Journal of Machine Learning Research 7 (2006) 1183–1204 Submitted 12/05; Revised 3/06; Published 7/06

Computational and Theoretical Analysis of Null Space and Orthogonal Linear Discriminant Analysis

Jieping Ye JIEPING.YE@ASU.EDU
Department of Computer Science and Engineering
Arizona State University
Tempe, AZ 85287, USA

Tao Xiong TXIONG@ECE.UMN.EDU
Department of Electrical and Computer Engineering
University of Minnesota
Minneapolis, MN 55455, USA

Editor: David Madigan

Abstract

Dimensionality reduction is an important pre-processing step in many applications. Linear discriminant analysis (LDA) is a classical statistical approach for supervised dimensionality reduction. It aims to maximize the ratio of the between-class distance to the within-class distance, thus maximizing the class discrimination. It has been used widely in many applications. However, the classical LDA formulation requires the nonsingularity of the scatter matrices involved. For undersampled problems, where the data dimensionality is much larger than the sample size, all scatter matrices are singular and classical LDA fails. Many extensions, including null space LDA (NLDA) and orthogonal LDA (OLDA), have been proposed in the past to overcome this problem. NLDA aims to maximize the between-class distance in the null space of the within-class scatter matrix, while OLDA computes a set of orthogonal discriminant vectors via the simultaneous diagonalization of the scatter matrices. They have been applied successfully in various applications.

In this paper, we present a computational and theoretical analysis of NLDA and OLDA. Our main result shows that under a mild condition which holds in many applications involving high-dimensional data, NLDA is equivalent to OLDA. We have performed extensive experiments on various types of data and results are consistent with our theoretical analysis. We further apply the regularization to OLDA. The algorithm is called regularized OLDA (or ROLDA for short). An efficient algorithm is presented to estimate the regularization value in ROLDA. A comparative study on classification shows that ROLDA is very competitive with OLDA. This confirms the effectiveness of the regularization in ROLDA.

Keywords: linear discriminant analysis, dimensionality reduction, null space, orthogonal matrix, regularization

1. Introduction

Dimensionality reduction is important in many applications of data mining, machine learning, and bioinformatics, due to the so-called *curse of dimensionality* (Bellmanna, 1961; Duda et al., 2000; Fukunaga, 1990; Hastie et al., 2001). Many methods have been proposed for dimensionality reduction, such as principal component analysis (PCA) (Jolliffe, 1986) and linear discriminant analysis

(LDA) (Fukunaga, 1990). LDA aims to find the optimal discriminant vectors (transformation) by maximizing the ratio of the between-class distance to the within-class distance, thus achieving the maximum class discrimination. It has been applied successfully in many applications including information retrieval (Berry et al., 1995; Deerwester et al., 1990), face recognition (Belhumeour et al., 1997; Swets and Weng, 1996; Turk and Pentland, 1991), and microarray gene expression data analysis (Dudoit et al., 2002). However, classical LDA requires the so-called *total scatter matrix* to be nonsingular. In many applications such as those mentioned above, all scatter matrices in question can be singular since the data points are from a very high-dimensional space and in general the sample size does not exceed this dimensionality. This is known as the *singularity* or *undersampled problem* (Krzanowski et al., 1995).

In recent years, many approaches have been proposed to deal with such high-dimensional, undersampled problem, including null space LDA (NLDA) (Chen et al., 2000; Huang et al., 2002), orthogonal LDA (OLDA) (Ye, 2005), uncorrelated LDA (ULDA) (Ye et al., 2004a; Ye, 2005), subspace LDA (Belhumeour et al., 1997; Swets and Weng, 1996), regularized LDA (Friedman, 1989), and pseudo-inverse LDA (Raudys and Duin, 1998; Skurichina and Duin, 1996). Null space LDA computes the discriminant vectors in the null space of the within-class scatter matrix. Uncorrelated LDA and orthogonal LDA are among a family of algorithms for generalized discriminant analysis proposed in (Ye, 2005). The features in ULDA are uncorrelated, while the discriminant vectors in OLDA are orthogonal to each other. Subspace LDA (or PCA+LDA) applies an intermediate dimensionality reduction stage such as PCA to reduce the dimensionality of the original data before classical LDA is applied. Regularized LDA uses a scaled multiple of the identity matrix to make the scatter matrix nonsingular. Pseudo-inverse LDA employs the pseudo-inverse to overcome the singularity problem. More details on these methods, as well as their relationship, can be found in (Ye, 2005). In this paper, we present a detailed computational and theoretical analysis of null space LDA and orthogonal LDA.

In (Chen et al., 2000), the null space LDA (NLDA) was proposed, where the between-class distance is maximized in the null space of the within-class scatter matrix. The singularity problem is thus implicitly avoided. Similar idea has been mentioned briefly in (Belhumeour et al., 1997). (Huang et al., 2002) improved the efficiency of the algorithm by first removing the null space of the total scatter matrix, based on the observation that the null space of the total scatter matrix is the intersection of the null space of the between-class scatter matrix and the null space of the within-class scatter matrix.

In orthogonal LDA (OLDA), a set of orthogonal discriminant vectors is computed, based on a generalized optimization criterion (Ye, 2005). The optimal transformation is computed through the simultaneous diagonalization of the scatter matrices, while the singularity problem is overcome implicitly. Discriminant analysis with orthogonal transformations has been studied in (Duchene and Leclerq, 1988; Foley and Sammon, 1975). By a close examination of the computations involved in OLDA, we can decompose the OLDA algorithm into three steps: first remove the null space of the total scatter matrix; followed by classical uncorrelated LDA (ULDA), a variant of classical LDA (details can be found in Section 2.1); and finally apply an orthogonalization step to the transformation.

Both the NLDA algorithm (Huang et al., 2002) and the OLDA algorithm (Ye, 2005) result in orthogonal transformations. However, they applied different schemes in deriving the optimal transformations. NLDA computes an orthogonal transformation in the null space of the within-class scatter matrix, while OLDA computes an orthogonal transformation through the simultaneous diagonaliza-

tion of the scatter matrices. Interestingly, we show in Section 5 that NLDA is equivalent to OLDA, under a mild condition C1,[1] which holds in many applications involving high-dimensional data (see Section 7). Based on the equivalence result, an improved algorithm for NLDA, called iNLDA, is presented, which further reduces the computational cost of the original NLDA algorithm.

We extend the OLDA algorithm by applying the regularization technique, which is commonly used to stabilize the sample covariance matrix estimation and improve the classification performance (Friedman, 1989). The algorithm is called regularized OLDA (or ROLDA for short). The key idea in ROLDA is to add a constant λ to the diagonal elements of the total scatter matrix. Here $\lambda > 0$ is known as the *regularization parameter*. Choosing an appropriate regularization value is a critical issue in ROLDA, as a large λ may significantly disturb the information in the scatter matrix, while a small λ may not be effective in improving the classification performance. Cross-validation is commonly used to estimate the optimal λ from a finite set of candidates. Selecting an optimal value for a parameter such as λ is called *model selection* (Hastie et al., 2001). The computational cost of model selection for ROLDA can be expensive, especially when the candidate set is large, since it requires expensive matrix computations for each λ. We show in Section 6 that the computations in ROLDA can be decomposed into two components: the first component involves matrices of high dimensionality but independent of λ, while the second component involves matrices of low dimensionality. When searching for the optimal λ from a set of candidates via cross-validation, we repeat the computations involved in the second component only, thus reducing the computational cost of model selection in ROLDA.

We have conducted experiments using 14 data sets from various data sources, including low-dimensional data from UCI Machine Learning Repository[2] and high-dimensional data such as text documents, face images, and gene expression data. (Details on these data sets can be found in Section 7.) We did a comparative study of NLDA, iNLDA, OLDA, ULDA, ROLDA, and Support Vector Machines (SVM) (Schökopf and Smola, 2002; Vapnik, 1998) in classification. Experimental results show that

- For all low-dimensional data sets, the null space of the within-class scatter matrix is empty, and both NLDA and iNLDA do not apply. However, OLDA is applicable and the reduced dimensionality of OLDA is in general $k-1$, where k is the number of classes. Condition C1 holds for most high-dimensional data sets (eight out of nine data sets). NLDA, iNLDA, and OLDA achieve the same classification performance, in all cases when condition C1 holds. For cases where condition C1 does not hold, OLDA outperforms NLDA and iNLDA, as OLDA has a larger number of reduced dimensions than NLDA and iNLDA. These empirical results are consistent with our theoretical analysis.

- iNLDA and NLDA achieve similar performance in all cases. OLDA is very competitive with ULDA. This confirms the effectiveness of the final orthogonalization step in OLDA. ROLDA achieves a better classification performance than OLDA, which shows the effectiveness of the regularization in ROLDA. Overall, ROLDA and SVM are very competitive with other methods in classification.

