{T}}S_wU_{w2}U_{w2}^{\mathrm{T}}[\tilde{U}_{b1}\tilde{U}_{b2}] = 0. \tag{29}$$

The second equation holds due to Eq. (25). Eqs. (28)–(29) imply that the column vectors of G_e given in Eq. (27) belong to null(S_w) $\cap$ 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 2 shows that the LDA/GSVD solution also includes the vectors from null(S_w) $\cap$ null(S_b)^c. Based on

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

this observation, this method *To-N*(S_w) can be compared with LDA/GSVD. By denoting X in LDA/GSVD as

$$X = [\underbrace{X_1}_{\mu}\ \underbrace{X_2}_{\tau}\ \underbrace{X_3}_{s-\mu-\tau}\ \underbrace{X_4}_{m-s}], \tag{30}$$

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

Eq. (14) 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 $[X_1\ \ X_4]$. 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 I. *Any vector x belongs to null$(S_w) \cap$ 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 $x \in$ null$(S_w) \cap$ null$(S_b)^c$. Since $[X_1\ \ X_4]$ is a basis of 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 $x = X_4k \in$ 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 if $h \neq 0$ then $x = X_1h + X_4k$ belongs to null$(S_w) \cap$ null$(S_b)^c$. Since $x = X_1h + X_4k \in$ null(S_w), it is enough to show $x \notin$ null(S_b). From Eq. (13),

$$x^{\mathrm{T}}S_bx = (X_1h)^{\mathrm{T}}S_b(X_1h) = h^{\mathrm{T}}(X_1^{\mathrm{T}}S_bX_1)h$$
$$= h^{\mathrm{T}}I_\mu h = \|h\|_2^2 \neq 0. \qquad \square$$

By Theorem I,

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

for some matrices $H \in \mathbb{R}^{\mu\times s_2}$ and $K \in \mathbb{R}^{(m-s)\times s_2}$ with $s_2 =$ 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 $G_e = U_{w2}U_{w2}^{\mathrm{T}}\tilde{U}_{b1}$ is given as

$$G_e^{\mathrm{T}}z = H^{\mathrm{T}}X_1^{\mathrm{T}}z + K^{\mathrm{T}}X_4^{\mathrm{T}}z. \tag{31}$$

As explained in Eq. (17) of Section 2.2, 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 $K^{\mathrm{T}}X_4^{\mathrm{T}}z$ in (31) corresponds to the translation which does not affect the classification performance.

While the transformation matrix $G_e = U_{w2}U_{w2}^{\mathrm{T}}\tilde{U}_{b1}$ by the method *To-N*(S_w) is related to X_1 of LDA/GSVD as in Eq. (31), the main difference between the two methods is due to the eigenvectors in null$(S_w)^c \cap$ null$(S_b)^c$, which correspond to the second row in Table 2. The projection to null(S_w) by $U_{w2}U_{w2}^{\mathrm{T}}$ excludes vectors in null$(S_w)^c$, and therefore null$(S_w)^c \cap$ null$(S_b)^c$. When

$$\mathrm{rank}(\tilde{S}_b) < \mathrm{rank}(S_b) \leqslant r - 1,$$

where r is the number of classes, the reduced dimension by $G_e = U_{w2}U_{w2}^{\mathrm{T}}\tilde{U}_{b1}$ is rank$(\tilde{S}_b)$, therefore less than $r-1$, while LDA/GSVD includes $r-1$ 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 four classes. Each class has 80 samples and the data dimension is 2412. After splitting the data set 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 data set. 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 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. Fig. 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 illustrates the reduced dimensional space. The second and third figures show that adding one more column vector from $U_{w2}U_{w2}^{\mathrm{T}}\tilde{U}_{b2}$ 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 null$(S_w) \cap$ null(S_b). On the other hand, when

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

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

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

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 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 range(S_b), and then in the transformed space the minimization of within-class scatter is pursued.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

