A Two-Stage Linear Discriminant Analysis via QR-Decomposition

Jieping Ye, *Student Member*, *IEEE*, and Qi Li, *Student Member*, *IEEE*

Abstract—Linear Discriminant Analysis (LDA) is a well-known method for feature extraction and dimension reduction. It has been used widely in many applications involving high-dimensional data, such as image and text classification. An intrinsic limitation of classical LDA is the so-called *singularity problems*; that is, it fails when all scatter matrices are singular. Many LDA extensions were proposed in the past to overcome the singularity problems. Among these extensions, PCA+LDA, a two-stage method, received relatively more attention. In PCA+LDA, the LDA stage is preceded by an intermediate dimension reduction stage using Principal Component Analysis (PCA). Most previous LDA extensions are computationally expensive, and not scalable, due to the use of Singular Value Decomposition or Generalized Singular Value Decomposition. In this paper, we propose a two-stage LDA method, namely LDA/QR, which aims to overcome the singularity problems of classical LDA, while achieving efficiency and scalability simultaneously. The key difference between LDA/QR and PCA+LDA lies in the first stage, where LDA/QR applies QR decomposition to a small matrix involving the class centroids, while PCA+LDA applies PCA to the total scatter matrix involving all training data points. We further justify the proposed algorithm by showing the relationship among LDA/QR and previous LDA methods. Extensive experiments on face images and text documents are presented to show the effectiveness of the proposed algorithm.

Index Terms—Linear discriminant analysis, dimension reduction, QR decomposition, classification.

1 Introduction

Linear Discriminant Analysis [6], [9] is a well-known method for feature extraction and dimension reduction. It has been used widely in many applications such as face recognition [2], [19], [21], [27], text classification [4], [11], [32], microarray data classification [7], etc. Classical LDA aims to find an optimal transformation by minimizing the within-class distance and maximizing the between-class distance simultaneously, thus achieving maximum discrimination. The optimal transformation is readily computed by applying the eigen-decomposition to the scatter matrices. An intrinsic limitation of classical LDA is that its objective function requires that one of the scatter matrices be nonsingular. For many applications, such as face recognition and text classification, all scatter matrices in question can be singular since the dimension, in general, exceeds the number of data points. This is known as the *singularity* or *undersampled* problem [16], [32].

In recent years, many approaches have been brought to bear on such high-dimensional, undersampled problems. We will review four important extensions of classical discriminant analysis, including pseudoinverse LDA [24], Regularized LDA [8], PCA+LDA [2], [27], and LDA/GSVD [11], [32]. The difference of these four extensions can be briefly described as follows: Pseudoinverse LDA applies pseudoinverse to deal with the singularity of matrices; Regularized LDA adds a scaled identity matrix to the scatter matrix so that the perturbed scatter matrix is positive definite and, hence, nonsingular; PCA+LDA applies an intermediate dimension reduction stage using PCA on the original data to obtain a more compact representation so that the singularity of the scatter matrix is decreased; and LDA/GSVD applies Generalized Singular Value Decomposition [31] to deal with the inversion of the scatter matrix. The common point of these LDA extensions is the use of Singular Value Decomposition (SVD) [10] or Generalized Singular Value Decomposition (GSVD) [31], which not only degrades the training efficiency but also makes them hard to scale to large data sets.

In this paper, we propose a two-stage LDA extension, namely, LDA/QR. The first stage of LDA/QR maximizes the separation between different classes by applying QR decomposition to a small size matrix. The distinct property of this stage is its low time/space complexity. It can be used independently as a dimension reduction algorithm. We name it pre-LDA/QR for convenience. The second stage of LDA/QR incorporates both between-class and within-class information by applying LDA to the "reduced" scatter matrices resulting from the first stage. Our theoretical analysis indicates that the computational complexity of LDA/QR is linear on the number of training data points as well as the number of dimensions. Unlike many LDA methods, LDA/QR scales to large data sets since it does not require the entire data in main memory.