The rest of the paper is organized as follows. An overview of classical LDA and classical uncorrelated LDA is given in Section 2. NLDA and OLDA are discussed in Section 3 and Section 4,

1. Condition C1 requires that the rank of the total scatter matrix equals to the sum of the rank of the between-class scatter matrix and the rank of the within-class scatter matrix. More details will be given in Section 5.
2. http://www.ics.uci.edu/∼mlearn/MLRepository.html

Notation	Description	Notation	Description
A	data matrix	n	number of training data points
m	data dimensionality	ℓ	reduced dimensionality
k	number of classes	S_b	between-class scatter matrix
S_w	within-class scatter matrix	S_t	total scatter matrix
G	transformation matrix	S_i	covariance matrix of the i-th class
c_i	centroid of the i-th class	n_i	sample size of the i-th class
c	global centroid	K	number of neighbors in K-NN
t	rank of S_t	q	rank of S_b

Table 1: Notation.

respectively. The relationship between NLDA and OLDA is studied in Section 5. The ROLDA algorithm is presented in Section 6. Section 7 includes the experimental results. We conclude in Section 8.

For convenience, Table 1 lists the important notation used in the rest of this paper.

2. Classical Linear Discriminant Analysis

Given a data set consisting of n data points $\{a_j\}_{j=1}^n$ in $\mathbb{R}^m$, classical LDA computes a linear transformation $G \in \mathbb{R}^{m\times\ell}$ ($\ell < m$) that maps each a_j in the m-dimensional space to a vector $\hat{a}_j$ in the ℓ-dimensional space by $\hat{a}_j = G^T a_j$. Define three matrices H_w, H_b, and S_t as follows:

$$H_w = \frac{1}{\sqrt{n}}[(A_1 - c_1 e^T), \cdots, (A_k - c_k e^T)], \quad (1)$$

$$H_b = \frac{1}{\sqrt{n}}[\sqrt{n_1}(c_1 - c), \cdots, \sqrt{n_k}(c_k - c)], \quad (2)$$

$$H_t = \frac{1}{\sqrt{n}}(A - ce^T), \quad (3)$$

where $A = [a_1, \cdots, a_n]$ is the data matrix, A_i, c_i, S_i, and n_i are the data matrix, the centroid, the covariance matrix, and the sample size of the i-th class, respectively, c is the global centroid, k is the number of classes, and e is the vector of all ones. Then the *between-class scatter matrix* S_b, the *within-class scatter matrix* S_w, and the *total scatter matrix* S_t are defined as follows (Fukunaga, 1990):

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

It follows from the definition (Ye, 2005) that trace(S_w) measures the within-class cohesion, trace(S_b) measures the between-class separation, and trace(S_t) measures the variance of the data set, where the trace of a square matrix is the summation of its diagonal entries (Golub and Van Loan, 1996). It is easy to verify that $S_t = S_b + S_w$. In the lower-dimensional space resulting from the linear transformation G, the scatter matrices become $S_w^L = G^T S_w G$, $S_b^L = G^T S_b G$, and $S_t^L = G^T S_t G$. An optimal transformation G would maximize trace(S_b^L) and minimize trace(S_w^L). Classical LDA

aims to compute the optimal G by solving the following optimization problem:

$$G = \arg \max_{G \in \mathbb{R}^{m \times \ell} : G^T S_w G = I_\ell} \text{trace} \left(\left(G^T S_w G \right)^{-1} G^T S_b G \right). \tag{4}$$

Other optimization criteria, including those based on the determinant could also be used instead (Duda et al., 2000; Fukunaga, 1990). The solution to the optimization problem in Eq. (4) is given by the eigenvectors of $S_w^{-1} S_b$ corresponding to the nonzero eigenvalues, provided that the within-class scatter matrix S_w is nonsingular (Fukunaga, 1990). The columns of G form the discriminant vectors of classical LDA. Since the rank of the between-class scatter matrix is bounded from above by $k-1$, there are at most $k-1$ discriminant vectors in classical LDA. Note that classical LDA does not handle singular scatter matrices, which limits its applicability to low-dimensional data. Several methods, including null space LDA and orthogonal LDA subspace LDA, were proposed in the past to deal with such singularity problem as discussed in Section 1.

2.1 Classical Uncorrelated LDA

Classical uncorrelated LDA (cULDA) is an extension of classical LDA. A key property of cULDA is that the features in the transformed space are uncorrelated, thus reducing the redundancy in the transformed space.

cULDA aims to find the optimal discriminant vectors that are S_t-orthogonal.[3] Specifically, suppose r vectors $\phi_1, \phi_2, \cdots, \phi_r$ are obtained, then the $(r+1)$-th vector ϕ_{r+1} is the one that maximizes the Fisher criterion function (Jin et al., 2001):

$$f(\phi) = \frac{\phi^T S_b \phi}{\phi^T S_w \phi}, \tag{5}$$

subject to the constraints: $\phi_{r+1}^T S_t \phi_i = 0$, for $i = 1, \cdots, r$.

The algorithm in (Jin et al., 2001) finds the discriminant vectors ϕ_i's successively by solving a sequence of generalized eigenvalue problems, which is expensive for large and high-dimensional data sets. However, it has been shown (Ye et al., 2004a) that the discriminant vectors of cULDA can be computed efficiently by solving the following optimization problem:

$$G = \arg \max_{G \in \mathbb{R}^{m \times \ell} : G^T S_t G = I_\ell} \text{trace} \left(\left(G^T S_w G \right)^{-1} G^T S_b G \right), \tag{6}$$

where $G = [\phi_1, \cdots, \phi_\ell]$, if there exist ℓ discriminant vectors in cULDA. Note that in Eq. (6), all discriminant vectors in G are computed simultaneously. The optimization problem above is a variant of the one in Eq. (4). The optimal G is given by the eigenvectors of $S_t^{-1} S_b$.

3. Null Space LDA

(Chen et al., 2000) proposed the null space LDA (NLDA) for dimensionality reduction, where the between-class distance is maximized in the null space of the within-class scatter matrix. The basic idea behind this algorithm is that the null space of S_w may contain significant discriminant information if the projection of S_b is not zero in that direction (Chen et al., 2000; Lu et al., 2003).

3. Two vectors x and y are S_t-orthogonal, if $x^T S_t y = 0$.

The singularity problem is thus overcome implicitly. The optimal transformation of NLDA can be computed by solving the following optimization problem:

$$G = \text{argmax}_{G^T S_w G = 0} \text{trace}(G^T S_b G). \tag{7}$$

The computation of the optimal G involves the computation of the null space of S_w, which may be large for high-dimensional data. Indeed, the dimensionality of the null space of S_w is at least $m+k-n$, where m is the data dimensionality, k is the number of classes, and n is the sample size. In (Chen et al., 2000), a pixel grouping method was used to extract geometric features and reduce the dimensionality of samples, and then NLDA was applied in the new feature space. (Huang et al., 2002) improved the efficiency of the algorithm in (Chen et al., 2000) by first removing the null space of the total scatter matrix S_t. It is based on the observation that the null space of S_t is the intersection of the null space of S_b and the null space of S_w, as $S_t = S_w + S_b$.

We can efficiently remove the null space of S_t as follows. Let $H_t = U\Sigma V^T$ be the Singular Value Decomposition (SVD) (Golub and Van Loan, 1996) of H_t, where H_t is defined in Eq. (3), U and V are orthogonal,

$$\Sigma = \begin{pmatrix} \Sigma_t & 0 \\ 0 & 0 \end{pmatrix},$$

$\Sigma_t \in \mathbb{R}^{t \times t}$ is diagonal with the diagonal entries sorted in the non-increasing order, and $t = \text{rank}(S_t)$. Then

$$S_t = H_t H_t^T = U\Sigma V^T V \Sigma^T U^T = U\Sigma\Sigma^T U^T = U \begin{pmatrix} \Sigma_t^2 & 0 \\ 0 & 0 \end{pmatrix} U^T. \tag{8}$$

Let $U = (U_1, U_2)$ be a partition of U with $U_1 \in \mathbb{R}^{m \times t}$ and $U_2 \in \mathbb{R}^{m \times (m-t)}$. Then the null space of S_t can be removed by projecting the data onto the subspace spanned by the columns of U_1. Let $\tilde{S}_b$, $\tilde{S}_w$, and $\tilde{S}_t$ be the scatter matrices after the removal of the null space of S_t. That is,

$$\tilde{S}_b = U_1^T S_b U_1, \ \tilde{S}_w = U_1^T S_w U_1, \text{ and } \tilde{S}_t = U_1^T S_t U_1.$$

Note that only U_1 is involved for the projection. We can thus apply the reduced SVD computation (Golub and Van Loan, 1996) on H_t with the time complexity of $O(mn^2)$, instead of $O(m^2 n)$. When the data dimensionality m is much larger than the sample size n, this leads to a big reduction in terms of the computational cost.

