Generalized Linear Discriminant Analysis: A Unified Framework and Efficient Model Selection
Shuiwang Ji, Student Member, IEEE, and Jieping Ye, Member, IEEE

Abstract—High-dimensional data are common in many domains, and dimensionality reduction is the key to cope with the curse-of-dimensionality. Linear discriminant analysis (LDA) is a well-known method for supervised dimensionality reduction. When dealing with high-dimensional and low sample size data, classical LDA suffers from the singularity problem. Over the years, many algorithms have been developed to overcome this problem, and they have been applied successfully in various applications. However, there is a lack of a systematic study of the commonalities and differences of these algorithms, as well as their intrinsic relationships. In this paper, a unified framework for generalized LDA is proposed, which elucidates the properties of various algorithms and their relationships. Based on the proposed framework, we show that the matrix computations involved in LDA-based algorithms can be simplified so that the cross-validation procedure for model selection can be performed efficiently. We conduct extensive experiments using a collection of high-dimensional data sets, including text documents, face images, gene expression data, and gene expression pattern images, to evaluate the proposed theories and algorithms.

Index Terms—Dimensionality reduction, linear discriminant analysis (LDA), model selection, principal component analysis (PCA), regularization, visualization.

I. INTRODUCTION

Learning in high-dimensional spaces is challenging because data points are far apart from each other in such spaces, and the similarities between data points are difficult to compare and analyze [7]. This phenomenon is traditionally known as the curse-of-dimensionality [3], which states that an enormous number of samples is required to perform accurate predictions on problems with high dimensionality. Meanwhile, there has been increasing prevalence of data sets that contain a large number of dimensions, including microarray gene expression data [23], [24], [27], gene expression pattern images [14], [33], text documents [22], and face images [5], etc. The proliferation of these data has tempted the researchers to discover knowledge and extract patterns from the data using computational approaches. Dimensionality reduction, which extracts a small number of features by removing irrelevant, redundant, and noisy information, can be an effective solution [31]. The commonly used dimensionality reduction methods include supervised approaches such as linear discriminant analysis (LDA) [10], [11], and unsupervised ones such as principal component analysis (PCA) [18], and additional spectral and manifold learning methods [2], [4], [25], [26], [28]. When class labels are available, the supervised approaches such as LDA are usually more effective than the unsupervised ones such as PCA for classification.

LDA is a classical statistical approach for dimensionality reduction [9], [11], [16], [21]. LDA computes an optimal transformation (projection) by minimizing the within-class distance and maximizing the between-class distance simultaneously, thus achieving maximum class discrimination. The optimal transformation in LDA can be readily computed by applying an eigendecomposition on the scatter matrices. It has been used widely in many applications involving high-dimensional data [1], [6], [13], [20], [29], [32]. However, classical LDA requires the so-called total scatter matrix to be nonsingular. In many applications involving high-dimensional and low sample size data, the total scatter matrix can be singular, because the data points are from a very high-dimensional space and the sample size may not exceed this dimension. This is known as the singularity problem in LDA.

In recent years, many approaches have been proposed to deal with the singularity problem, including PCA+LDA [1], regularized LDA (RLDA) [13], null space LDA (NLDA) [6], orthogonal centroid method (OCM) [22], uncorrelated LDA (ULDA) [32], orthogonal LDA (OLDA) [32], and LDA/GSVD [17]. Different algorithms have been applied successfully in various domains, such as PCA+LDA in face recognition [1], OCM in text categorization [22], and RLDA in microarray gene expression data analysis [13]. However, there is a lack of a systematic study to explore the commonalities and differences of these algorithms, as well as their intrinsic relationships. This has been a challenging task, because different algorithms apply different schemes when dealing with the singularity problem.

In this paper, we propose a unified framework for generalized LDA via a transfer function. We show that various LDA-based algorithms differ in their transfer functions. Details on this unified framework as well as the transfer functions for different LDA-based algorithms are given in Section III. The proposed framework elucidates the properties of various algorithms and their relationships. In particular, ULDA is shown to be a special case of PCA+LDA and RLDA. We show that under a mild condition that has been shown to hold for many high-dimensional data sets, the ULDA transformation maps all data points from the same class to a common vector. This leads to a perfect separation between different classes in the dimensionality-reduced space. However, this may also result in overfitting. PCA+LDA

Manuscript received November 7, 2007; revised March 31, 2008; accepted June 3, 2008. First published September 26, 2008; current version published October 8, 2008. This work was supported in part by the Arizona State University and by the National Science Foundation under Grant IIS-0612069.

The authors are with the Department of Computer Science and Engineering, and the Center for Evolutionary Functional Genomics of The Biosign Institute, Arizona State University, Tempe, AZ 85287 USA (e-mail: shuiwang.ji@asu.edu; jieping.ye@asu.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2002078

1045-9227/$25.00 © 2008 IEEE

Authorized licensed use limited to: Arizona State University. Downloaded on October 13, 2008 at 12:25 from IEEE Xplore. Restrictions apply.
and RLDA overcome the overfitting problem by applying the PCA dimensionality reduction and the regularization, respectively.

In practice, a challenging issue in applying PCA+LDA and RLDA is the estimation of model parameters, i.e., the dimension for intermediate PCA-subspace in PCA+LDA and the regularization parameter in RLDA. Motivated by the theoretical properties of various methods and their relationships in the proposed unified framework, we develop model selection algorithms for PCA+LDA and RLDA in this paper. The model selection algorithm for PCA+LDA can choose an optimal dimension for the intermediate PCA-subspace from the set of all possible values efficiently. For RLDA, the number of possible values for the regularization parameter is infinite, and we show that the matrix computations involved in RLDA can be simplified so that the cross-validation procedure for model selection can be performed efficiently. Extensive experiments have been conducted on a collection of high-dimensional data sets to evaluate the proposed theories and algorithms. We also show that methods based on dimensionality reduction facilitate data visualization, and this is particularly useful for outlier detection in gene expression pattern images.