With the (training) efficiency and scalability, LDA/QR is desirable in retrieval applications involving large, high-dimensional, and dynamic databases. (In real-life applications, databases can be extremely large and dynamic [15].) Scalability makes LDA/QR suitable in handling extremely

- *J. Ye is with the Department of Computer Science and Engineering, University of Minnesota—Twin Cities, 4-192 EE/CSCI Bldg., 200 Union St. S.E., Minneapolis, MN 55455. E-mail: jieping@cs.umn.edu.*
- *Q. Li is with the Department of Computer and Information Sciences, University of Delaware, Newark, DE 19716. E-mail: qili@cis.udel.edu.*

Manuscript received 13 May 2004; revised 9 Nov. 2004; accepted 15 Nov. 2004; published online 14 Apr. 2005.
Recommended for acceptance by L. Kuncheva.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0237-0504.

0162-8828/05/$20.00 © 2005 IEEE Published by the IEEE Computer Society

large databases, and training efficiency makes it superior in handling dynamic databases over the traditional LDA methods.

We further justify the proposed algorithm by showing the relationship among LDA/QR and other LDA methods. More specifically, LDA/QR is shown to be a special case of pseudoinverse LDA, where the pseudoinverse is applied to the between-class scatter matrix. We also show that both LDA/QR and PCA+LDA are approximations of LDA/GSVD. The main difference is that LDA/QR applies pre-LDA/QR before the LDA stage, while PCA+LDA applies PCA instead.

We have conducted extensive experiments to evaluate the proposed algorithm on various well-known data sets of both face images and text documents and compare it with other algorithms. Results have shown that the LDA/QR algorithm has low computational cost, while it achieves or approximates closely to the best accuracy that other LDA methods achieve. One interesting observation is that pre-LDA/QR followed by the classifier yields better accuracy than PCA followed by the classifier. (K-Nearest Neighbor [6] is used as the classifier in this paper.) The rationale behind this lies in the fact that pre-LDA/QR makes use of the class label information, while PCA is unsupervised. This partly explains why LDA/QR is competitive with PCA+LDA in classification.

The main contributions of this paper include:

- We propose a two-stage LDA method, namely, LDA/QR, which couples the QR decomposition with LDA for dimension reduction. LDA/QR has significantly lower costs in time and space than many LDA methods, which is shown both theoretically and empirically.
- We present a detailed theoretical analysis on the relationship among LDA/QR and other LDA methods. In particular, we show that LDA/QR is a special case of pseudoinverse LDA (with the pseudoinverse applied to the between-class scatter matrix), and both LDA/QR and PCA+LDA are approximations of LDA/GSVD.
- We have conducted extensive experiments on face images and text documents to evaluate the effectiveness of LDA/QR and compare it with other LDA methods and PCA.

The rest of this paper is organized as follows: Section 2 is on related work. Section 3 reviews classical LDA and its several extensions. The LDA/QR algorithm is presented in Section 4, where the relationship among LDA/QR and other LDA methods is also discussed. A comprehensive study on the performance of the LDA/QR algorithm is presented in Section 5. A further discussion on LDA/QR is given in Section 6. We conclude in Section 7 with a discussion of related future work.

2 Related Work

Principal Component Analysis (PCA), also known as Karhunen-Loeve transform (KLT), is one of the well-known methods for dimension reduction [14], [28], [30]. PCA is an orthogonal transformation of the coordinate system. The new coordinates are called *principal components*. It is often the case that a small number of principal components is sufficient to account for the main structure in the data. The principal components are readily computed by applying an eigen-decomposition to the covariance matrix. Turk and Pentland developed the *eigenface* technique in [30], which is the landmark of PCA entering appearance-based face recognition. Linear Discriminant Analysis (LDA) is another well-known method for dimension reduction [6], [9]. A comparative study of PCA and LDA can be found in [2], [21], [27].