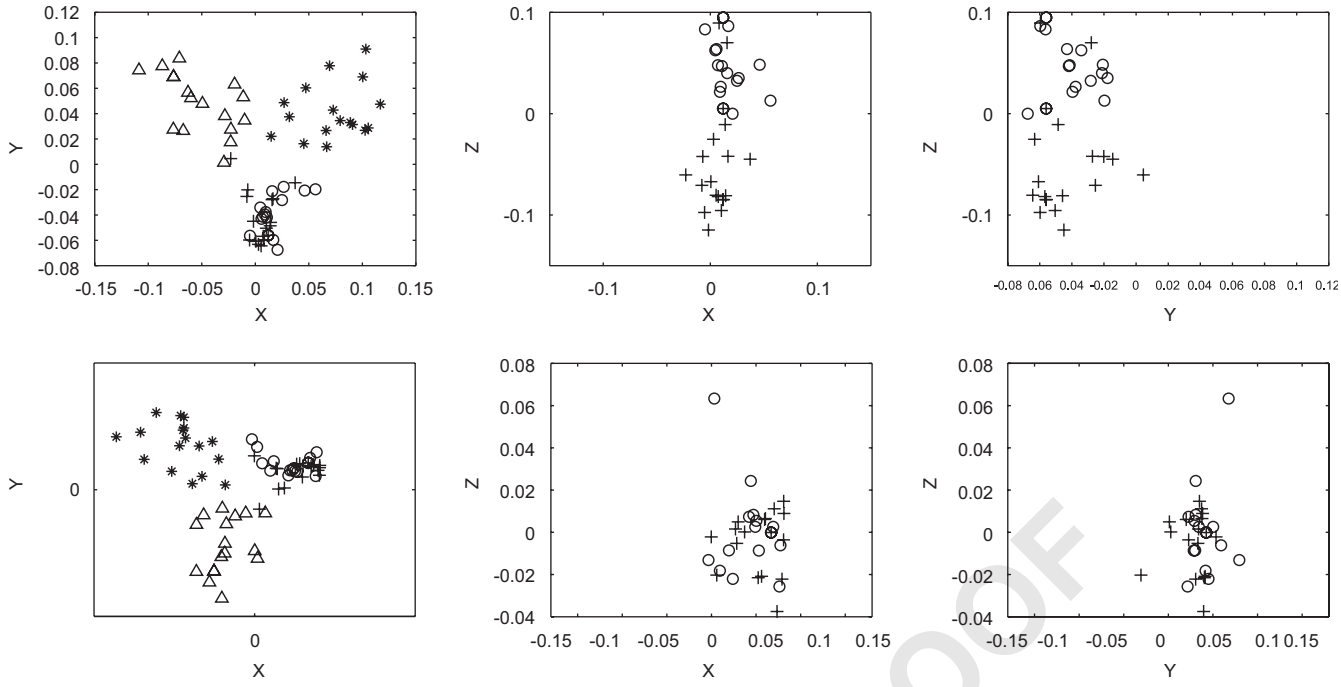

Fig. 1. The visualization of the data in the reduced dimensional spaces by LDA/GSVD (figures in the first row) and the method *To-N*(S_w) (figures in the second row).

Consider the EVD of S_b,

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

where U_b is orthogonal, $\mathrm{rank}(S_b) = s_1$ and Σ_{b1} is a diagonal matrix with nonincreasing positive diagonal components. Then $\mathrm{range}(S_b) = \mathrm{span}(U_{b1})$. 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

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

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

$$\tilde{S}_w = \tilde{U}_w \tilde{\Sigma}_w \tilde{U}_w^{\mathrm{T}}, \tag{32}$$

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

$$\tilde{U}_w^{\mathrm{T}} V_y^{\mathrm{T}} S_b V_y \tilde{U}_w = I_{s_1} \quad \text{and} \quad \tilde{U}_w^{\mathrm{T}} V_y^{\mathrm{T}} S_w V_y \tilde{U}_w = \tilde{\Sigma}_w. \tag{33}$$

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

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

Scaling (33) by $\tilde{\Sigma}_w^{-1/2}$, we have

$$(\tilde{\Sigma}_w^{-1/2} \tilde{U}_w^{\mathrm{T}} V_y^{\mathrm{T}}) S_b (V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}) = \tilde{\Sigma}_w^{-1},$$
$$(\tilde{\Sigma}_w^{-1/2} \tilde{U}_w^{\mathrm{T}} V_y^{\mathrm{T}}) S_w (V_y \tilde{U}_w \tilde{\Sigma}_w^{-1/2}) = I_{s_1}. \tag{34}$$

The authors in Ref. [6] proposed the transformation matrix

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

Eqs. (34) imply that each column of G_y belongs to $\mathrm{null}(S_w)^c \cap \mathrm{null}(S_b)^c$. We call this method *To-R*(S_b) for short.

2.4.1. Two-class problem

In a two-class problem, since

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

where $\rho = n_1 n_2 / n$, a data item is transformed to the 1-dimensional space by $g = (c_1 - c_2)/(\sqrt{\rho}\|c_1 - c_2\|_2^2)$. The dimension reduced representation of any data item z is given by $g^{\mathrm{T}} z = v(c_1 - c_2)^{\mathrm{T}} z$ for some scalar v. 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,

$$J_i(G) = J_i(GF), \quad i = 2, 3, \tag{35}$$

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

Table 3
The prediction accuracies (%)

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

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 Eqs. (32)–(34)) 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 3. After dimension reduction, 1-NN classifier was used in the reduced dimensional space.