With the computed U_1, the optimal transformation of NLDA is given by $G = U_1 N$, where N is obtained by solving the following optimization problem:

$$N = \text{argmax}_{N^T \tilde{S}_w N = 0} \text{trace}(N^T \tilde{S}_b N). \tag{9}$$

That is, the columns of N lie in the null space of $\tilde{S}_w$, while maximizing $\text{trace}(N^T \tilde{S}_b N)$.

Let W be the matrix so that the columns of W span the null space of $\tilde{S}_w$. Then $N = WM$, for some matrix M, which is to be determined next. Since the constraint in Eq. (9) is satisfied with $N = WM$ for any M, the optimal M can be computed by maximizing

$$\text{trace}(M^T W^T \tilde{S}_b W M).$$

By imposing the orthogonality constraint on M (Huang et al., 2002), the optimal M is given by the eigenvectors of $W^T \tilde{S}_b W$ corresponding to the nonzero eigenvalues. With the computed U_1, W, and M above, the optimal transformation of NLDA is given by

$$G = U_1 W M.$$

Algorithm 1: NLDA (Null space LDA)

Input: data matrix A

Output: transformation matrix G

1. Form the matrix H_t as in Eq. (3);
2. Compute the reduced SVD of H_t as $H_t = U_1 \Sigma_t V_1^T$;
3. Form the matrices $\tilde{S}_b = U_1^T S_b U_1$ and $\tilde{S}_w = U_1^T S_w U_1$;
4. Compute the null space, W, of $\tilde{S}_w$, via the eigen-decomposition;
5. Construct the matrix M, consisting of the top eigenvectors of $W^T \tilde{S}_b W$;
6. $G \leftarrow U_1 W M$.

In (Huang et al., 2002), the matrix W is computed via the eigen-decomposition of $\tilde{S}_w$. More specifically, let

$$\tilde{S}_w = [W, \tilde{W}] \begin{pmatrix} 0 & 0 \\ 0 & \Delta_w \end{pmatrix} [W, \tilde{W}]^T$$

be its eigen-decomposition, where $[W, \tilde{W}]$ is orthogonal and Δ_w is diagonal with positive diagonal entries. Then W forms the null space of $\tilde{S}_w$. The pseudo-code for the NLDA algorithm is given in **Algorithm 1**.

4. Orthogonal LDA

Orthogonal LDA (OLDA) was proposed in (Ye, 2005) as an extension of classical LDA. The discriminant vectors in OLDA are orthogonal to each other. Furthermore, OLDA is applicable even when all scatter matrices are singular, thus overcoming the singularity problem. It has been applied successfully in many applications, including document classification, face recognition, and gene expression data classification. The optimal transformation in OLDA can be computed by solving the following optimization problem:

$$G = \text{argmax}_{G \in \mathbb{R}^{m \times \ell} : G^T G = I_\ell} \text{trace}\left((G^T S_t G)^+ G^T S_b G \right), \tag{10}$$

where M^+ denotes the pseudo-inverse of matrix M (Golub and Van Loan, 1996). The orthogonality condition is imposed in the constraint. The computation of the optimal transformation of OLDA is based on the simultaneous diagonalization of the three scatter matrices as follows (Ye, 2005).

From Eq. (8), U_2 lies in the null space of both S_b and S_w. Thus,

$$U^T S_b U = \begin{pmatrix} U_1^T S_b U_1 & 0 \\ 0 & 0 \end{pmatrix}, \quad U^T S_w U = \begin{pmatrix} U_1^T S_w U_1 & 0 \\ 0 & 0 \end{pmatrix}. \tag{11}$$

Denote $B = \Sigma_t^{-1} U_1^T H_b$ and let $B = P \tilde{\Sigma} Q^T$ be the SVD of B, where P and Q are orthogonal and $\tilde{\Sigma}$ is diagonal. Define the matrix X as

$$X = U \begin{pmatrix} \Sigma_t^{-1} P & 0 \\ 0 & I_{m-t} \end{pmatrix}. \tag{12}$$

It can be shown (Ye, 2005) that X simultaneously diagonalizes S_b, S_w, and S_t. That is

$$X^T S_b X = D_b, \; X^T S_w X = D_w, \text{ and } X^T S_t X = D_t, \tag{13}$$

Algorithm 2: OLDA (Orthogonal LDA)
Input: data matrix A
Output: transformation matrix G
1. Compute U_1, Σ_t, and P;
2. $X_q \leftarrow U_1\Sigma_t^{-1}P_q$, where $q = \text{rank}(S_b)$;
3. Compute the QR decomposition of X_q as $X_q = QR$;
4. $G \leftarrow Q$.

where D_b, D_w, and D_t are diagonal with the diagonal entries in D_b sorted in the non-increasing order. The main result in (Ye, 2005) has shown that the optimal transformation of OLDA can be computed through the orthogonalization of the columns in X, as summarized in the following theorem:

Theorem 4.1 *Let X be the matrix defined in Eq. (12) and let X_q be the matrix consisting of the first q columns of X, where $q = rank(S_b)$. Let $X_q = QR$ be the QR-decomposition of X_q, where Q has orthonormal columns and R is upper triangular. Then $G = Q$ solves the optimization problem in Eq. (10).*

From Theorem 4.1, only the first q columns of X are used in computing the optimal G. From Eq. (12), the first q columns of X are given by

$$X_q = U_1\Sigma_t^{-1}P_q, \tag{14}$$

where P_q consists of the first q columns of the matrix P. We can observe that U_1 corresponds to the removal of the null space of S_t as in NLDA, while $\Sigma_t^{-1}P_q$ is the optimal transformation when classical ULDA is applied to the intermediate (dimensionality) reduced space by the projection of U_1. The OLDA algorithm can thus be decomposed into three steps: (1) Remove the null space of S_t; (2) Apply classical ULDA as an intermediate step, since the reduced total scatter is nonsingular; and (3) Apply an orthogonalization step to the transformation, which corresponds to the QR decomposition of X_q in Theorem 4.1. The pseudo-code for the OLDA algorithm is given in **Algorithm 2**.

Remark 1 *The ULDA algorithm in (Ye et al., 2004a; Ye, 2005) consists of steps 1 and 2 above, without the final orthogonalization step. Experimental results in Section 7 show that OLDA is competitive with ULDA. The rationale behind this may be that ULDA involves the minimum redundancy in the transformed space and is susceptible to overfitting; OLDA, on the other hand, removes the R matrix through the QR decomposition in the final orthogonalization step, which introduces the redundancy in the reduced space, but may be less susceptible to overfitting.*

5. Relationship Between NLDA and OLDA

Both the NLDA algorithm and the OLDA algorithm result in orthogonal transformations. Our empirical results show that they often lead to similar performance, especially for high-dimensional data. This implies there may exist an intrinsic relationship between these two algorithms. In this section, we take a closer look at the relationship between NLDA and OLDA. More specifically, we show that NLDA is equivalent to OLDA, under a mild condition

$$C1 : \text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w), \tag{15}$$

which holds in many applications involving high-dimensional data (see Section 7). It is easy to verify from the definition of the scatter matrices that $\text{rank}(S_t) \leq \text{rank}(S_b) + \text{rank}(S_w)$.

From Eqs. (8) and (11), the null space, U_2, of S_t can be removed, as follows:

$$\tilde{S}_t = U_1^T S_t U_1 = U_1^T S_b U_1 + U_1^T S_w U_1 = \tilde{S}_w + \tilde{S}_b \in \mathbb{R}^{t\times t}.$$

Since the null space of S_t is the intersection of the null space of S_b and the null space of S_w, the following equalities hold:

$$\text{rank}(\tilde{S}_t) = \text{rank}(S_t) = t, \quad \text{rank}(\tilde{S}_b) = \text{rank}(S_b), \text{ and } \text{rank}(\tilde{S}_w) = \text{rank}(S_w).$$

Thus condition C1 is equivalent to

$$\text{rank}(\tilde{S}_t) = \text{rank}(\tilde{S}_b) + \text{rank}(\tilde{S}_w).$$

The null space of $\tilde{S}_b$ and the null space of $\tilde{S}_w$ are critical in our analysis. The relationship between these two null spaces is studied in the following lemma.