The null space LDA [5] attempts to solve the small sample size problems directly. Here, the null space refers to the null space of the within-class scatter. More specifically, it has been observed that the null space of the within-class scatter contains useful discriminant information. The method in [5] works as follows: First, project the data onto the null space of the within-class scatter and, then, in the projected space, compute the transformation that maximizes the between-class scatter. The null space LDA is based on the eigen-decomposition of the (original) scatter matrices, which is hard to be scalable. Besides, it may ignore some useful information by considering the null space of the within-class scatter only. The Discriminative Common Vector method was recently proposed for face recognition [3], which addressed computational difficulties encountered in null space LDA. Yu and Yang [33] developed a direct method for LDA and claimed that the direct method was equivalent to PCA+LDA. They called the direct method "unified PCA+LDA" since there was no separate PCA stage. Detailed analysis of the algorithm was not provided.

The LDA/GSVD algorithm is shown to be a special case of pseudoinverse LDA, where the pseudoinverse is applied to the total scatter matrix [32]. Interestingly, the LDA/QR algorithm proposed in this paper is also a special case of pseudoinverse LDA, where the pseudoinverse is applied to the between-class scatter matrix instead. Detailed theoretical analysis on this equivalence will be presented in Section 4.3.

Efficient algorithms that combine discriminant analysis with tree classifiers were proposed in [12], [17]. The algorithm in [12] casted a classification task into a regression problem and applied doubly clustered subspace-based hierarchical discriminating regression (HDR) for image retrieval. In [17], a hierarchical technique was proposed to recursively decompose a k-class problem into $k-1$ two-class/binary problems. Both algorithms use clustering/partitioning techniques to handle the decomposition and build the hierarchical tree. The resulting tree in [17] is binary, while the one in [12] is usually not.

Discriminant analysis can also be studied in the nonlinear fashion, so-called kernel discriminant analysis. It is desirable if the data has weak linear separability. Our paper focuses on linear discriminant analysis. The interested readers can find more details on kernel discriminant analysis in [1], [20], [26].

TABLE 1
Notations

Notation	Description
N	number of training data points
n	number of dimensions
k	number of classes
A	data matrix
H_b	precursor of between-class scatter
H_w	precursor of within-class scatter
S_b	between-class scatter matrix
S_w	within-class scatter matrix
S_t	total scatter matrix
G	transformation matrix
ℓ	number of retained dimensions in LDA
A_i	data matrix of the i-th class
m_i	centroid of the i-th class
N_i	number of data points in the i-th class
m	global centroid of the training dataset
K	number of nearest neighbors in KNN
p	number of retained dimensions in PCA
σ	perturbation in Regularized LDA

3 An Overview of Linear Discriminant Analysis

In this section, we give a brief overview of classical LDA and its four extensions: pseudoinverse LDA, Regularized LDA, PCA+LDA, and LDA/GSVD. For convenience, we present in Table 1 the important notations used in the paper.

3.1 Classical LDA

Given a data matrix $A \in \mathbb{R}^{n\times N}$, we consider finding a linear transformation $G \in \mathbb{R}^{n\times \ell}$ that maps each column a_i of A, for $1 \le i \le N$, in the n-dimensional space to a vector y_i in the ℓ-dimensional space as $y_i = G^T a_i \in \mathbb{R}^{\ell}(\ell < n)$. Assume that the original data in A is partitioned into k classes as $A = [A_1, \cdots, A_k]$, where $A_i \in \mathbb{R}^{n\times N_i}$ contains data points from the ith class and $\sum_{i=1}^{k} N_i = N$. Classical LDA aims to find the optimal transformation G such that the class structure of the original high-dimensional space is preserved in the low-dimensional space.