2.5. A method of PCA plus transformations to range(S_w) and null(S_w)

As shown in the analysis of the compared methods, they search for discriminative vectors in null$(S_w)\cap$ null$(S_b)^c$ and null$(S_w)^c\cap$ null$(S_b)^c$. The method *To-N*(S_w) by Chen et al. finds solution vectors in null$(S_w)\cap$ null$(S_b)^c$ and *To-R*(S_b) by Yu et al. restricts the search space to null$(S_w)^c\cap$ null$(S_b)^c$. LDA/GSVD by Howland et al. finds solution from both spaces, however, the number of possible discriminative vectors cannot be greater than rank(S_b), possibly resulting in solution vectors only from null$(S_w)\cap$ 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 range(S_t), as in PCA, is performed. Let the SVD of S_t be

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

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

$$\tilde{S}_w = \tilde{U}_w\tilde{\Sigma}_w\tilde{U}_w^{\mathrm{T}} = [\tilde{U}_{w1}\ \tilde{U}_{w2}]\begin{bmatrix}\tilde{\Sigma}_{w1} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tilde{U}_{w1}^{\mathrm{T}}\\ \tilde{U}_{w2}^{\mathrm{T}}\end{bmatrix}. \quad (36)$$

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

$$\bar{S}_b \equiv \tilde{U}_{w2}^{\mathrm{T}}U_{t1}^{\mathrm{T}}S_bU_{t1}\tilde{U}_{w2}. \quad (37)$$

On the other hand, in the transformed space by the basis $\tilde{U}_{w1}$ of range$(\tilde{S}_w)$, let Z be the matrix whose columns are the eigenvectors[4] with the k largest nonzero eigenvalues of $\hat{S}_t^{-1}\hat{S}_b$ where $\hat{S}_b \equiv \tilde{U}_{w1}^{\mathrm{T}}U_{t1}^{\mathrm{T}}S_bU_{t1}\tilde{U}_{w1}$ and $\hat{S}_t \equiv \tilde{U}_{w1}^{\mathrm{T}}U_{t1}^{\mathrm{T}}S_tU_{t1}\tilde{U}_{w1}$. Then the transformation matrix by the method *To-NR*(S_w) is constructed as

$$G_d = [U_{t1}\tilde{U}_{w2}Y \quad U_{t1}\tilde{U}_{w1}Z]. \quad (38)$$

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.

2.5.1. 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 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 null(S_t) and null$(S_t)\subset$ null(S_w), from (36)

$$\begin{bmatrix}U_{t1}^{\mathrm{T}}\\ U_{t2}^{\mathrm{T}}\end{bmatrix}S_w\,[U_{t1}\quad U_{t2}] = \begin{bmatrix}\tilde{U}_w\tilde{\Sigma}_w\tilde{U}_w^{\mathrm{T}} & 0\\ 0 & 0\end{bmatrix}. \quad (39)$$

By Eq. (39), we can obtain the EVD of S_w as

$$S_w = [U_{t1}\tilde{U}_w\ \ U_{t2}]\begin{bmatrix}\tilde{\Sigma}_w & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tilde{U}_w^{\mathrm{T}}U_{t1}^{\mathrm{T}}\\ U_{t2}^{\mathrm{T}}\end{bmatrix}$$
$$= [U_{t1}\tilde{U}_{w1}\quad U_{t1}\tilde{U}_{w2}\quad U_{t2}]\begin{bmatrix}\tilde{\Sigma}_{w1} & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\tilde{U}_{w1}^{\mathrm{T}}U_{t1}^{\mathrm{T}}\\ \tilde{U}_{w2}^{\mathrm{T}}U_{t1}^{\mathrm{T}}\\ U_{t2}^{\mathrm{T}}\end{bmatrix}. \quad (40)$$

Eq. (40) shows that the columns of $V \equiv [U_{t1}\tilde{U}_{w2}\ \ U_{t2}]$ is an orthonormal basis of null(S_w). Hence the transformation by VV^{T} gives the projection onto the null space of S_w.

Now by notation (37) and span$(U_{t2}) = $ null$(S_t)\subset$ null(S_b),

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

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

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

[4] In Ref. [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 $\hat{S}_t^{-1}\hat{S}_b$.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

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

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

$$= U_{t1}\tilde{U}_{w2}\bar{U}_{b1}, \tag{41}$$

which is exactly same as $U_{t1}\tilde{U}_{w2}Y$ in G_d of (38).

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. In Section 2.7 where we present experimental comparison of the discussed algorithms, we demonstrate the difficulty with choosing the optimal dimension reduced by PCA.