The rest of this paper is organized as follows. We give an overview of LDA and its generalizations in Section II. A unified framework for generalized LDA is presented in Section III and some theoretical properties of various techniques and their relationships are also discussed. Section IV proposes the model selection algorithms for PCA+LDA and RLDA. Experimental results are presented in Section V. This paper concludes in Section VI with discussions and future work.

II. OVERVIEW OF LINEAR DISCRIMINANT ANALYSIS

LDA is a supervised dimensionality reduction technique. It projects high-dimensional data onto a lower dimensional space by maximizing the separation of data points from different classes and minimizing the dispersion of data from the same class simultaneously, thus achieving maximum class discrimination in the dimensionality-reduced space. Given a data matrix \(X = [x_1, \ldots, x_n] \in \mathbb{R}^{d \times n}\) consisting of \(n\) samples \(\{x_i\}_{i=1}^n\) in \(\mathbb{R}^d\), we focus on linear feature extraction that constructs a small number \(\ell\) of features by applying a linear transformation \(G \in \mathbb{R}^{d \times \ell}\) that maps each data point \(x_i\), for \(1 \leq i \leq n\), in the \(d\)-dimensional space to a vector \(x_i^G\) in the \(\ell\)-dimensional space as follows:

\[x_i \in \mathbb{R}^d \rightarrow x_i^G = G^T x_i \in \mathbb{R}^\ell (\ell < d).\]

Assume that the data matrix \(X\) is grouped as

\[X = [X_1, \ldots, X_k]\]

where \(k\) is the number of classes, \(X_i \in \mathbb{R}^{d \times n_i}\) is the data matrix consisting of data points from the \(i\)th class, \(n_i\) is the sample size of the \(i\)th class, and \(\sum_{i=1}^k n_i = n\). In LDA, three matrices, called within-class, between-class, and total scatter matrices, are defined as follows:

\[S_w = \frac{1}{n} \sum_{i=1}^k \sum_{x \in X_i} (x - c_i)(x - c_i)^T \]  
\[S_b = \frac{1}{n} \sum_{i=1}^k n_i (c_i - c)(c_i - c)^T \]  
\[S_t = \frac{1}{n} \sum_{i=1}^n (x_i - c)(x_i - c)^T \]

where \(c_i\) is the centroid of the \(i\)th class, and \(c\) is the global centroid. It can be verified that \(S_t = S_w + S_b\) [11]. Define three matrices as follows:

\[H_w = \frac{1}{\sqrt{n}} [X_1 - c_1(e_1)^T, \ldots, X_k - c_k(e_k)^T] \]  
\[H_b = \frac{1}{\sqrt{n}} \sqrt{n_1} (c_1 - c), \ldots, \sqrt{n_k} (c_k - c) \]  
\[H_t = \frac{1}{\sqrt{n}} (X - c e^T) \]

where \(e_i \in \mathbb{R}^{n_i}\) and \(e \in \mathbb{R}^n\) are vectors of all ones. Then, the three scatter matrices in (1)–(3) can be expressed as

\[S_w = H_w H_w^T \]  
\[S_b = H_b H_b^T \]  
\[S_t = H_t H_t^T . \]

It follows from (1) and (2) that

\[\text{trace}(S_w) = \frac{1}{n} \sum_{i=1}^k \sum_{x \in X_i} ||x - c_i||^2_2 \]  
\[\text{trace}(S_b) = \frac{1}{n} \sum_{i=1}^k n_i ||c_i - c||^2_2 . \]

It can be observed that the trace of \(S_w\) measures the closeness of each data point to its class centroid, and the trace of \(S_b\) measures the distances between each class centroid to the global centroid. In the dimensionality-reduced space transformed by \(G\), the three scatter matrices become

\[S_w^L = G^T S_w G \]  
\[S_b^L = G^T S_b G \]  
\[S_t^L = G^T S_t G^T . \]

LDA aims to maximize the between-class distance and minimize the within-class distance in the dimensionality-reduced space. This can be achieved by minimizing \(\text{trace}(S_b^L)\) and maximizing \(\text{trace}(S_w^L)\) simultaneously. In classical LDA, the following objective function is commonly maximized [11]:

\[f(G) = \text{trace}((G^T S_w G)^{-1} G^T S_b G) . \]

It can be shown [11] that solution to the classical LDA corresponding to the objective function in (10) can be obtained by solving the following generalized eigenvalue problem:

\[S_b y_i = \lambda_i S_w y_i \]  

Authorized licensed use limited to: Arizona State University. Downloaded on October 13, 2008 at 12:25 from IEEE Xplore. Restrictions apply.
whose eigenvectors corresponding to the \( k - 1 \) largest eigenvalues form the columns of transformation matrix \( G \). When \( S_w \) is nonsingular, this problem becomes an eigenvalue problem as
\[
S_w^{-1} S_b \theta = \lambda \theta \tag{12}
\]
Note that because \( S_t = S_w + S_b \), \( S_t \) can be used in place of \( S_w \) in optimization problems in (10) [11], [21], [32], [34].

When the data dimensionality is larger than the sample size, which is the case for many high-dimensional and low sample size data, all of the three scatter matrices are singular and the classical LDA cannot be applied directly. This is known as the singularity or undersampled problem in LDA. In recent years, various generalizations of classical LDA to deal with this singularity problem have been proposed. In Section II-A, we first review those generalizations of LDA. To elucidate the commonalities and differences of various techniques, a general framework is proposed in Section III that unifies various approaches.

### III. A Unified Framework for Generalized LDA

The generalizations of LDA discussed in the last section employ various techniques to deal with the singularity problem. However, there is a lack of systematic study that elucidates the commonalities and differences of diverse algorithms. In this section, we propose a four-step general framework for generalized LDA algorithms. The proposed framework unifies most of the generalized LDA algorithms, and the properties of various algorithms and their relationships are elucidated from this framework. The unified framework consists of four steps.

1. Compute the set of eigenvalues \( \{\lambda_i\}^d_{i=1} \) of \( S_t \) in (3) and the corresponding eigenvectors \( \{\theta_i\}^d_{i=1} \) with \( \lambda_1 \geq \cdots \geq \lambda_d \). Then, \( S_t \) can be expressed as \( S_t = \sum_{i=1}^{d} \lambda_i \theta_i \theta_i^T \).