In discriminant analysis, three scatter matrices, called *within-class*, *between-class*, and *total scatter* matrices, are defined as follows [9]:

$$S_b = \frac{1}{N}\sum_{i=1}^{k} N_i(m_i - m)(m_i - m)^T = H_b H_b^T, \tag{1}$$

$$S_w = \frac{1}{N}\sum_{i=1}^{k}\sum_{x\in A_i}(x - m_i)(x - m_i)^T = H_w H_w^T, \tag{2}$$

$$S_t = S_b + S_w, \tag{3}$$

where the precursors H_b and H_w of the between-class and within-class scatter matrices in (1) and (2) are

$$H_b = \frac{1}{\sqrt{N}}\left[\sqrt{N_1}(m_1 - m), \cdots, \sqrt{N_k}(m_k - m)\right], \tag{4}$$

$$H_w = \frac{1}{\sqrt{N}}\left[A_1 - m_1 \cdot e_1^T, \cdots, A_k - m_k \cdot e_k^T\right], \tag{5}$$

$e_i = (1, \cdots, 1)^T \in \mathbb{R}^{N_i}$, A_i is the data matrix of the ith class, m_i is the centroid of the ith class, and m is the global centroid of the training data set. It is worthwhile to note that the total scatter matrix S_t is equal to a multiple of the so-called *covariance matrix* in statistics.

The *traces* of the within-class and between-class scatter matrices can be computed as follows:

$$\mathrm{trace}(S_b) = \frac{1}{N}\sum_{i=1}^{k} N_i ||m_i - m||_2^2,$$

$$\mathrm{trace}(S_w) = \frac{1}{N}\sum_{i=1}^{k}\sum_{x\in A_i} ||x - m_i||_2^2.$$

Hence, $\mathrm{trace}(S_w)$ measures the closeness of the vectors within the classes, while $\mathrm{trace}(S_b)$ measures the separation between the classes.

In the low-dimensional space resulting from the linear transformation G, the within-class, between-class, and total scatter matrices become $S_b^L = G^T S_b G$, $S_w^L = G^T S_w G$, and $S_t^L = G^T S_t G$, respectively.

An optimal transformation G would maximize $\mathrm{trace}(S_b^L)$ and minimize $\mathrm{trace}(S_w^L)$. Common optimizations in classical LDA include [9]:

$$\max_G\left\{\mathrm{trace}((S_w^L)^{-1}S_b^L)\right\} \text{ and } \min_G\left\{\mathrm{trace}((S_b^L)^{-1}S_w^L)\right\}. \tag{6}$$

The optimization problems in (6) are equivalent to finding the generalized eigenvectors satisfying $S_b x = \lambda S_w x$, for $\lambda \neq 0$. The solution can be obtained by applying the eigen-decomposition to the matrix $S_w^{-1}S_b$, if S_w is nonsingular, or $S_b^{-1}S_w$, if S_b is nonsingular. It was shown in [9] that the solution to the optimization problem in (6) can also be obtained by computing the eigen-decomposition on the matrix $S_t^{-1}S_b$, assuming S_t is nonsingular. There are at most $k-1$ eigenvectors corresponding to nonzero eigenvalues since the rank of the matrix S_b is bounded from above by $k-1$. Therefore, the number of retained dimensions in classical LDA is at most $k-1$. A stable way to compute the eigen-decomposition is to apply SVD on the scatter matrices. Details can be found in [27].

3.2 Extensions of Classical LDA

Note that a limitation of classical LDA in many applications involving undersampled data is that at least one scatter matrix is nonsingular. Several extensions including pseudoinverse LDA, Regularized LDA, PCA+LDA, and

LDA/GSVD were proposed in the past to deal with the singularity problems as follows.

3.2.1 Pseudoinverse LDA

Pseudoinverse is commonly used to deal with the singularity of matrices. A natural extension of classical LDA, using the pseudoinverse, is to apply the eigen-decomposition to the matrix $S_b^+ S_w$, $S_w^+ S_b$, or $S_t^+ S_b$.