Lemma 5.1 *Let $\tilde{S}_t$, $\tilde{S}_b$, and $\tilde{S}_w$ be defined as above and $t = \text{rank}(\tilde{S}_t)$. Let $\{w_1, \cdots, w_r\}$ forms an orthonormal basis for the null space of $\tilde{S}_w$, and let $\{b_1, \cdots, b_s\}$ forms an orthonormal basis for the null space of $\tilde{S}_b$. Then, $\{w_1, \cdots, w_r, b_1, \cdots, b_s\}$ are linearly independent.*

Proof Prove by contradiction. Assume there exist α_i's and β_j's, not all zeros, such that

$$\sum_{i=1}^{r} \alpha_i w_i + \sum_{j=1}^{s} \beta_j b_j = 0.$$

It follows that

$$0 = \left(\sum_{i=1}^{r} \alpha_i w_i + \sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_w \left(\sum_{i=1}^{r} \alpha_i w_i + \sum_{j=1}^{s} \beta_j b_j\right) = \left(\sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_w \left(\sum_{j=1}^{s} \beta_j b_j\right),$$

since w_i's lie in the null space of $\tilde{S}_w$. Hence,

$$\begin{aligned}\left(\sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_t \left(\sum_{j=1}^{s} \beta_j b_j\right) &= \left(\sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_w \left(\sum_{j=1}^{s} \beta_j b_j\right) + \left(\sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_b \left(\sum_{j=1}^{s} \beta_j b_j\right) \\ &= 0.\end{aligned}$$

Since $\tilde{S}_t$ is nonsingular, we have $\sum_{j=1}^{s} \beta_j b_j = 0$. Thus $\beta_j = 0$, for all j, since $\{b_1, \cdots, b_s\}$ forms an orthonormal basis for the null space of $\tilde{S}_b$.

Similarly, we have

$$0 = \left(\sum_{i=1}^{r} \alpha_i w_i + \sum_{j=1}^{s} \beta_j b_j\right)^T \tilde{S}_b \left(\sum_{i=1}^{r} \alpha_i w_i + \sum_{j=1}^{s} \beta_j b_j\right) = \left(\sum_{i=1}^{r} \alpha_i w_i\right)^T \tilde{S}_b \left(\sum_{i=1}^{r} \alpha_i w_i\right).$$

and

$$\begin{aligned}\left(\sum_{i=1}^{r} \alpha_i w_i\right)^T \tilde{S}_t \left(\sum_{i=1}^{r} \alpha_i w_i\right) &= \left(\sum_{i=1}^{r} \alpha_i w_i\right)^T \tilde{S}_w \left(\sum_{i=1}^{r} \alpha_i w_i\right) + \left(\sum_{i=1}^{r} \alpha_i w_i\right)^T \tilde{S}_b \left(\sum_{i=1}^{r} \alpha_i w_i\right) \\ &= 0.\end{aligned}$$

Hence $\sum_{i=1}^{r} \alpha_i w_i = 0$, and $\alpha_i = 0$, for all i, since $\{w_1, \cdots, w_r\}$ forms are orthonormal basis for the null space of $\tilde{S}_w$. This contradicts our assumption that not all of the α_i's and the β_j's are zero, Thus, $\{w_1, \cdots, w_r, b_1, \cdots, b_s\}$ are linearly independent. ■

Next, we show how to compute the optimal transformation of NLDA using these two null spaces. Recall that in NLDA, the null space of S_t may be removed first. In the following discussion, we work on the reduced scatter matrices $\tilde{S}_w$, $\tilde{S}_b$, and $\tilde{S}_t$ directly as in Lemma 5.1. The main result is summarized in the following theorem.

Theorem 5.1 *Let U_1, $\tilde{S}_t$, $\tilde{S}_b$, and $\tilde{S}_w$ be defined as above and $t = rank(\tilde{S}_t)$. Let $R = [W, B]$, where $W = [w_1, \cdots, w_r]$, $B = [b_1, \cdots, b_s]$, and $\{w_1, \cdots, w_r, b_1, \cdots, b_s\}$ are defined as in Lemma 5.1. Assume that condition C1: $rank(S_t) = rank(S_b) + rank(S_w)$ holds. Then $G = U_1 W M$ solves the optimization problem in Eq. (9), where the matrix M, consisting of the eigenvectors of $W^T \tilde{S}_b W$, is orthogonal.*

Proof From Lemma 5.1, $\{w_1, \cdots, w_r, b_1, \cdots, b_s\} \in \mathbb{R}^t$ is linearly independent. Condition C1 implies that $t = r + s$. Thus $\{w_1, \cdots, w_r, b_1, \cdots, b_s\}$ forms a basis for $\mathbb{R}^t$, that is, $R = [W, B]$ is nonsingular. It follows that

$$\begin{aligned} R^T \tilde{S}_t R &= R^T \tilde{S}_b R + R^T \tilde{S}_w R \\ &= \begin{pmatrix} W^T \tilde{S}_b W & W^T \tilde{S}_b B \\ B^T \tilde{S}_b W & B^T \tilde{S}_b B \end{pmatrix} + \begin{pmatrix} W^T \tilde{S}_w W & W^T \tilde{S}_w B \\ B^T \tilde{S}_w W & B^T \tilde{S}_w B \end{pmatrix} \\ &= \begin{pmatrix} W^T \tilde{S}_b W & 0 \\ 0 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 \\ 0 & B^T \tilde{S}_w B \end{pmatrix}. \end{aligned}$$

Since matrix $R^T \tilde{S}_t R$ has full rank, $W^T \tilde{S}_b W$, the projection of $\tilde{S}_b$ onto the null space of $\tilde{S}_w$, is nonsingular. Let $W^T \tilde{S}_b W = M \Delta_b M^T$ be the eigen-decomposition of $W^T \tilde{S}_b W$, where M is orthogonal and Δ_b is diagonal with positive diagonal entries (note that $W^T \tilde{S}_b W$ is positive definite). Then, from Section 3, the optimal transformation G of NLDA is given by $G = U_1 W M$. ■

Recall that the matrix M in NLDA is computed so that $\text{trace}(M^T W^T \tilde{S}_b W M)$ is maximized. Since $\text{trace}(QAQ^T) = \text{trace}(A)$ for any orthogonal Q, the solution in NLDA is invariant under an arbitrary orthogonal transformation. Thus $G = U_1 W$ is also a solution to NLDA, since M is orthogonal, as summarized in the following corollary.

Corollary 5.1 *Assume condition C1: $rank(S_t) = rank(S_b) + rank(S_w)$ holds. Let U_1 and W be defined as in Theorem 5.1. Then $G = U_1 W$ solves the optimization problem in Eq. (9). That is, $G = U_1 W$ is an optimal transformation of NLDA.*

Corollary 5.1 implies that when condition C1 holds, Step 5 in **Algorithm 1** may be removed, as well as the formation of $\tilde{S}_b$ in Step 3 and the multiplication of $U_1 W$ with M in Step 6. This improves the efficiency of the NLDA algorithm. The improved NLDA (iNLDA) algorithm is given in **Algorithm 3**. Note that it is recommended in (Liu et al., 2004) that the maximization of the between-class distance in Step 5 of **Algorithm 1** should be removed to avoid possible overfitting. However, Corollary 5.1 shows that under condition C1, the removal of Step 5 has no effect on the performance of the NLDA algorithm.

Next, we show the equivalence relationship between NLDA and OLDA, when condition C1 holds. The main result is summarized in the following theorem.

Algorithm 3: iNLDA (improved NLDA)
Input: data matrix A
Output: transformation matrix G
1. Form the matrix H_t as in Eq. (3);
2. Compute the reduced SVD of H_t as $H_t = U_1 \Sigma_t V_1^T$;
3. Construct the matrix $\tilde{S}_w = U_1^T S_w U_1$;
4. Compute the null space, W, of $\tilde{S}_w$, via the eigen-decomposition;
5. $G \leftarrow U_1 W$.

Theorem 5.2 *Assume that condition C1: $rank(S_t) = rank(S_b) + rank(S_w)$ holds. Let U_1 and W be defined as in Theorem 5.1. Then, $G = U_1 W$ solves the optimization problem in Eq. (10). That is, under the given assumption, OLDA and NLDA are equivalent.*

Proof Recall that the optimization involved in OLDA is

$$G = \text{argmax}_{G \in \mathbb{R}^{m \times \ell} : G^T G = I_\ell} \text{trace}\left((S_t^L)^+ S_b^L\right), \tag{16}$$

where $S_t^L = G^T S_t G$ and $S_b^L = G^T S_b G$. From Section 4, the maximum number, ℓ, of discriminant vectors is no larger than q, which is the rank of S_b. Recall that

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

where r is the dimension of the null space of $\tilde{S}_w$.

Based on the property of the trace of matrices, we have

$$\text{trace}\left((S_t^L)^+ S_b^L\right) + \text{trace}\left((S_t^L)^+ S_w^L\right) = \text{trace}\left((S_t^L)^+ S_t^L\right) = \text{rank}(S_t^L) \leq q = r,$$

where the second equality follows since $\text{trace}(A^+ A) = \text{rank}(A)$ for any square matrix A, and the inequality follows since the rank of $S_t^L \in \mathbb{R}^{\ell \times \ell}$ is at most $\ell \leq q$.