2. Given a transfer function \( \Phi \), let \( \lambda_i \equiv \Phi(\lambda_i) \), for all \( i \). Construct matrix \( \tilde{S}_t = \sum_{i=1}^{d} \lambda_i \theta_i \theta_i^T \).

3. Compute the set of eigenvectors \( \{\phi_i\}^m_{i=1} \) of \( \tilde{S}_t \) corresponding to nonzero eigenvalues, where \( m = \text{rank}(S_b) \), \( \tilde{S}_t^+ \) denotes the pseudoinverse of \( \tilde{S}_t \). Construct matrix \( G = [\phi_1, \ldots, \phi_m] \).

4. Optional orthogonalization step: Compute the QR decomposition [12] of \( G \) as \( G = QR \), where \( Q \in \mathbb{R}^{d \times q} \) has orthonormal columns and \( R \in \mathbb{R}^{q \times q} \) is upper triangular. With this four-step procedure, the final transformation is either given by matrix \( G \) from step 3), if the optional orthogonalization step is not applied, or given by matrix \( Q \) from step 4) if the transformation matrix is required to be orthogonal. In this framework, different transfer functions \( \Phi \) in step 2) lead to different generalized LDA algorithms, as summarized in the following.

- In PCA+LDA, the intermediate dimensionality reduction stage by PCA keeps the top \( p \) eigenvalues of \( S_t \), thus it applies the following linear step function: \( \Phi(\lambda_i) = \lambda_i \), for \( 1 \leq i \leq p \), and \( \Phi(\lambda_i) = 0 \), for \( i > p \). The optional orthogonalization step is not employed in PCA+LDA.

- In RLDA, a regularization term is applied to \( S_t \) as \( S_t + \mu I_d \), for some \( \mu > 0 \). It corresponds to the use of the following transfer function: \( \Phi(\lambda_i) = \lambda_i + \mu \), for all \( i \). The optional orthogonalization step is not employed in RLDA.

- In ULDA, the optimal transformation consists of the top eigenvectors of \( S_t^+ S_b \) [32]. The corresponding transfer function is thus given by \( \Phi(\lambda_i) = \lambda_i \), for all \( i \). The same transfer function is used in OLDL. The difference between ULDA and OLDL is that OLDL requires the optional orthogonalization step while it is not necessary in ULDA.

- In OCM, the optimal transformation is given by the top eigenvectors of \( S_b \) [22]. The transfer function is thus given by \( \Phi(\lambda_i) = 1 \), for all \( i \). Because the eigenvectors of \( S_b \) form an orthonormal set, the optional orthogonalization step is not necessary in OCM.

In the following, we show that in RLDA, the regularization is only effective for nonzero eigenvalues, and the transfer function can be simplified further. More importantly, we show in
Section IV-B that based on this property, the matrix computations involved in RLDA can be simplified. Let

\[ S_t = U \text{diag}(\Sigma_t, 0) U^T \]

be the singular value decomposition (SVD) [12] of \( S_t \), where \( U \) is orthogonal and \( \Sigma_t \in \mathbb{R}^{d \times t} \) is diagonal and nonsingular with \( t = \text{rank}(S_t) \). Let

\[ U = (U_1, U_2) \]

be a partition of \( U \), such that \( U_1 \in \mathbb{R}^{d \times t} \) and \( U_2 \in \mathbb{R}^{d \times (d-t)} \). Because \( S_t = S_b + S_w \), the null space of \( S_t \) is a subset of the null space of \( S_b \). That is, \( S_b U_2 = 0 \). It follows that \( (S_t + \mu I_d)^{-1} S_b \) can be expressed as

\[
U \left( \begin{pmatrix} \Sigma_t & 0 \\ 0 & 0 \end{pmatrix} + \mu I_d \right)^{-1} U^T S_b U_1 U_1^T U^T
\]

\[
= U \left( \Sigma_t + \mu I_{d-t} \right)^{-1} \left( U_1^T S_b U_1 \right) U^T
\]

\[
= U_1 \left( \Sigma_t + \mu I_{d-t} \right)^{-1} U_1^T S_b U_1 U_1^T U^T. \tag{15}
\]

It is clear from (15) that the regularization is only effective for the nonzero eigenvalues in \( \Sigma_t \) and has no effect on the zero eigenvalues. Thus, we can apply the following transfer function for RLDA: \( \Phi(\lambda_i) = \lambda_i + \mu \), for all \( i = 1, \ldots, t \), and zero otherwise, where \( t = \text{rank}(S_t) \).

In NLDA [5], [6], the data is first projected onto the null space of \( S_w \), which is then followed by classical LDA. It is not clear which transfer function \( \Phi \) corresponds to the projection onto the null space of \( S_w \). In [35], the equivalence relationship between NLDA and OLDA was established under a mild condition

\[ C1 : \text{rank}(S_t) = \text{rank}(S_b) + \text{rank}(S_w) \tag{16} \]

which has been shown to hold for many high-dimensional data. Thus, for high-dimensional data, we can use the following transfer function for NLDA: \( \Phi(\lambda_i) = \lambda_i \), for all \( i \).

A. Analysis

The unified framework proposed above summarizes the commonalities and differences of various LDA-based algorithms. This unification of diverse algorithms into a common framework sheds light on the understanding of the key features of various algorithms as well as their relationships.

It follows from the proposed framework that ULDA is reduced to the OCM algorithm [22] when \( S_t \) is a multiple of the identity matrix. Recent studies on the geometric representation of high-dimensional and small sample size data [15] show that under mild conditions, the covariance matrix \( S_t \) tends to a scaled identity matrix when the data dimension \( d \) tends to infinity with the sample size \( n_l \) fixed. This implies that all the eigenvalues of \( S_t \) are the same. In other words, the data behave as if the underlying distribution were spherical. In this case, OCM is equivalent to ULDA. This partially explains the effectiveness of OCM when working on high-dimensional data.