2.6.2. GSLDA

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

$$\mathrm{null}(S_t)^{\perp} \cap \mathrm{null}(S_w), \tag{42}$$

where $\mathrm{null}(S_t)^{\perp}$ denotes the orthogonal complement of $\mathrm{null}(S_t)$ [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 $\mathrm{null}(S_t)^{\perp} \cap \mathrm{null}(S_w)$. In Ref. [14], under the assumption that the given data items are independent, MGS is applied to

$$[H_w^*, H_b^*], \tag{43}$$

obtaining an orthogonal basis Q of Eq. (43), where H_w^* is constructed by deleting one column from each subblock $A_i - c_i e_1$, $1 \leqslant i \leqslant r$, in H_w and $H_b^* = [c_1 - c, \ldots, c_{r-1} - c]$. Then the last $r-1$ columns of Q give an orthogonal basis of Eq. (42). When applying L_2-norm as a similarity measure, using any orthogonal basis of $\mathrm{null}(S_t)^{\perp} \cap \mathrm{null}(S_w)$ 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 Eq. (42), and therefore prediction accuracies by the method *To-N*(S_w) and GSLDA should be same.

2.6.3. Uncorrelated LDA

Instead of the orthogonality of the columns $\{g_i\}$ in the transformation matrix G, i.e., $g_i^{\mathrm{T}} g_j = 0$ for $i \neq j$, uncorrelated LDA (ULDA) imposes the S_t-orthogonal constraint, $g_i^{\mathrm{T}} S_t g_j = 0$ for $i \neq j$ [15]. In Ref. [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.

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 sets,[5] 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 4. 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 neighbor. Table 5 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/nonsmiling), and facial details (glasses/noglasses). 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

[5] The text data sets were downloaded 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 and preprocessed to reduce to manageable data size.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

Table 4
The description of data sets

Data	Re1	Tr12	Tr23	Tr31	Tr41	Tr45	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 5
Prediction accuracies (%)

Data	RLDA	LDA/GSVD	*To-N*(S_w)	*To-R*(S_b)	*To-NR*(S_w)
Text classification					
Re1	**95.8**	95.1	94.5	94.2	94.7
Tr12	95.7	**98.3**	98.1	96.7	97.6
Tr23	87.9	90.3	91.5	88.2	**91.8**
Tr31	98.6	98.4	98.6	97.7	**98.7**
Tr41	**98.0**	97.3	97.0	96.3	97.1
Tr45	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

For RLDA, the best accuracy among $\alpha = 0.5, 1, 1.5$ is reported. For each data set, the best prediction accuracy is shown in boldface.

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 4. 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 5 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 range(S_b), *To-R*(S_b), gives 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.

When PCA is performed as a preprocessing step for LDA, it is not easy to determine the dimension obtained by PCA. In the next experiment we compare *PCA plus LDA* with the generalized LDA methods discussed. Varying the dimensions reduced by PCA, LDA was applied to reduce the data dimension further to $r - 1$ where r is the number of classes. Three figures in Fig. 2 show prediction accuracies for three data sets Tr12, Tr31 and Yale face data, respectively. The values on the horizontal axis denote the intermediate data dimensions obtained by PCA. They demonstrate the difficulty in choosing the optimal dimension in applying PCA as a preprocessing step for LDA, although the best prediction accuracies indicated by the peak points on the graphs are comparable with those in Table 5.

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 $p \gg q$,

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

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, p. 254].

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 Eqs. (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

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

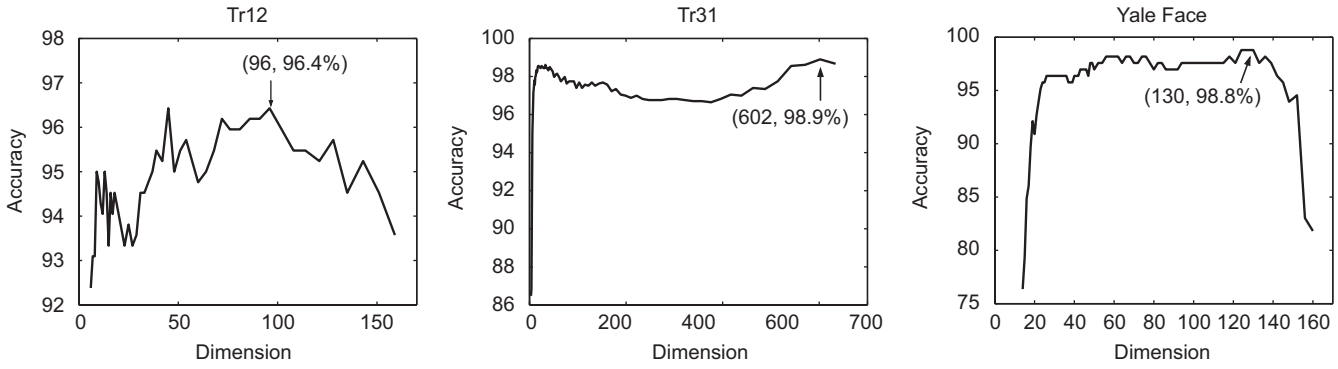

Fig. 2. The effects of the dimensions reduced by PCA on the prediction accuracies. The values on the horizontal axis denote data dimensions reduced by PCA before LDA is applied.