It follows that $\text{trace}\left((S_t^L)^+ S_b^L\right) \leq r$, since $\text{trace}\left((S_t^L)^+ S_w^L\right)$, the trace of the product of two positive semi-definite matrices, is always nonnegative. Next, we show that the maximum is achieved, when $G = U_1 W$.

Recall that the dimension of the null space, W, of $\tilde{S}_w$ is r. That is, $W \in \mathbb{R}^{t \times r}$. It follows that $(U_1 W)^T S_t (U_1 W) \in \mathbb{R}^{r \times r}$, and $\text{rank}((U_1 W)^T S_t (U_1 W)) = r$. Furthermore,

$$(U_1 W)^T S_w (U_1 W) = W^T \tilde{S}_w W = 0,$$

as W forms the null space of $\tilde{S}_w$. It follows that,

$$\text{trace}\left(\left((U_1 W)^T S_t (U_1 W)\right)^+ (U_1 W)^T S_w (U_1 W)\right) = 0.$$

Hence,

$$\begin{aligned} \text{trace}\left(\left((U_1 W)^T S_t (U_1 W)\right)^+ (U_1 W)^T S_b (U_1 W)\right) &= \text{rank}\left((U_1 W)^T S_t (U_1 W)\right) \\ - \quad \text{trace}\left(\left((U_1 W)^T S_t (U_1 W)\right)^+ \left((U_1 W)^T S_w (U_1 W)\right)\right) &= r. \end{aligned}$$

Thus $G = U_1 W$ solves the optimization problem in Eq. (10). That is, OLDA and NLDA are equivalent. ■

Theorem 5.2 above shows that under condition C1, OLDA and NLDA are equivalent. Next, we show that condition C1 holds when the data points are linearly independent as summarized below.

Theorem 5.3 *Assume that condition C2, that is, the n data points in the data matrix $A \in \mathbb{R}^{m\times n}$ are linearly independent, holds. Then condition C1: $rank(S_t) = rank(S_b) + rank(S_w)$ holds.*

Proof Since the n columns in A are linearly independent, $H_t = A - ce^T$ is of rank $n-1$. That is, $\mathrm{rank}(S_t) = n-1$. Next we show that $\mathrm{rank}(S_b) = k-1$ and $\mathrm{rank}(S_w) = n-k$. Thus condition C1 holds.

It is easy to verify that $\mathrm{rank}(S_b) \le k-1$ and $\mathrm{rank}(S_w) \le n-k$. We have

$$n-1 = \mathrm{rank}(S_t) \le \mathrm{rank}(S_b) + \mathrm{rank}(S_w) \le (k-1) + (n-k) = n-1. \tag{17}$$

It follows that all inequalities in Eq. (17) become equalities. That is,

$$\mathrm{rank}(S_b) = k-1,\ \mathrm{rank}(S_w) = n-k,\ \text{and } \mathrm{rank}(S_t) = \mathrm{rank}(S_b) + \mathrm{rank}(S_w). \tag{18}$$

Thus, condition C1 holds. ■

Our experimental results in Section 7 show that for high-dimensional data, the linear independence condition C2 holds in many cases, while condition C1 is satisfied in most cases. This explains why NLDA and OLDA often achieve the same performance in many applications involving high-dimensional data, such as text documents, face images, and gene expression data.

6. Regularized Orthogonal LDA

Recall that OLDA involves the pseudo-inverse of the total scatter matrix, whose estimation may not be reliable especially for undersampled data, where the number of dimensions exceeds the sample size. In such case, the parameter estimates can be highly unstable, giving rise to high variance. By employing a method of regularization, one attempts to improve the estimates by regulating this bias variance trade-off (Friedman, 1989). We employ the regularization technique to OLDA by adding a constant λ to the diagonal elements of the total scatter matrix. Here $\lambda > 0$ is known as the *regularization parameter*. The algorithm is called regularized OLDA (ROLDA). The optimal transformation, G^r, of ROLDA can be computed by solving the following optimization problem:

$$G^r = \mathrm{argmax}_{G\in\mathbb{R}^{m\times\ell}:\, G^TG=I_\ell} \mathrm{trace}\left(\left(G^T(S_t+\lambda I_m)G\right)^+ G^T S_b G\right). \tag{19}$$

The optimal G^r can be computed by solving an eigenvalue problem as summarized in the following theorem (The proof follows Theorem 3.1 in (Ye, 2005) and is thus omitted):

Theorem 6.1 *Let X_q^r be the matrix consisting of the first q eigenvectors of the matrix*

$$(S_t + \lambda I_m)^{-1} S_b \tag{20}$$

corresponding to the nonzero eigenvalues, where $q = rank(S_b)$. Let $X_q^r = QR$ be the QR-decomposition of X_q^r, where Q has orthonormal columns and R is upper triangular. Then $G = Q$ solves the optimization problem in Eq. (19).

Theorem 6.1 implies that the main computation involved in ROLDA is the eigen-decomposition of the matrix $(S_t+\lambda I_m)^{-1}S_b$. Direct formation of the matrix is expensive for high-dimensional data, as it is of size m by m. In the following, we present an efficient way of computing the eigen-decomposition. Denote

$$B^r = (\Sigma_t^2+\lambda I_t)^{-1/2}U_1^T H_b \tag{21}$$

and let

$$B^r = P^r\tilde{\Sigma}^r(Q^r)^T \tag{22}$$

be the SVD of B^r. From Eqs. (8) and (11), we have

$$\begin{aligned}
(S_t+\lambda I_m)^{-1}S_b &= U\begin{pmatrix}(\Sigma_t^2+\lambda I_t)^{-1} & 0\\ 0 & \lambda^{-1}I_{m-t}\end{pmatrix}U^TU\begin{pmatrix}U_1^TS_bU_1 & 0\\ 0 & 0\end{pmatrix}U^T\\
&= U\begin{pmatrix}(\Sigma_t^2+\lambda I_t)^{-1}U_1^TH_bH_b^TU_1 & 0\\ 0 & 0\end{pmatrix}U^T\\
&= U\begin{pmatrix}(\Sigma_t^2+\lambda I_t)^{-1/2}B^r(B^r)^T(\Sigma_t^2+\lambda I_t)^{1/2} & 0\\ 0 & 0\end{pmatrix}U^T\\
&= U\begin{pmatrix}(\Sigma_t^2+\lambda I_t)^{-1/2}P^r\tilde{\Sigma}^r(\tilde{\Sigma}^r)^T(P^r)^T(\Sigma_t^2+\lambda I_t)^{1/2} & 0\\ 0 & 0\end{pmatrix}U^T.
\end{aligned}$$

It follows that the columns of the matrix

$$U_1(\Sigma_t^2+\lambda I_t)^{-1/2}P_q^r$$

form the eigenvectors of $(S_t+\lambda I_m)^{-1}S_b$ corresponding to the top q nonzero eigenvalues, where P_q^r denotes the first q columns of P^r. That is, X_q^r in Theorem 6.1 is given by

$$X_q^r = U_1(\Sigma_t^2+\lambda I_t)^{-1/2}P_q^r. \tag{23}$$

The pseudo-code for the ROLDA algorithm is given in **Algorithm 4**. The computations in ROLDA can be decomposed into two components: the first component involves the matrix, $U_1\in\mathbb{R}^{m\times t}$, of high dimensionality but independent of λ, while the second component involves the matrix,

$$(\Sigma_t^2+\lambda I_t)^{-1/2}P_q^r\in\mathbb{R}^{t\times q},$$

of low dimensionality. When we apply cross-validation to search for the optimal λ from a set of candidates, we repeat the computations involved in the second component only, thus making the computational cost of model selection small.

More specifically, let

$$\Lambda = \{\lambda_1,\cdots,\lambda_{|\Lambda|}\} \tag{24}$$

be the candidate set for the regularization parameter λ, where $|\Lambda|$ denotes the size of the candidate set Λ. We apply v-fold cross-validation for model selection (we choose $v=5$ in our experiment), where the data is divided into v subsets of (approximately) equal size. All subsets are mutually exclusive, and in the i-th fold, the i-th subset is held out for testing and all other subsets are used for training. For each $\lambda_j\,(j=1,\cdots,|\Lambda|)$, we compute the cross-validation accuracy, $\mathrm{Accu}(j)$, defined as the mean of the accuracies for all folds. The optimal regularization value λ_{j^*} is the one with

$$j^* = \arg\max_j \mathrm{Accu}(j). \tag{25}$$

Algorithm 4: ROLDA (Regularized OLDA)

Input: data matrix A and regularization value λ
Output: transformation matrix G^r

1. Compute U_1, Σ_t, and P_q^r, where $q = \text{rank}(S_b)$;
2. $X_q^r \leftarrow U_1(\Sigma_t^2 + \lambda I_t)^{-1/2} P_q^r$;
3. Compute the QR decomposition of X_q^r as $X_q^r = QR$;
4. $G^r \leftarrow Q$.