When the reduced dimensionality \( p \) in the PCA stage of PCA+LDA is chosen to be the rank of \( S_t \), that is, the PCA stage keeps all the information, then the transfer functions for PCA+LDA and ULDA are identical, that is, PCA+LDA is equivalent to ULDA in this case. Moreover, the transfer function for RLDA equals the one for ULDA when \( \mu = 0 \). Thus, ULDA can be considered as a special case of both PCA+LDA and RLDA. In [37], a correlation-based noisy bases pruning method was proposed. It was shown that PCA+LDA corresponds to a special case of this method in which the principal eigenvectors are selected [37].

It follows from the above discussion that RLDA with \( \mu \) approaching zero, and PCA+LDA with \( p = \text{rank}(S_t) \), are essentially ULDA. Under condition \( C1 \) in (16), the transformation matrix of ULDA has been shown to lie in the null space of \( S_w \) [35], that is, \( G^T S_w = 0 \). In this case, it follows from (1) that \( G^T(x - c(i)) = 0 \), for all \( x \in X_i \), and \( G^T x = G^T c(i) \). This shows that the ULDA transformation maps all data points from the same class to a common vector, provided that condition \( C1 \) is satisfied. This leads to a perfect separation between different classes in the dimensionality-reduced space. However, it may also result in overfitting. A similar result has been shown in [5] when all classes in the data set have a common sample size. RLDA overcomes this limitation by choosing a nonzero regularization value \( \mu \), while PCA+LDA overcomes this limitation by setting \( p < \text{rank}(S_t) \).

The above analysis shows that the regularization in RLDA and the PCA dimensionality reduction in PCA+LDA are expected to alleviate the overfitting problem, provided that appropriate values for \( \mu \), the regularization parameter in RLDA, and \( p \), the dimension of the intermediate PCA-subspace in PCA+LDA can be estimated. Selecting an optimal value for a parameter such as \( p \) in PCA+LDA and \( \mu \) in RLDA from a given candidate set is called model selection [16]. Existing studies have focused on the estimation from a small candidate set, as it involves expensive matrix computations for each candidate value. However, a large candidate set is usually required to achieve satisfactory performance. Motivated by the special properties of various algorithms and their relationships in the unified framework, we propose model selection algorithms for PCA+LDA and RLDA in Section IV. Experimental results are presented in Section V to demonstrate the efficiency and effectiveness of the proposed algorithms.

IV. Efficient Model Selection

It follows from the analysis in the last section that setting \( \mu > 0 \) for RLDA and \( p < \text{rank}(S_t) \) for PCA+LDA can overcome the overfitting problem. In PCA+LDA, \( p \) is an integer in the range \([k, \text{rank}(S_t)]\) where \( k \) is the number of classes. By taking advantage of the special properties of PCA and LDA, we propose an efficient model selection algorithm for PCA+LDA in Section IV-A. The proposed algorithm can choose an optimal value in the range \([k, \text{rank}(S_t)]\) efficiently. For RLDA, there are an infinite number of choices for the value of \( \mu \), and it is usually estimated by cross validation. To achieve satisfactory performance in practice, a large set of candidate values are usually used. We show in Section IV-B that the matrix computations
involved in RLDA can be simplified so that the cross-validation procedure can be performed efficiently.

A. Model Selection for PCA+LDA

Let

$$H_t = U_t \Sigma V_t^T$$  (17)

be the skinny SVD of $H_t$, where $H_t$ is defined in (6), $U_t \in \mathbb{R}^{d \times t}$ is defined in (14), $V_t \in \mathbb{R}^{n \times t}$ has orthonormal columns, $\Sigma \in \mathbb{R}^{t \times t}$ is diagonal, and $t = \text{rank}(H_t)$. Then, it follows from (7) that

$$S_t = H_t H_t^T = U_t \Sigma_4 U_t^T$$

where $\Sigma_4 = \Sigma^2$. When $p = \text{rank}(S_t)$, PCA projects the original data onto the column space of $U_t$. From the definitions of $S_t$ and $S_b$ in (2) and (3) that in this dimensionality-reduced space by PCA, the total scatter matrix and between-class scatter matrix, denoted as $\tilde{S}_t$ and $\tilde{S}_b$, can be expressed as

$$\tilde{S}_t = U_t^T S_t U_t = \Sigma_t \tilde{S}_b = U_t^T S_b U_t.$$  (18)

It follows that performing classical LDA in this PCA-transformed space requires the diagonalization of the matrix $\Sigma_t^{-1} \tilde{S}_b = \Sigma_t^{-1} U_t^T S_b U_t$, which can be written as

$$\Sigma_t^{-1/2} \Sigma_t^{-1/2} U_t^T H_b \Sigma_t^{-1/2} U_t \Sigma_t^{-1/2} \Sigma_t^{-1/2}$$

because $\tilde{S}_b = H_b H_b^T$. Let $B = \Sigma_t^{-1/2} U_t^T H_b$ and $B = U_b \Sigma_b V_b^T$ be the SVD of $B$, then

$$\Sigma_t^{-1/2} \Sigma_t^{-1/2} U_t^T H_b \Sigma_t^{-1/2} U_t \Sigma_t^{-1/2} \Sigma_t^{-1/2}$$

$$= \Sigma_t^{-1/2} (\Sigma_t^{-1/2} U_t^T H_b (\Sigma_t^{-1/2} U_t^T H_b)^T) \Sigma_t^{-1/2}$$

$$= \Sigma_t^{-1/2} B B^T \Sigma_t^{-1/2}$$

$$= (\Sigma_t^{-1/2} U_b) \Sigma_b (\Sigma_t^{-1/2} U_b)^{-1}.$$  (19)

Thus, the matrix $\Sigma_t^{-1/2} U_b$ diagonalizes $\Sigma_t^{-1} \tilde{S}_b$. This leads to a two-step procedure for computing the eigenvectors of $\Sigma_t^{-1} \tilde{S}_b$ as follows:

- Compute the SVD of $H_t$ as

$$H_t = U_t \Sigma V_t^T.$$  

- Compute the SVD of $\Sigma_t^{-1/2} U_t^T H_b$ as

$$\Sigma_t^{-1/2} U_t^T H_b = U_b \Sigma_b V_b^T$$

where $\Sigma_4 = \Sigma^2$.