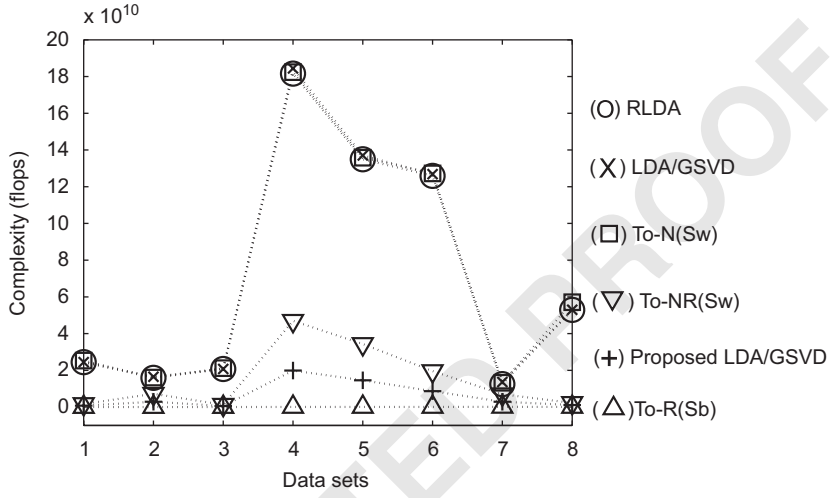

Fig. 3. Comparison of computational complexities of the generalized LDA methods using the sizes of training data used in experiments. From the left on x-axis, the data sets, Tr12, Re1, Tr23, Tr31, Tr41, Tr45, AT&T and Yale, are corresponded.

only eigenvectors corresponding to the nonzero eigenvalues of $HH^{\mathrm{T}} \in \mathbb{R}^{p\times p}$ are needed, the approach of computing the EVD of $H^{\mathrm{T}}H$ instead of HH^{T} as explained in Section 2.2 was utilized.

Fig. 3 compares computational complexities of the discussed methods by using specific sizes of training data sets used in the experiments. As shown in Fig. 3, 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 cannot be ranked highly. The proposed algorithm 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 [20–25]. 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

$$\kappa(x, y) = \langle \Phi(x), \Phi(y) \rangle, \tag{44}$$

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

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 $\Phi : \mathscr{A} \subset \mathbb{R}^m \to \mathscr{F} \subset \mathbb{R}^N$ satisfying Eq. (44). As long as the problem formulation depends only on the inner products between data points in $\mathscr{F}$ and not on the data points themselves, without explicit representation of the feature mapping Φ or the feature space $\mathscr{F}$, we can work on the feature space $\mathscr{F}$ through relation (44). As

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

positive definite kernel functions satisfying Mercer's condition, polynomial kernel and Gaussian kernel

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

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

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 Eq. (44). Then the LDA in $\mathcal{F}$ finds a linear transformation $\mathcal{G} = [\varphi_1, \ldots, \varphi_l] \in \mathbb{R}^{N\times l}$, where the columns of $\mathcal{G}$ are the generalized eigenvectors corresponding to the l largest eigenvalues of

$$\mathcal{S}_b\varphi = \lambda\mathcal{S}_w\varphi. \tag{45}$$

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

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

where

$$\mathcal{H}_b = [\sqrt{n_1}(\tilde{c}_1 - \tilde{c}), \ldots, \sqrt{n_r}(\tilde{c}_r - \tilde{c})] \in \mathbb{R}^{N\times r}, \tag{46}$$

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

$$\tilde{c}_i = \frac{1}{n_i}\sum_{j\in N_i}\Phi(a_j), \quad \tilde{c} = \frac{1}{n}\sum_{i=1}^{n}\Phi(a_i) \quad \text{and}$$

$$e_i = [1, \ldots, 1] \in \mathbb{R}^{1\times n_i}. \tag{47}$$

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

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

$$u = [u_1, \ldots, u_n]^{\mathrm{T}},$$

$$\mathcal{K}_b = [b_{ij}]_{(1\leqslant i\leqslant n, 1\leqslant j\leqslant r)},$$

$$b_{ij} = \sqrt{n_j}\left(\frac{1}{n_j}\sum_{p\in N_j}\kappa(a_i, a_p) - \frac{1}{n}\sum_{p=1}^{n}\kappa(a_i, a_p)\right). \tag{48}$$