The K-Nearest Neighbor algorithm with $K = 1$, called 1-NN, is used for computing the accuracy. The pseudo-code for the model selection procedure in ROLDA is given in **Algorithm 5**. Note that we apply the QR decomposition to

$$(\Sigma_t^2 + \lambda I_t)^{-1/2} P_q^r \in \mathbb{R}^{t \times q} \tag{26}$$

instead of

$$X_q^r = U_1(\Sigma_t^2 + \lambda I_t)^{-1/2} P_q^r \in \mathbb{R}^{m \times q}, \tag{27}$$

as done in Theorem 6.1, since U_1 has orthonormal columns.

Algorithm 5: Model selection for ROLDA

Input: data matrix A and candidate set $\Lambda = \{\lambda_1, \cdots, \lambda_{|\Lambda|}\}$
Output: optimal regularization value λ_{j^*}

1. For $i = 1 : v$ /* v-fold cross-validation */
2. Construct A^i and $A^{\hat{i}}$;
   /* $A^i = i$-th fold, for training and $A^{\hat{i}} =$ rest, for testing */
3. Construct H_b and H_t using A^i as in Eqs. (2) and (3), respectively;
4. Compute the reduced SVD of H_t as $H_t = U_1 \Sigma_t V_1^T$; $t \leftarrow \text{rank}(H_t)$;
5. $H_{b,L} \leftarrow U_1^T H_b$, $q \leftarrow \text{rank}(H_b)$;
6. $A_L^i \leftarrow U_1^T A^i$; $A_L^{\hat{i}} \leftarrow U_1^T A^{\hat{i}}$; /* Projection by U_1 */
7. For $j = 1 : |\Lambda|$ /* $\lambda_1, \cdots, \lambda_{|\Lambda|}$ */
8. $D_j \leftarrow (\Sigma_t^2 + \lambda_j I_t)^{-1/2}$; $B^r \leftarrow D_j H_{b,L}$
9. Compute the SVD of B^r as $B^r = P^r \tilde{\Sigma}^r (Q^r)^T$;
10. $D_{q,P} \leftarrow D_j P_q^r$; Compute the QR decomposition of $D_{q,P}$ as $D_{q,P} = QR$;
11. $A_L^i \leftarrow Q^T A_L^i$; $A_L^{\hat{i}} \leftarrow Q^T A_L^{\hat{i}}$;
12. Run 1-NN on $\left(A_L^i, A_L^{\hat{i}}\right)$ and compute the accuracy, denoted as $\text{Accu}(i, j)$;
13. EndFor
14. EndFor
15. $\text{Accu}(j) \leftarrow \frac{1}{v} \sum_{i=1}^{v} \text{Accu}(i, j)$;
16. $j^* \leftarrow \arg\max_j \text{Accu}(j)$;
17. Output λ_{j^*} as the optimal regularization value.

6.1 Time Complexity

We conclude this section by analyzing the time complexity of the model selection procedure described above.

Line 4 in **Algorithm 5** takes $O(n^2m)$ time for the reduced SVD computation. Lines 5 and 6 take $O(mtk) = O(mnk)$ and $O(tmn) = O(mn^2)$ time, respectively, for the matrix multiplications. For each λ_j, for $j = 1, \cdots, |\Lambda|$, of the "For" loop, Lines 9 and 10 take $O(tk^2) = O(nk^2)$ time for the SVD and QR decomposition and matrix multiplication. Line 11 takes $O(ktn) = O(kn^2)$ time for the matrix multiplication. The computation of the classification accuracy by 1-NN in Line 12 takes $O(n^2k/v)$ time, as the size of the test set, $A_L^{\hat{i}}$, is about n/v. Thus, the time complexity, $T(|\Lambda|)$, of the model selection procedure is

$$T(|\Lambda|) \quad = \quad O\left(v\left(n^2m + mn^2 + mnk + |\Lambda|(nk^2 + kn^2 + n^2k/v)\right)\right).$$

For high-dimensional and undersampled data, where the sample size, n, is much smaller than the dimensionality m, the time complexity is simplified to

$$T(|\Lambda|) = O\left(v(n^2m + |\Lambda|n^2k)\right) = O\left(vn^2m\left(1 + \frac{k}{m}|\Lambda|\right)\right).$$

When the number, k, of classes in the data set is much smaller than the dimensionality, m, the overhead of estimating the optimal regularization value among a large candidate set may be small. Our experiments on a collection of high-dimensional and undersampled data (see Section 7) show that the computational cost of the model selection procedure in ROLDA grows slowly as $|\Lambda|$ increases.

7. Experimental Studies

In this section, we perform extensive experimental studies to evaluate the theoretical results and the ROLDA algorithm presented in this paper. Section 7.1 describes our test data sets. We perform a detailed comparison of NLDA, iNLDA, and OLDA in Section 7.2. Results are consistent with our theoretical analysis. In Section 7.3, we compare the classification performance of NLDA, iNLDA, OLDA, ULDA, ROLDA, and SVM. The K-Nearest-Neighbor (K-NN) algorithm with $K = 1$ is used as the classifier for all LDA based algorithms.

7.1 Data Sets

We used 14 data sets from various data sources in our experimental studies. The statistics of our test data sets are summarized in Table 2.

The first five data sets, including spambase,[4] balance, wine, waveform, and vowel, are low-dimensional data from the UCI Machine Learning Repository. The next nine data sets, including text documents, face images, and gene expression data, have high dimensionality: re1, re0, and tr41 are three text document data sets, where re1 and re0 are derived from *Reuters-21578* text categorization test collection Distribution 1.0,[5] and tr41 is derived from the TREC-5, TREC-6, and TREC-7 collections;[6] ORL,[7] AR,[8] and PIX[9] are three face image data sets; GCM, colon, and ALLAML4 are three gene expression data sets (Ye et al., 2004b).

4. Only a subset of the original spambase data set is used in our study.
5. http://www.daviddlewis.com/resources/testcollections/reuters21578/
6. http://trec.nist.gov
7. http://www.uk.research.att.com/facedatabase.html
8. http://rvl1.ecn.purdue.edu/∼aleix/aleix_face_DB.html
9. http://peipa.essex.ac.uk/ipa/pix/faces/manchester/test-hard/

Data Set	Sample size (n)			# of dimensions	# of classes
	training	test	total	(m)	(k)
spambase	400	600	1000	56	2
balance	416	209	625	4	3
wine	118	60	178	13	3
waveform	300	500	800	21	3
vowel	528	462	990	10	11
re1	—	—	490	3759	5
re0	—	—	320	2887	4
tr41	—	—	210	7454	7
ORL	—	—	400	10304	40
AR	—	—	650	8888	50
PIX	—	—	300	10000	30
GCM	—	—	198	16063	14
colon	—	—	62	2000	2
ALLAML4	—	—	72	7129	4

Table 2: Statistics of our test data sets. For the first five data sets, we used the given partition of training and test sets, while for the last nine data sets, we did random splittings into training and test sets of ratio 2:1.

7.2 Comparison of NLDA, iNLDA, and OLDA

In this experiment, we did a comparative study of NLDA, iNLDA, and OLDA. For the first five low-dimensional data sets from the UCI Machine Learning Repository, we used the given splitting of training and test sets. The result is summarized in Table 3. For the next nine high-dimensional data sets, we performed our study by repeated random splittings into training and test sets. The data was partitioned randomly into a training set, where each class consists of two-thirds of the whole class and a test set with each class consisting of one-third of the whole class. The splitting was repeated 20 times and the resulting accuracies of different algorithms for the first ten splittings are summarized in Table 4. Note that the mean accuracy for the 20 different splittings will be reported in the next section. The rank of three scatter matrices, S_b, S_w, and S_t, for each of the splittings is also reported.

The main observations from Table 3 and Table 4 include:

- For the first five low-dimensional data sets, we have $\text{rank}(S_b) = k-1$, and $\text{rank}(S_w) = \text{rank}(S_t) = m$, where m is the data dimensionality. Thus the null space of $\tilde{S}_w$ is empty, and both NLDA and iNLDA do not apply. However, OLDA is applicable and the reduced dimensionality of OLDA is $k-1$.