Then, the eigenvectors of $\Sigma_t^{-1} \tilde{S}_b$ corresponding to nonzero eigenvalues are given by the columns of $U_b \Sigma_b$. The key observations that underlie our efficient model selection algorithm are that the first step needs to be computed only once regardless of the number of values tried, and for the second step, the matrix $B = \Sigma_t^{-1/2} U_t^T H_b$ for small values of $p$ are submatrices of those for larger values. In particular, once the matrix $B$ for the maximum $p$, i.e., $\text{rank}(S_t)$, is computed, subsequent $B$’s can be obtained by removing rows of the original $B$ matrix incrementally. Therefore, except for $p = \text{rank}(S_t)$, all that the algorithm needs to do is to remove the last row of current matrix $B$ and compute the SVD of this matrix of decreasing size.

Let $\Omega = \{p_1, \ldots, p_m\}$ be the candidate set for values of $p$. In the PCA+LDA model selection algorithm, $v$-fold cross validation is applied, where the data is divided into $v$ subsets of (approximately) equal size. In the $i$th fold, the $i$th subset is held out for test and all other subsets are used in training. For each $p_j, j = 1, \ldots, m$, we compute the cross-validation accuracy $\text{Accu}(j)$, defined as the mean of the accuracies for all folds. The best $p$ value $p_*$ is the one with

$$j^* = \arg \max_j \text{Accu}(j).$$

The pseudocode for this PCA+LDA model selection algorithm is presented in Algorithm 1 in which the nearest-neighbor (NN) or nearest-centroid (NC); also called nearest mean in [11] classifiers can be used in the dimensionality-reduced space.

Algorithm 1: PCA+LDA model selection algorithm

1. For $i = 1 : v$ $v$-fold cross validation
2. Construct $A^i$ and $A_t^i$;  
   // $A^i$: training set
3. Construct $H_t$ and $H_b$ using $A_t^i$;  
4. Compute the SVD of $H_t$: $H_t = U_t \Sigma V_t^T$;  
5. $B = \Sigma_t^{-1/2} U_t^T H_b$;  
6. $A_t^i = \Sigma_t^{-1/2} U_t^T A_t^i; \quad A_t^i = \Sigma_t^{-1/2} U_t^T A_t^i$;  
7. For $j = \text{rank}(S_t) : k$  
   8. Compute the SVD: $B_j = U_b \Sigma_b V_b^T;  
   // B_j$: first $j$ rows of $B$;  
9. $A_L^j = U_b \Sigma_b^j V_b^T; \quad A_L^j = U_b \Sigma_b^j V_b^T$;  
   // $A_L^j$: first $j$ rows of $A_t$;  
10. $A_L^j$: first $j$ rows of $A_t$;  
11. Run 1NN on $(A_L^j, A_L^j)$ and compute the accuracy, denoted as $\text{Accu}(i, j)$;  
12. EndFor  
13. $\text{Accu}(j) \leftarrow (1/v) \sum_{i=1}^v \text{Accu}(i, j)$;  
14. $j^* \leftarrow \arg \max_j \text{Accu}(j)$;  
15. Output $p_{j^*}$ as the best parameter.

We analyze the complexity of the proposed PCA+LDA model selection algorithm. Line 4 in Algorithm 1 takes $O(dh^2)$ time for computing the SVD. Both lines 5 and 6 take $O(r^2d)$ time for the matrix multiplications. For the second stage, each choice of $p$ incurs different costs. The SVD computation in line...
8 takes $O(pk^2)$ time, assuming $p > k$. The projections of the training and test samples in line 9 take $O(pkn)$ time in total. For the classification step, the associated cost is $O(n^2k)$ when the NN classifier is used. Summing up all those costs, we get the total cost for the second step as

$$\sum_{j=k}^n (O(pk^2) + O(pkn) + O(n^2k)) = O(n^3k).$$

Note that the complexity of the second step does not depend on $d$, the data dimensionality. Therefore, for high-dimensional data where $d$ is larger than the sample size $n$, the second step in the PCA-LDA model selection algorithm is very efficient. This is confirmed by our experimental results in Section V. To reduce the time complexity of the second stage further, we propose to use the NC classifier, instead of the NN classifier in line 10. Hence, the cost of classification at each cross-validation step is reduced from $O(n^2k)$ to $O(n^2k)$. Experimental results in Section V show that the use of NC classifier often improves classification performance.

### B. Model Selection for RLDA

It has been shown in Section III that regularizing the total scatter matrix $S_t$ in RLDA is equivalent to regularizing the nonzero eigenvalues of $S_t$. That is, regularization has no effect on zero eigenvalues of $S_t$. We show in the following that this property has significant implications in designing an efficient RLDA model selection algorithm for small sample size problems.

Let

$$H_t = U_t\Sigma V_t^T$$

be the SVD of $H_t$ as in (17). Because regularizing the total scatter matrix $S_t$ in RLDA is equivalent to regularizing its nonzero eigenvalues, we have

$$(S_t + \mu I_d)^{-1}S_b = U((\Sigma_d,0) + \mu I_d)^{-1}U^T S_b = U_1(U_1^T S_b U_1)^{-1}U_1^T S_b,$$

where

$$H_t = U_1\Sigma V_1^T.$$

### Proposition 1

Let $y$ be an eigenvector of $U_1(\Sigma_d + \mu I_d)^{-1}U_1^T S_b$ corresponding to a nonzero eigenvalue $\eta$. Then $y = U_1 x$ for some $x$, where $x$ is an eigenvector of $(\Sigma_d + \mu I_d)^{-1}U_1^T S_b U_1$.

### Proof

Let $y$ be an eigenvector of $U_1(\Sigma_d + \mu I_d)^{-1}U_1^T S_b$ corresponding to a nonzero eigenvalue $\eta$. Then

$$y = \frac{1}{\eta}U_1(\Sigma_d + \mu I_d)^{-1}U_1^T S_b y = U_1 x$$

for some $x$.