Then we have

$$\mathcal{H}_b^{\mathrm{T}}\varphi = \mathcal{K}_b^{\mathrm{T}}u, \tag{49}$$

since

$$\mathcal{H}_b^{\mathrm{T}}\varphi = \begin{bmatrix}\sqrt{n_1}(\tilde{c}_1 - \tilde{c})^{\mathrm{T}}\\ \vdots \\ \sqrt{n_r}(\tilde{c}_r - \tilde{c})^{\mathrm{T}}\end{bmatrix}\left(\sum_{i=1}^{n}u_i\Phi(a_i)\right)$$

$$= \begin{bmatrix}\sqrt{n_1}\left(\frac{1}{n_1}\sum_{p\in N_1}\Phi(a_p) - \frac{1}{n}\sum_{p=1}^{n}\Phi(a_p)\right)^{\mathrm{T}}\\ \vdots \\ \sqrt{n_r}\left(\frac{1}{n_r}\sum_{p\in N_r}\Phi(a_p) - \frac{1}{n}\sum_{p=1}^{n}\Phi(a_p)\right)^{\mathrm{T}}\end{bmatrix}$$

$$[\Phi(a_1), \quad \cdots, \quad \Phi(a_n)]\begin{bmatrix}\alpha_1\\ \vdots \\ \alpha_n\end{bmatrix}$$

$$= \mathcal{K}_b^{\mathrm{T}}u.$$

Similarly, we can obtain

$$\mathcal{H}_w^{\mathrm{T}}\varphi = \mathcal{K}_w^{\mathrm{T}}u, \tag{50}$$

where

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

$$w_{ij} = \kappa(a_i, a_j) - \frac{1}{n_\delta}\sum_{p\in N_\delta}\kappa(a_i, a_p) \tag{51}$$

when a_j belongs to the class δ.

From Eqs. (49) and (50), 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

$$\begin{aligned}&\mathcal{S}_b\varphi = \lambda\mathcal{S}_w\varphi\\ &\Leftrightarrow \psi^{\mathrm{T}}\mathcal{H}_b\mathcal{H}_b^{\mathrm{T}}\varphi = \lambda\psi^{\mathrm{T}}\mathcal{H}_w\mathcal{H}_w^{\mathrm{T}}\varphi\\ &\Leftrightarrow v^{\mathrm{T}}\mathcal{K}_b\mathcal{K}_b^{\mathrm{T}}u = \lambda v^{\mathrm{T}}\mathcal{K}_w\mathcal{K}_w^{\mathrm{T}}u\\ &\quad \text{for } u = [u_1, \ldots, u_n]^{\mathrm{T}}, \quad v = [v_1, \ldots, v_n]^{\mathrm{T}}\\ &\Leftrightarrow \mathcal{K}_b\mathcal{K}_b^{\mathrm{T}}u = \lambda\mathcal{K}_w\mathcal{K}_w^{\mathrm{T}}u.\end{aligned} \tag{52}$$

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

$$\mathcal{K}_b\mathcal{K}_b^{\mathrm{T}}u = \lambda\mathcal{K}_w\mathcal{K}_w^{\mathrm{T}}u. \tag{53}$$

Algorithm 2. Nonlinear discriminant analysis. Given a data matrix $A=[a_1, \ldots, a_n] \in \mathbb{R}^{m\times n}$ 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 $K = [\kappa(a_i, a_j)]_{(1\leqslant i\leqslant n, 1\leqslant j\leqslant n)}$.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

Table 6
Prediction accuracies (%) by the classical LDA in the original space and the generalized LDA algorithms in the nonlinearly transformed feature space

Data	Dim.	No. data	Classes	Linear	Nonlinear methods				
				LDA	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

In the Mfeature data set, the classical LDA was not applicable in the original space due to the singularity of the within-class scatter matrix.

(1) Compute $\mathscr{K}_b \in \mathbb{R}^{n\times r}$, $\mathscr{K}_w \in \mathbb{R}^{n\times n}$ and $\mathscr{K}_t \in \mathbb{R}^{n\times n}$ according to (48), (51) and (55).
(2) Compute transformation matrix $\mathscr{G}$ 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. (57).

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

$$K = [\kappa(a_i, a_j)]_{(1\leqslant i\leqslant n, 1\leqslant j\leqslant n)}, \tag{54}$$

when each column $[\kappa(a_1, a_j), \ldots, \kappa(a_n, a_j)]^{\mathrm{T}}$ in K is considered as a data point in the n-dimensional space. It can be observed by comparing the structures of $\mathscr{K}_b$ and $\mathscr{K}_w$ with those of $\mathscr{H}_b$ and $\mathscr{H}_w$ in Eqs. (46)–(47). As in $\mathscr{K}_b$ and $\mathscr{K}_w$ of Eqs. (48) and (51), $\mathscr{K}_t$ can be computed as

$$\mathscr{K}_t = [t_{ij}]_{(1\leqslant i\leqslant n, 1\leqslant j\leqslant n)},$$

$$t_{ij} = \kappa(a_i, a_j) - \frac{1}{n}\sum_{p=1}^{n}\kappa(a_i, a_p). \tag{55}$$