The pseudoinverse of a matrix can be computed by SVD [10]. More specifically, let $M = U\Sigma V^T$ be the SVD of M, where U and V have orthonormal columns and Σ is diagonal with positive diagonal entries, then the pseudoinverse of M can be computed as $M^+ = V\Sigma^{-1}U^T$. The following property of pseudoinverse is straightforward from its definition [10].

Proposition 1. *Let P and Q be orthogonal matrices and let M be any matrix with appropriate size. Then,*

$$(PMQ)^+ = Q^T M^+ P^T.$$

This proposition will be used in Section 4.3 to show the equivalence between LDA/QR and pseudoinverse LDA.

3.2.2 Regularized LDA

A simple way to deal with the singularity of S_w is to add a multiple of identity matrix to S_w, as $S_w + \sigma I_n$, for some $\sigma > 0$, where I_n is an identity matrix [8]. It is easy to check that $S_w + \sigma I_n$ is positive definite, hence nonsingular. This approach is called Regularized LDA (RLDA). A limitation of RLDA is that the optimal value of the parameter σ is difficult to determine. Cross-validation can be used for estimating the optimal σ [16].

3.2.3 PCA+LDA

A common way to deal with the singularity problems is to apply an intermediate dimension reduction stage, such as PCA, to reduce the dimension of the original data before classical LDA is applied. This is known as PCA+LDA. In this two-stage algorithm, the discriminant stage is preceded by a dimension reduction stage using PCA. It has received extensive study in face recognition [2], [27]. However, besides its expensive computation of SVD, the dimension reduction stage using PCA may potentially lose some useful information for discrimination.

3.2.4 LDA/GSVD

The LDA/GSVD algorithm in [11], [32] is a more recent approach. The inversion of the scatter matrix is avoided by the simultaneous diagonalization of the scatter matrices via the Generalized Singular Value Decomposition. Experiments in [11], [32] showed that the GSVD based method was competitive with other LDA methods on text classification. However, one limitation of this method is the expensive computation of GSVD on large data sets. It was shown in [11], [32] that the time complexity of LDA/GSVD is $O((N+k)^2 n)$, where N is the number of data points, n is the number of dimensions, and k is the number of classes.

4 LDA/QR: A Two-Stage Linear Discriminant Analysis

In this section, we propose an extension of classical LDA, namely, LDA/QR. This algorithm has two stages. The first stage maximizes the separation between different classes via QR decomposition [10]. This stage can be used independently as a dimension reduction algorithm. We name it pre-LDA/QR for convenience. The distinct property of pre-LDA/QR is the low time/space complexity. The second stage addresses the issue of within-class distance, while keeping low time/space complexity.

The first stage of LDA/QR aims to compute the optimal transformation matrix G that solves the following optimization problem:

$$G = \arg \max_{G^T G = I_\ell} \mathrm{trace}(G^T S_b G). \tag{7}$$

Note that this optimization problem only addresses the issue of maximizing between-class distance. The solution to (7) can be obtained through QR decomposition with column pivoting [10] on the precursor of between-class scatter matrix H_b in (4). More specifically, let $H_b = QR\Pi$ be the QR decomposition of H_b with column pivoting, where $Q \in \mathbb{R}^{n\times t}$ has orthonormal columns, $R \in \mathbb{R}^{t\times k}$ is upper triangular, $\Pi \in \mathbb{R}^{k\times k}$ is a permutation matrix, and $t = \mathrm{rank}(H_b)$; then, $G = QW$, for any orthogonal matrix $W \in \mathbb{R}^{t\times t}$, solves the optimization problem in (7), as stated in the following theorem.