Next, we show that $x$ is an eigenvector of $(\Sigma_d + \mu I_d)^{-1}U_1^T S_b U_1$. Multiplying both sides of the following equation by $U_1^T$:

$$U_1(\Sigma_d + \mu I_d)^{-1}U_1^T S_b y = \eta y$$

we have

$$(\Sigma_d + \mu I_d)^{-1}U_1^T S_b (U_1 x) = \eta U_1^T (U_1 x) = \eta x.$$
The complexity of the RLDA model selection algorithm can be analyzed as follows. Line 4 takes $O(n^2d)$ time for the SVD computation. Lines 5 and 6 take $O(dk^2)$ time for the matrix multiplication $1 \leq \tau \leq \text{rank}(S_k)$. For each choice of $\mu$, line 9 and 10 take $O(tk^2)$ time for the eigen-decomposition and matrix multiplication. Line 11 takes $O(k'tn)$ time for the matrix multiplication. The computation of the classification accuracy by INN in line 12 takes $O(n^2k)$ time. Thus, the total time complexity $T(m)$ for estimating the best parameter is

$$T(m) = O(v(n^2d + m d k + m (tk^2 + ktn + n^2k)))$$
$$= O(v(n^2d + m k^2))$$
$$= O(vn^2(d + mk)).$$

We can compare $T(m)$ with $T(1)$, where $m = 1$, and obtain

$$\frac{T(m)}{T(1)} \approx \frac{vn^2(d + mk)}{vn^2(d + k)} \approx 1 + \frac{mk}{d}.$$ 

For small sample size problems, where the number of classes $k$ is much smaller than the data dimensionality $d$, i.e., $k \ll d$, the overhead of estimating the optimal regularization value among a large set is small.

### V. EXPERIMENTAL EVALUATION

In this section, we perform experiments to evaluate the proposed theories and algorithms. Section V-A briefly summarizes the statistics of the data sets used in the experiments. We begin the experimental study by comparing the classification performance of RLDA, PCA+LDA, and ULDA on various data sets in Section V-B. We also report the results obtained by support vector machines (SVM) and correlation-based LDA (corrLDA) proposed in [37]. In Section V-C, we evaluate the effect of regularization in PCA+LDA and RLDA. Sections V-D and V-E provide experimental evidences regarding the efficiency and effectiveness of the model selection algorithms for PCA+LDA and RLDA, respectively. In Section V-F, we show that dimensionality reduction algorithm facilitates data visualization by projecting high-dimensional data onto 2-D planes. When evaluating the classification performance in the dimensionality-reduced space, we report the accuracies obtained by both NC and NN classifiers.

#### A. Data Sets

We use eight data sets in the experiments and their statistics are summarized in Table I. The data sets fall into four categories. re0 and re1 are two text document data sets that are derived from the Reuters-21578 text categorization test collection Distribution 1.0.1. ORL2 and AR [19] are two widely used face image data sets. 14_Tumors [24] and Brain tumor [23] are gene expression data sets. Fruitfly [14, 33] is the gene expression pattern image data set where the first three (correspond to stage ranges 1–3, 4–6, and 7–8) and six (correspond to stage ranges 1–3, 4–6, 7–8, 9–10, 11–12, and 13–16) classes are used in the experiments. They are denoted as Fruitfly(3) and Fruitfly(6), respectively.

### B. Performance Evaluation

The first six data sets listed in Table I are high-dimensional and they all have a small number of samples in each class. We observe that the $C'$ condition in (16) holds for all of the six data sets. Hence, ULDA will project all samples in the same class to a common point in this case. For each of the six data sets, we randomly partition the entire data set into training and test sets using the ratio 1:1 and the performance of five methods (RLDA, PCA+LDA, ULDA, corrLDA, and SVM) are recorded. The correlation interval for corrLDA and the $C'$ value for SVM are tuned using cross validation on the sets $[0.85, 0.87, 0.89, 0.91, 0.93, 0.95, 0.97]$ and $[0.001, 0.01, 0.1, 1, 10, 50, 100, 200, 500, 1000]$. Respectively. This entire process is repeated 30 times and Fig. 1 plots the results for each data set. For ease of comparison, the mean accuracies over 30 random splittings are also summarized in Table II.

From Fig. 1 and Table II, we can observe that ULDA achieves similar classification performance with PCA+LDA and RLDA on the six data sets. Compared to re0 and re1, the ORL, AR, 14_Tumors, and Brain tumor data sets have a smaller number of samples in each class (less than 20 data points). We can also observe from the results that SVM achieves similar performance with RLDA and PCA+LDA when the number of classes is small. However, when the number of classes is large, LDA-based algorithms tend to produce higher accuracies because they can handle multiclass problems naturally. It is interesting to observe that PCA+LDA outperforms corrLDA in our experiments. Because PCA+LDA is a special case of corrLDA, we expect that such difference in performance may be due to the lack of effective model selection strategy for corrLDA. The superior performance of PCA+LDA can also be explained by the fact that it is optimal with regard to the Fisher criterion [21, Th. 4].

To evaluate the relative performance of the LDA-based algorithms when there are an increasing number of samples in each class, we randomly partition the Fruitfly(3) and Fruitfly(6) data sets into training and test sets with increasing proportion of data in the training set. Each of the LDA-based algorithms is used to learn a mapping from the training data and the NN and NC classifiers are used to obtain the classification accuracy in the dimensionality-reduced space. This process is repeated 30 times and the results are plotted in Figs. 2 and 3. The mean accuracies over 30 randomizations are also summarized in Table III.
We can observe from Figs. 2 and 3 and Table III that as the size of the training set increases, the classification performance of RLDA and PCA+LDA improve steadily. On the other hand, the performance of ULDA decreases as the number of samples in each class increases. This shows that when there are relatively large number of samples in each class, ULDA will suffer from the overfitting problem and regularization can be used to prevent it. It can also be observed from Figs. 2 and 3 that as the training set size increases, the performance of RLDA and PCA+LDA tends to be more stable.