- For the next nine high-dimensional data sets, condition C1: $\text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w)$ is satisfied in all cases except the re0 data set. For the re0 data set, either $\text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w)$ or $\text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w) - 1$ holds, that is, condition C1 is not severely violated for re0. Note that re0 has the smallest number of dimensions among the nine high-

		Data Set				
		spambase	balance	wine	waveform	vowel
Method	NLDA	—	—	—	—	—
	iNLDA	—	—	—	—	—
	OLDA	88.17	86.60	98.33	73.20	56.28
Rank	S_b	1	2	2	2	10
	S_w	56	4	13	21	10
	S_t	56	4	13	21	10

Table 3: Comparison of NLDA, iNLDA, and OLDA on classification accuracy (in percentage) using five low-dimensional data sets from UCI Machine Learning Repository. The ranks of three scatter matrices are reported.

dimensional data sets. From the experiments, we may infer that condition C1 is more likely to hold for high-dimensional data.

- NLDA, iNLDA, and OLDA achieve the same classification performance in all cases when condition C1 holds. The empirical result confirms the theoretical analysis in Section 5. This explains why NLDA and OLDA often achieve similar performance for high-dimensional data. We can also observe that NLDA and iNLDA achieve similar performance in all cases.

- The numbers of training data points for the nine high-dimensional data (in the same order as in the table) are 325, 212, 140, 280, 450, 210, 125, 68, and 48, respectively. By examining the rank of S_t in Table 4, we can observe that the training data in six out of nine data sets, including tr41, ORL, AR, GCM, colon, and ALLAML4, are linearly independent. That is, the independence assumption C2 from Theorem 5.3 holds for these data sets. It is clear from the table that for these six data sets, condition C1 holds and NLDA, iNLDA, and OLDA achieve the same performance. These are consistent with the theoretical analysis in Section 5.

- For the re0 data set, where condition C1 does not hold, i.e., $\text{rank}(S_t) < \text{rank}(S_b) + \text{rank}(S_w)$, OLDA achieves higher classification accuracy than NLDA and iNLDA. Recall that the reduced dimensionality of OLDA equals $\text{rank}(S_b) \equiv q$. The reduced dimensionality in NLDA and iNLDA equals the dimension of the null space of $\tilde{S}_w$, which equals $\text{rank}(S_t) - \text{rank}(S_w) < \text{rank}(S_b)$. That is, OLDA keeps more dimensions in the transformed space than NLDA and iNLDA. Experimental results in re0 show that these extra dimensions used in OLDA improve its classification performance.

7.3 Comparative Studies on Classification

In this experiment, we conducted a comparative study of NLDA, iNLDA, OLDA, ULDA, ROLDA, and SVM in terms of classification. For ROLDA, the optimal λ is estimated through cross-validation on a candidate set, $\Lambda = \{\lambda_j\}_{j=1}^{|\Lambda|}$. Recall that $T(|\Lambda|)$ denotes the computational cost of the model selection procedure in ROLDA, where $|\Lambda|$ is the size of the candidate set of the regularization values. We have performed model selection on all nine high-dimensional data sets using different values of

Data Set	Method	Ten different splittings into training and test sets of ratio 2:1									
re1	NLDA	92.73	93.33	93.33	93.94	94.55	95.15	96.36	95.15	92.12	93.94
	iNLDA	92.73	93.33	93.33	93.94	94.55	95.15	96.36	95.15	92.12	93.94
	OLDA	92.73	93.33	93.33	93.94	94.55	95.15	96.36	95.15	92.12	93.94
	S_b	4	4	4	4	4	4	4	4	4	4
	S_w	316	318	319	316	316	320	316	318	317	318
	S_t	320	322	323	320	320	324	320	322	321	322
re0	NLDA	64.81	62.04	64.81	68.52	87.96	70.37	71.30	73.15	87.04	75.93
	iNLDA	65.74	62.04	64.81	69.44	87.96	70.37	71.30	72.22	87.04	75.93
	OLDA	75.93	75.00	77.78	74.07	87.96	80.56	74.07	78.70	87.04	79.63
	S_b	3	3	3	3	3	3	3	3	3	3
	S_w	205	204	203	203	205	204	201	203	203	205
	S_t	207	206	205	205	208	206	203	205	206	207
tr41	NLDA	97.14	95.71	97.14	98.57	97.14	98.57	100.0	95.71	98.57	95.71
	iNLDA	97.14	95.71	97.14	98.57	97.14	98.57	100.0	95.71	98.57	95.71
	OLDA	97.14	95.71	97.14	98.57	97.14	98.57	100.0	95.71	98.57	95.71
	S_b	6	6	6	6	6	6	6	6	6	6
	S_w	133	133	133	133	133	133	133	133	133	133
	S_t	139	139	139	139	139	139	139	139	139	139
ORL	NLDA	99.17	96.67	98.33	98.33	95.00	95.83	98.33	97.50	98.33	95.83
	iNLDA	99.17	96.67	98.33	98.33	95.00	95.83	98.33	97.50	98.33	95.83
	OLDA	99.17	96.67	98.33	98.33	95.00	95.83	98.33	97.50	98.33	95.83
	S_b	39	39	39	39	39	39	39	39	39	39
	S_w	240	240	240	240	240	240	240	240	240	240
	S_t	279	279	279	279	279	279	279	279	279	279
AR	NLDA	96.50	94.50	96.50	94.00	93.50	94.50	93.50	97.00	94.00	96.00
	iNLDA	96.50	94.50	96.50	94.00	93.50	94.50	93.50	97.00	94.00	96.00
	OLDA	96.50	94.50	96.50	94.00	93.50	94.50	93.50	97.00	94.00	96.00
	S_b	49	49	49	49	49	49	49	49	49	49
	S_w	400	400	400	400	400	400	400	400	400	400
	S_t	449	449	449	449	449	449	449	449	449	449
PIX	NLDA	98.89	97.78	98.89	97.78	98.89	98.89	98.89	97.78	98.89	97.78
	iNLDA	98.89	97.78	98.89	97.78	98.89	98.89	98.89	97.78	98.89	97.78
	OLDA	98.89	97.78	98.89	97.78	98.89	98.89	98.89	97.78	98.89	97.78
	S_b	29	29	29	29	29	29	29	29	29	29
	S_w	178	179	179	179	178	180	179	179	180	178
	S_t	207	208	208	208	207	209	208	208	209	207
GCM	NLDA	81.54	80.00	81.54	83.08	84.62	87.69	75.38	78.46	84.62	83.08
	iNLDA	81.54	80.00	81.54	83.08	84.62	87.69	75.38	78.46	84.62	83.08
	OLDA	81.54	80.00	81.54	83.08	84.62	87.69	75.38	78.46	84.62	83.08
	S_b	13	13	13	13	13	13	13	13	13	13
	S_w	111	111	111	111	111	111	111	111	111	111
	S_t	124	124	124	124	124	124	124	124	124	124
colon	NLDA	91.18	94.12	100.0	97.06	91.18	91.18	97.06	94.12	94.12	97.06
	iNLDA	91.18	94.12	100.0	97.06	91.18	91.18	97.06	94.12	94.12	97.06
	OLDA	91.18	94.12	100.0	97.06	91.18	91.18	97.06	94.12	94.12	97.06
	S_b	1	1	1	1	1	1	1	1	1	1
	S_w	66	66	66	66	66	66	66	66	66	66
	S_t	67	67	67	67	67	67	67	67	67	67
ALLAML4	NLDA	95.83	91.67	95.83	95.83	87.50	95.83	95.83	100.0	91.67	95.83
	iNLDA	95.83	91.67	95.83	95.83	87.50	95.83	95.83	100.0	91.67	95.83
	OLDA	95.83	91.67	95.83	95.83	87.50	95.83	95.83	100.0	91.67	95.83
	S_b	3	3	3	3	3	3	3	3	3	3
	S_w	44	44	44	44	44	44	44	44	44	44
	S_t	47	47	47	47	47	47	47	47	47	47

Table 4: Comparison of classification accuracy (in percentage) for NLDA, iNLDA, and OLDA using nine high-dimensional data sets. Ten different splittings into training and test sets of ratio 2:1 (for each of the k classes) are applied. The rank of three scatter matrices for each splitting is reported.