Since $\mathscr{K}_b\mathscr{K}_b^{\mathrm{T}}$ and $\mathscr{K}_w\mathscr{K}_w^{\mathrm{T}}$ are both singular in the feature space, the classical LDA cannot be applied for the generalized eigenvalue problem (53). Now we apply the generalized LDA algorithms discussed in Section 2 to solve Eq. (53), obtaining nonlinear discriminant analysis. Let

$$\mathscr{G} = [u^{(1)}, \ldots, u^{(l)}] \in \mathbb{R}^{n\times l} \tag{56}$$

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

$$\mathscr{G}^{\mathrm{T}}\begin{bmatrix}\kappa(a_1, z)\\ \vdots \\ \kappa(a_n, z)\end{bmatrix} \in \mathbb{R}^{l\times 1}. \tag{57}$$

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

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, 10 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, five cross-validation was used in order to determine the optimal value for σ in the Gaussian kernel function $\kappa(x, y) = \exp(-(\|x-y\|^2)/2\sigma^2)$. 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 6. In the regularization method, while the regularization parameter was set as 1, the optimal σ value was searched by the cross-validation. Table 6 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 LDA.

Fig. 4 illustrates the computational complexities using the specific sizes of the training data used in Table 6. As in the comparison of the generalized LDA algorithms, the method *To-R*(S_b) [5] gives the lowest computational complexities among the compared methods. However, combining *To-R*(S_b) with kernel methods does not make effective nonlinear dimension reduction method as shown in Table 6. In the generalized eigenvalue problem,

$$\mathscr{K}_b\mathscr{K}_b^{\mathrm{T}}u = \lambda\mathscr{K}_w\mathscr{K}_w^{\mathrm{T}}u,$$

where $\mathscr{K}_b\mathscr{K}_b^{\mathrm{T}}, \mathscr{K}_w\mathscr{K}_w^{\mathrm{T}} \in \mathbb{R}^{n\times n}$. The data dimension is equal to the number of data and the rank of $\mathscr{K}_w\mathscr{K}_w^{\mathrm{T}}$ is not severely smaller than the data dimension. However, poor performances by *To-R*(S_b) demonstrate that the null space of $\mathscr{K}_w\mathscr{K}_w^{\mathrm{T}}$ contains discriminative information. Figs. 3 and 4 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.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

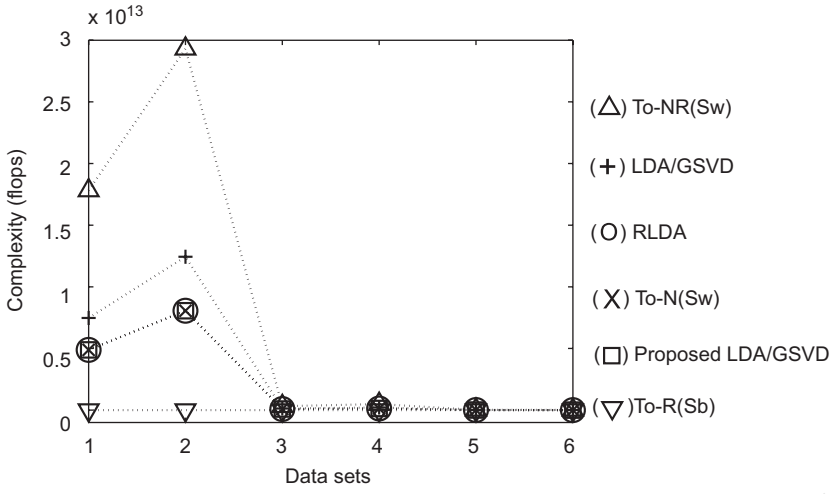

Fig. 4. The figures compare complexities required for the generalized LDA algorithms in the feature space for specific problem sizes of training data used in Table 6. From the left on x-axis, the data sets, Musk, Isolet, Car, Mfeature, Bcancer and Bscale are corresponded.

4. Conclusions/Discussions

We presented the relationships among several generalized LDA 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 complexities 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 LDA.