C. Effect of Regularization in RLDA and PCA+LDA

The performance studies in Section V-B show that the regularization employed in RLDA and PCA+LDA is effective to prevent the overfitting problem in ULDA. To examine this effect in detail, we visualize the samples after the projection by RLDA.
and PCA+LDA with different parameter settings in this section. In particular, we run RLDA with $\mu = (0, 100, 1000, 10000)$ and PCA+LDA with $p = (299, 180, 51, 15)$ on a training set of 300 images and apply the projection to a test set of 2405 images for the Fruitfly(3) data set. There are $k = 3$ stage ranges (classes) in the data set, and all images are projected onto a 2-D plane. In Figs. 4 and 5, we show the projection of the training images (top row) and a subset of test images (bottom row) for RLDA and PCA+LDA, respectively. We depict each image by the corresponding stage range (1, 2, and 3). We can observe from Figs. 4 and 5 that when $\mu = 0$ or $p = 299$, which correspond to the case where no regularization is applied, all training points from the same class are mapped to a common point, which leads to the perfect separation in the training set. However, the test data points are scattered around and the classification accuracy using NC classifier is about 63.86% only. Note that RLDA with $\mu = 0$ and PCA+LDA with $p = \text{rank}(S_e)$ are equivalent to ULDA. When the value of $\mu$ increases or the value of $p$ decreases, the diameter of each class in the training set increases, while the three classes in the test set are better separated. We apply the proposed model selection algorithms and the optimal values of $\mu$ and $p$ obtained are 1000 and 51 for RLDA and PCA+LDA, respectively. Under this optimal parameter values, RLDA and PCA+LDA achieve their respective highest accuracies. When the value of $\mu$ further increases or the value of $p$ decreases, the accuracy starts to go down. This experiment shows the effective-
Fig. 3. Comparison of classification accuracy (in percentage) of RLDA, PCA+LDA, ULDA, corrLDA, and SVM on the Fruitfly(6) data set as the proportion of samples in the training set increases from 3% to 15%.

TABLE III

<table>
<thead>
<tr>
<th>Data set</th>
<th>Fruitfly(3)</th>
<th>Fruitfly(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>3%</td>
<td>5%</td>
</tr>
<tr>
<td>Classifier</td>
<td>NC</td>
<td>NN</td>
</tr>
<tr>
<td>RLDA</td>
<td>84.67</td>
<td>84.36</td>
</tr>
<tr>
<td>PCA+LDA</td>
<td>84.14</td>
<td>80.22</td>
</tr>
<tr>
<td>ULDA</td>
<td>82.01</td>
<td>82.01</td>
</tr>
<tr>
<td>corrLDA</td>
<td>81.95</td>
<td>78.97</td>
</tr>
<tr>
<td>SVM</td>
<td>84.85</td>
<td>87.02</td>
</tr>
</tbody>
</table>

D. PCA+LDA Model Selection Algorithm

To examine how the proposed PCA+LDA model selection algorithm works in practice, we randomly partition the re0 data into training and test sets using the ratio of 1:1 and the training data are fed into the proposed PCA+LDA model selection algorithm for optimal model selection. The accuracies for all possible values of $p$ are also computed. Fig. 6 plots the changes of accuracy while $p$ varies on two random partitions of the re0 data. Ideally, the PCA+LDA model selection algorithm should uncover the peak on this curve. The red horizontal line indicates the performance achieved when the proposed PCA+LDA model selection algorithm is used to select $p$ with tenfold cross validation. We can observe from Fig. 6 that the proposed algorithm is effective in estimating the optimal value of $p$.

To evaluate the relative efficiencies of the two stages in the PCA+LDA model selection algorithm, we run the algorithm on the re0 and tr41 data sets and the time spent in each stage is recorded. Fig. 7 shows the running time (in seconds) for the two stages in the PCA+LDA model selection algorithm for each fold when tenfold cross validation is used. The results indicate...
that even though the second stage needs to be repeated once for each choice of the value for $p$ in cross validation, the time spent in this stage is still less than that of the first stage. Thus, the overhead of estimating the optimal value of $p$ among a large set of candidates is small. Note that the time complexity of the second stage of the PCA+LDA model selection algorithm does not depend on the dimensionality of data. Thus, the second stage is expected to be more efficient when the data dimensionality is high. This is confirmed by our experiments where the tr41 data set has a higher dimensionality than the re0 data set (7454 and 2887, respectively), and the second stage of the algorithm on tr41 takes less time than that on re0 relative to their first stages.

### E. RLDA Model Selection Algorithm

We present experimental results to demonstrate the efficiency of the proposed RLDA model selection algorithm. In particular, we show how the running time scales as the size of the candidate set for value of $\mu$ increases. We start with the candidate set...
Fig. 6. Change of accuracy for PCA+LDA with different $p$ values on the re0 data. The horizontal line indicates the performance achieved by the $p$ value estimated by the PCA+LDA model selection algorithm when tenfold cross validation is used. The two figures are from two random splittings of the entire data set.

Fig. 7. Running time comparison of the two stages of the PCA+LDA model selection algorithm in each fold when tenfold cross validation is used on the re0 (left) and Fruitfly(3) (right) data.