Data Set	NLDA	iNLDA	OLDA	ULDA	ROLDA	SVM
re1	94.33 (1.72)	94.33 (1.72)	94.33 (1.72)	94.76 (1.67)	94.79 (1.64)	94.54 (1.88)
re0	74.03 (9.22)	74.15 (8.19)	79.54 (4.73)	79.72 (4.82)	85.79 (3.66)	85.87 (3.34)
tr41	97.00 (2.01)	97.00 (2.01)	97.00 (2.01)	97.14 (2.02)	97.17 (2.04)	97.14 (2.01)
ORL	97.29 (1.79)	97.29 (1.79)	97.29 (1.79)	92.75 (1.82)	97.52 (1.64)	97.55 (1.34)
AR	95.42 (1.30)	95.42 (1.30)	95.42 (1.30)	94.37 (1.46)	97.30 (1.32)	95.75 (1.43)
PIX	98.22 (1.41)	98.22 (1.41)	98.22 (1.41)	96.61 (1.92)	98.29 (1.32)	98.50 (1.24)
GCM	81.77 (3.61)	81.77 (3.61)	81.77 (3.61)	80.46 (3.71)	82.69 (3.42)	75.31 (4.45)
Colon	86.50 (5.64)	86.50 (5.64)	86.50 (5.64)	86.50 (5.64)	87.00 (6.16)	87.25 (5.25)
ALLAML4	93.54 (3.70)	93.54 (3.70)	93.54 (3.70)	93.75 (3.45)	93.75 (3.45)	93.70 (3.40)

Table 5: Comparison of classification accuracy (in percentage) for six different methods: NLDA, iNLDA, OLDA, ULDA, ROLDA, and SVM using nine high-dimensional data sets. The mean accuracy and standard deviation (in parenthesis) from 20 different runs are reported.

$|\Lambda|$. We have observed that $T(|\Lambda|)$ grows slowly as $|\Lambda|$ increases, and the ratio, $T(1024)/T(1)$, on all nine data sets ranges from 1 to 5. Thus, we can run model selection using a large candidate set of regularization values, without dramatically increasing the cost. In the following experiments, we apply model selection to ROLDA with a candidate set of size $|\Lambda| = 1024$, where

$$\lambda_j = \alpha_j/(1-\alpha_j), \tag{28}$$

with $\{\alpha_j\}_{j=1}^{|\Lambda|}$ uniformly distributed between 0 and 1. As for SVM, we employed the cross-validation to estimate the optimal parameter using a candidate set of size 50. To compare different classification algorithms, we applied the same experimental setting as in Section 7.2. The splitting into training and test sets of ratio 2:1 (for each of the *k* classes) was repeated 20 times. The final accuracy reported was the average of the 20 different runs. The standard deviation for each data set was also reported. The result on the nine high-dimensionality data sets is summarized in Table 5.

As observed in Section 7.2, OLDA has the same performance as NLDA and iNLDA in all cases except the re0 data set, while NLDA and iNLDA achieve similar performance in all cases. Overall, ROLDA and SVM are very competitive with other methods. SVM performs well in all cases except GCM. The poor performance of SVM in GCM has also been observed in (Li et al., 2004). ROLDA outperforms OLDA for re0, AR, and GCM, while it is comparable to OLDA for all other cases. This confirms the effectiveness of the regularization applied in ROLDA. Note that from Remark 1, ULDA is closely related to OLDA. However, unlike OLDA, ULDA does not apply the final orthogonalization step. Experimental result in Table 5 confirms the effectiveness of the orthogonalization step in OLDA, especially for three face image data sets and GCM.

8. Conclusions

In this paper, we present a computational and theoretical analysis of two LDA based algorithms, including null space LDA and orthogonal LDA. NLDA computes the discriminant vectors in the null space of the within-class scatter matrix, while OLDA computes a set of orthogonal discriminant vectors via the simultaneous diagonalization of the scatter matrices. They have been applied successfully in many applications, such as document classification, face recognition, and gene expression data classification.

Both NLDA and OLDA result in orthogonal transformations. However, they applied different schemes in deriving the optimal transformation. Our theoretical analysis in this paper shows that under a mild condition C1 which holds in many applications involving high-dimensional data, NLDA is equivalent to OLDA. Based on the theoretical analysis, an improved algorithm for null space LDA algorithm, called iNLDA, is proposed. We have performed extensive experimental studies on 14 data sets, including both low-dimensional and high-dimensional data. Results have shown that condition C1 holds for eight out of the nine high-dimensional data sets, while the null space of $\tilde{S}_w$ is empty for all five low-dimensional data. Thus, NLDA may not be applicable for low-dimensional data, while OLDA is still applicable in this case. Results are also consistent with our theoretical analysis. That is, for all cases when condition C1 holds, NLDA, iNLDA, and OLDA achieve the same classification performance. We also observe that for other cases with condition C1 violated, OLDA outperforms NLDA and iNLDA, due to the extra number of dimensions used in OLDA. We also compare NLDA, iNLDA, and OLDA with uncorrelated LDA (ULDA), which does not perform the final orthogonalization step. Results show that OLDA is very competitive with ULDA, which confirms the effectiveness of the orthogonalization step used in OLDA. Our empirical and theoretical results presented in this paper provide further insights into the nature of these two LDA based algorithms.

We also present the ROLDA algorithm, which extends the OLDA algorithm by applying the regularization technique. Regularization may stabilize the sample covariance matrix estimation and improve the classification performance. ROLDA involves the regularization parameter λ, which is commonly estimated via cross-validation. To speed up the cross-validation process, we decompose the computations in ROLDA into two components: the first component involves matrices of high dimensionality but independent of λ, while the second component involves matrices of low dimensionality. When searching for the optimal λ from a candidate set, we repeat the computations involved in the second component only. A comparative study on classification shows that ROLDA is very competitive with OLDA, which shows the effectiveness of the regularization applied in ROLDA.

Our extensive experimental studies have shown that condition C1 holds for most high-dimensional data sets. We plan to carry out theoretical analysis on this property in the future. Some of the theoretical results in (Hall et al., 2005) may be useful for our analysis.

The algorithms in (Yang et al., 2005; Yu and Yang, 2001) are closely related to the null space LDA algorithm discussed in this paper. The analysis presented in this paper may be useful in understanding why these algorithms perform well in many applications, especially in face recognition. We plan to explore this further in the future.

Acknowledgements

We thank the reviewers for helpful comments. Research of JY is sponsored, in part, by the Center for Evolutionary Functional Genomics of the Biodesign Institute at the Arizona State University.

References

P. N. Belhumeour, J. P. Hespanha, and D. J. Kriegman. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Trans Pattern Analysis and Machine Intelligence*, 19 (7):711–720, 1997.

R. Bellmanna. *Adaptive Control Processes: A Guided Tour*. Princeton University Press, 1961.

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

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

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

L. Duchene and S. Leclerq. An optimal transformation for discriminant and principal component analysis. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 10(6):978–983, 1988.

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

S. Dudoit, J. Fridlyand, and T. P. Speed. Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 97(457): 77–87, 2002.

D. H. Foley and J. W. Sammon. An optimal set of discriminant vectors. *IEEE Trans Computers*, 24 (3):281–289, 1975.

J. H. Friedman. Regularized discriminant analysis. *Journal of the American Statistical Association*, 84(405):165–175, 1989.

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

G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, Baltimore, MD, USA, third edition, 1996.

P. Hall, J. S. Marron, and A. Neeman. Geometric representation of high dimension, low sample size data. *Journal of the Royal Statistical Society series B*, 67:427–444, 2005.

T. Hastie, R. Tibshirani, and J. H. Friedman. *The Elements of Statistical Learning : Data Mining, Inference, and Prediction*. Springer, 2001.

R. Huang, Q. Liu, H. Lu, and S. Ma. Solving the small sample size problem of LDA. In *Proc. International Conference on Pattern Recognition*, pages 29–32, 2002.

Z. Jin, J. Y. Yang, Z. S. Hu, and Z. Lou. Face recognition based on the uncorrelated discriminant transformation. *Pattern Recognition*, 34:1405–1416, 2001.

I. T. Jolliffe. *Principal Component Analysis*. Springer-Verlag, New York, 1986.

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

T. Li, C. Zhang, and M. Ogihara. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20(15): 2429–2437, 2004.

W. Liu, Y. Wang, S. Z. Li, and T. Tan. Null space approach of Fisher discriminant analysis for face recognition. In *Proc. European Conference on Computer Vision, Biometric Authentication Workshop*, 2004.

J. Lu, K. N. Plataniotis, and A. N. Venetsanopoulos. Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Networks*, 14(1):117–126, 2003.

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

B. Schökopf and A. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization and Beyond*. MIT Press, 2002.

M. Skurichina and R. P. W. Duin. Stabilizing classifiers for very small sample size. In *Proc. International Conference on Pattern Recognition*, pages 891–896, 1996.

D. L. Swets and J. Y. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 18(8):831–836, 1996.

M. A. Turk and A. P. Pentland. Face recognition using Eigenfaces. In *Proc. Computer Vision and Pattern Recognition Conference*, pages 586–591, 1991.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.

J. Yang, A. F. Frangi, J. Y. Yang, D. Zhang, and Z. Jin. KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(2):230– 244, 2005.

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

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

J. Ye, T. Li, T. Xiong, and R. Janardan. Using uncorrelated discriminant analysis for tissue classification with gene expression data. *IEEE/ACM Trans. Computational Biology and Bioinformatics*, 1(4):181–190, 2004b.

H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with applications to face recognition. *Pattern Recognition*, 34:2067–2070, 2001.