References

[1] K. Fukunaga, Introduction to Statistical Pattern Recognition, second ed., Academic Press, New York, 1990.
[2] R.O. Duda, P.E. Hart, D.G. Stork, Pattern Classification, Wiley-Interscience, New York, 2001.
[3] J.H. Friedman, Regularized discriminant analysis, J. Am. Stat. Assoc. 84 (405) (1989) 165–175.
[4] P. Howland, M. Jeon, H. Park, Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition, SIAM J. Matrix Anal. Appl. 25 (1) (2003) 165–179.
[5] L. Chen, H.M. Liao, M. Ko, J. Lin, G. Yu, A new LDA-based face recognition system which can solve the small sample size problem, Pattern Recognition 33 (2000) 1713–1726.
[6] H. Yu, J. Yang, A direct LDA algorithm for high-dimensional data—with application to face recognition, Pattern Recognition 34 (2001) 2067–2070.
[7] J. Yang, J.-Y. Yang, Why can LDA be performed in PCA transformed space?, Pattern Recognition 36 (2003) 563–566.
[8] W. Zheng, L. Zhao, C. Zou, An efficient algorithm to solve the small sample size problem for LDA, Pattern Recognition 37 (2004) 1077–1079.
[9] N. Cristianini, J. Shawe-Taylor, An Introduction to Support Vector Machines and other Kernel-based Learning Methods, Cambridge, 2000.
[10] P. Howland, H. Park, Generalizing discriminant analysis using the generalized singular value decomposition, IEEE Trans. Pattern Anal. Mach. Intell. 26 (8) (2004) 995–1006.
[11] C.C. Paige, M.A. Saunders, Towards a generalized singular value decomposition, SIAM J. Numer. Anal. 18 (1981) 398–405.
[12] G.H. Golub, C.F. Van Loan, Matrix Computations, third ed., Johns Hopkins University Press, Baltimore, MD, 1996.
[13] P.N. Belhumeur, J.P. Hespanha, D.J. Kriegman, Eigenfaces vs fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Learn. 19 (7) (1997) 711–720.
[14] W. Zheng, C. Zou, 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, Z. Lou, Face recognition based on the uncorrelated discriminant transformation, Pattern Recognition 34 (2001) 1405–1416.
[16] J. Ye, R. Janardan, Q. Li, 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, D.P. O'Leary, A semidiscrete matrix decomposition for latent semantic indexing in information retrieval, ACM Trans. Inf. Systems 16 (4) (1998) 322–346.
[18] ⟨http://www.uk.research.att.com/facedatabase.html⟩.
[19] ⟨http://cvc.yale.edu/projects/yalefaces/yalefaces.html⟩.
[20] B. Scholkopf, A.J. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Comput. 10 (1998) 1299–1319.
[21] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K.-R. Muller, Fisher discriminant analysis with kernels, in: E. Wilson, J. Larsen, S. Douglas (Eds.), Neural Networks for Signal Processing, vol. IX, IEEE, New York, 1999, p. 41–48.
[22] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Comput. 12 (2000) 2385–2404.
[23] V. Roth, V. Steinhage, Nonlinear discriminant analysis using kernel functions, Adv. Neural Inf. Process. Systems 12 (2000) 568–574.
[24] S.A. Billings, K.L. Lee, Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm, Neural Networks 15 (2) (2002) 263–270.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022

[25] C.H. Park, H. Park, Nonlinear discriminant analysis using kernel functions and the generalized singular value decomposition, SIAM J. Matrix Anal. Appl. 27 (1) (2005) 87–102.

[26] B. Scholkopf, S. Mika, C.J.C. Burges, P. Knirsch, K.-R. Muller, G. Ratsch, A.J. Smola, Input space versus feature space in kernel-based methods, IEEE Trans. Neural Networks 10 (5) (1999) 1000–1017.

[27] C.J.C. Burges, A tutorial on support vector machines for pattern recognition, Data Min. Knowl. Discovery 2 (2) (1998) 121–167.

About the Author—CHEONG HEE PARK received her Ph.D. in Mathematics from Yonsei University, Korea in 1998. She received the M.S. and Ph.D. degrees in Computer Science at the Department of Computer Science and Engineering, University of Minnesota in 2002 and 2004, respectively. She is currently in the Department of Computer Science and Engineering, Chungnam National University, Korea as an Assistant Professor. Her research interests include pattern recognition, data mining, bioinformatics and machine learning.

About the Author—HAESUN PARK received her B.S. degree in Mathematics from Seoul National University, Seoul Korea, in 1981 with summa cum laude and the University President's Medal for the top graduate, and her M.S. and Ph.D. degrees in Computer Science from Cornell University, Ithaca, NY, in 1985 and 1987, respectively. She was on the faculty of the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, from 1987 to 2005. From 2003 to 2005, she served as a Program Director, the Computing and Communication Foundations Division at the National Science Foundation, Arlington, VA, USA. Since July 2005, she has been a Professor in Division of Computational Science and Engineering, College of Computing, Georgia Institute of Technology, Atlanta, GA. Her current research interests include numerical algorithms, pattern recognition, bioinformatics, information retrieval, and data mining. She has published over 100 research papers in these areas. Prof. Park served on numerous conference committees and editorial boards of journals. Currently she is on the editorial board of BIT Numerical Mathematics, SIAM Journal on Matrix Analysis and Applications, and International Journal of Bioinformatics Research and Applications.

Please cite this article as: C.H. Park, H. Park, A comparison of generalized linear discriminant analysis algorithms, Pattern Recognition (2007), doi: 10.1016/j.patcog.2007.07.022