### TABLE IV

<table>
<thead>
<tr>
<th>Size</th>
<th>re0</th>
<th>re1</th>
<th>ORL</th>
<th>AR</th>
<th>Brain tumor</th>
<th>14_Tumors</th>
<th>Fruitfly(3)</th>
<th>Fruitfly(6)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5.33</td>
<td>14.78</td>
<td>30.34</td>
<td>68.73</td>
<td>1.45</td>
<td>31.55</td>
<td>262.45</td>
<td>359.67</td>
</tr>
<tr>
<td>2</td>
<td>5.33</td>
<td>14.89</td>
<td>30.50</td>
<td>69.23</td>
<td>1.50</td>
<td>31.72</td>
<td>263.00</td>
<td>360.16</td>
</tr>
<tr>
<td>4</td>
<td>5.36</td>
<td>14.92</td>
<td>30.83</td>
<td>69.27</td>
<td>1.48</td>
<td>31.84</td>
<td>263.34</td>
<td>362.73</td>
</tr>
<tr>
<td>8</td>
<td>5.39</td>
<td>15.03</td>
<td>31.44</td>
<td>71.08</td>
<td>1.50</td>
<td>32.06</td>
<td>264.27</td>
<td>365.23</td>
</tr>
<tr>
<td>16</td>
<td>5.42</td>
<td>15.25</td>
<td>32.66</td>
<td>75.11</td>
<td>1.55</td>
<td>32.56</td>
<td>266.11</td>
<td>367.77</td>
</tr>
<tr>
<td>32</td>
<td>5.67</td>
<td>15.66</td>
<td>34.89</td>
<td>82.72</td>
<td>1.63</td>
<td>33.64</td>
<td>271.63</td>
<td>376.66</td>
</tr>
<tr>
<td>64</td>
<td>6.00</td>
<td>16.50</td>
<td>39.86</td>
<td>98.23</td>
<td>1.75</td>
<td>35.78</td>
<td>277.41</td>
<td>392.97</td>
</tr>
<tr>
<td>128</td>
<td>6.83</td>
<td>18.33</td>
<td>49.72</td>
<td>129.22</td>
<td>2.02</td>
<td>40.36</td>
<td>297.13</td>
<td>425.69</td>
</tr>
<tr>
<td>256</td>
<td>8.56</td>
<td>22.06</td>
<td>69.02</td>
<td>191.61</td>
<td>2.56</td>
<td>49.45</td>
<td>339.16</td>
<td>492.73</td>
</tr>
<tr>
<td>512</td>
<td>12.22</td>
<td>28.33</td>
<td>107.91</td>
<td>314.63</td>
<td>3.63</td>
<td>66.33</td>
<td>418.81</td>
<td>629.22</td>
</tr>
<tr>
<td>1024</td>
<td>18.78</td>
<td>42.39</td>
<td>184.27</td>
<td>563.48</td>
<td>5.80</td>
<td>102.83</td>
<td>575.63</td>
<td>898.72</td>
</tr>
</tbody>
</table>

| $T(1024)/T(1)$ | 3.52 | 2.87 | 6.07 | 8.20 | 4.00 | 3.26 | 2.19 | 2.50 |

Running time (in second) comparison of the proposed RLDA model selection algorithm as the size of the candidate set for $p$ increases. For each data set, it is partitioned into training and test sets using the ratio of 1:1 and the time spent by the RLDA model selection algorithm is recorded. Table IV shows the change of time as the candidate set size doubles. The last row of Table IV also reports the ratio between the time taken by the algorithm when the candidate set sizes are 1024 and 1. It can be...
seen that though the candidate size increases by 1024 fold, the maximum increase in terms of running time is only 8.20 in the case of the AR data set. The increases in running time for most data sets are in the range [2, 3]. This shows that the proposed RLDA model selection algorithm can be applied to a large candidate set.

F. Dimensionality Reduction Algorithm for Data Visualization

One of the major advantages of algorithms based on dimensionality reduction is that they facilitate data visualization by projecting high-dimensional data onto low-dimensional spaces. We show in this experiment that visualization of high-dimensional data can help to identify outliers in the data. We use the fruit fly gene expression pattern image data Fruitfly(3) collected from the FlyExpress3 database in this experiment. These images are originally obtained from the Berkeley Drosophila Genome Project (BDGP)4 and literature. They are standardized manually to put the anterior and posterior at left and right, respectively, and dorsal and ventral at top and bottom, respectively. Images are also annotated by the view from which the digital photograph was taken. Such manual standardization sometimes mislabeled images in the database. For example, although we only collect images that are annotated with correct orientation and lateral view while obtaining the images, some mislabeled images have been introduced in our experiments. Thus, it is important to identify such mislabeled images so that the annotations can be corrected.

To visualize this high-dimensional image data, we project the training and test images onto 2-D planes using the PCA+LDA algorithm, and they are presented in Fig. 8. It is known that expression patterns in late stages are more complex, and manual annotation is prone to error in such cases. Hence, we choose to analyze the images in the stage range 7–8. We can observe from Fig. 8(b) that images in the stage range 7-8 overlap with those in stage ranges 1–3 and 4–6. We partition the distribution of images in stage range 7–8 into four regions so that the upper left and the bottom right regions each contain ten images. These images are distributed far away from other images in the same stage range, and thus they are candidates for detailed analysis.

The images in the upper left and the bottom right regions of Fig. 8(b) are shown in Figs. 9 and 10, respectively. It is clear that, in Fig. 9, the dorsal and ventral of images numbered 1, 2, and 6 are flipped while the anterior and posterior are flipped for image numbered 8.

3http://www.flyexpress.net/

4http://www.fruitfly.org
studied further. From the proposed unified framework we can see that all existing generalized LDA algorithms use simple transfer functions. Based on this unified framework, we plan to explore new LDA-based algorithms in the future by employing specific transfer functions. Some possible choices include the polynomial function $\Phi(\lambda) = \lambda^m$, for some positive integer $m$, and the exponential function $\Phi(\lambda) = e^{\beta \lambda}$, for some constant $\beta > 0$.

**ACKNOWLEDGMENT**

The authors would like to thank Dr. M. Zhu and Dr. A. M. Martinez for making their correlation-based LDA codes publicly available and their help on using them.

**REFERENCES**


Shuiwang Ji (S’08) is working towards the Ph.D. degree in the Department of Computer Science and Engineering, Arizona State University. He is also affiliated with the Center for Evolutionary Functional Genomics of the Biordesign Institute, Arizona State University. His research interests include machine learning, data mining, and bioinformatics.

Jieping Ye (S’04–M’05) received the Ph.D. degree in computer science from the University of Minnesota-Twin Cities in 2005. Currently, he is an Assistant Professor in the Department of Computer Science and Engineering, Arizona State University, Tempe. Since August 2005, he has been a core faculty member of the Center for Evolutionary Functional Genomics of the Biordesign Institute, Arizona State University. His research interests include machine learning, data mining, and bioinformatics. He has published extensively in these areas.

Dr. Ye was awarded the Guidant Fellowship in 2004–2005. In 2004, his paper on generalized low rank approximations of matrices won the outstanding student paper award at the 21st International Conference on Machine Learning. He is a member of the Association for Computing Machinery (ACM).