Theorem 1. *Let $H_b = QR\Pi$ be the QR decomposition of H_b with column pivoting defined above. Then, $G = QW$, for any orthogonal $W \in \mathbb{R}^{t\times t}$, solves the optimization problem in (7).*

Proof. Let $\hat{Q} \in \mathbb{R}^{n\times(n-t)}$ be the matrix such that $P = [Q, \hat{Q}]$ is orthogonal, i.e., $PP^T = P^T P = I_n$. It follows that $S_b = H_b H_b^T = (QR\Pi)(\Pi^T R^T Q^T) = QRR^T Q^T = P\Sigma P^T$, where

$$\Sigma = \begin{pmatrix} RR^T & 0 \\ 0 & 0 \end{pmatrix}.$$

Hence, $\mathrm{trace}(G^T S_b G) = \mathrm{trace}(\tilde{G}^T \Sigma \tilde{G})$, where $\tilde{G} = P^T G$. Note that $\tilde{G}^T \tilde{G} = G^T P P^T G = G^T G = I_\ell$ since P is orthogonal and G has orthonormal columns. It follows that

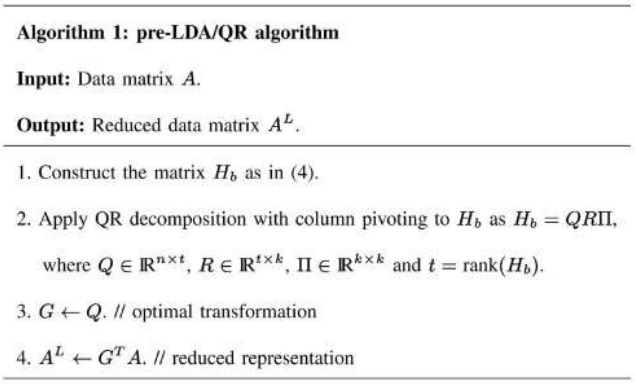
$$\mathrm{trace}(G^T S_b G) = \mathrm{trace}(\tilde{G}^T \Sigma \tilde{G}) \leq \mathrm{trace}(\Sigma) = \mathrm{trace}(RR^T),$$

where the inequality becomes equality if

$$\tilde{G} = \begin{pmatrix} W \\ 0 \end{pmatrix},$$

for any orthogonal $W \in \mathbb{R}^{t\times t}$. Hence, $G = P\tilde{G} = QW$, for any orthogonal W, solves the optimization problem in (7). This completes the proof of the theorem. □

In our implementation, we choose W to be the identity matrix for simplicity. The pseudocode for this algorithm is given in **Algorithm 1**. Note that the rank t of the matrix H_b is bounded from above by $k-1$. In practice, the k centroids in the data set are usually linearly independent. In this case, the number of retained dimensions is $t = k-1$.



```
Algorithm 1: pre-LDA/QR algorithm
Input: Data matrix $A$.
Output: Reduced data matrix $A^L$.
1. Construct the matrix $H_b$ as in (4).
2. Apply QR decomposition with column pivoting to $H_b$ as $H_b = QR\Pi$,
   where $Q \in \mathbb{R}^{n\times t}$, $R \in \mathbb{R}^{t\times k}$, $\Pi \in \mathbb{R}^{k\times k}$ and $t = \mathrm{rank}(H_b)$.
3. $G \leftarrow Q$. // optimal transformation
4. $A^L \leftarrow G^T A$. // reduced representation
```

The QR decomposition with column pivoting for computing the optimal transformation G takes $O(k^2n)$ time [10]. It then takes $O(knN)$ time to get the reduced representation A^L by $A^L = G^TA$, where each column in A (respectively, A^L) corresponds to a training data point in the original high-dimensional space (respectively, low-dimensional space). Hence, the total time to get the reduced representation in the first stage (pre-LDA/QR) is $O(knN)$. Note that the time complexity of pre-LDA/QR is much lower than many LDA methods. In summary, pre-LDA/QR algorithm gains its efficiency by ignoring the within-class information.

The second stage of LDA/QR refines the first stage by addressing the issue of within-class distance. It incorporates the within-class scatter information by applying a relaxation scheme to W (relaxing W from an orthogonal matrix to an arbitrary matrix). More specifically, we look for a transformation matrix G such that $G = QW$, for some W. (Note that W is not required to be orthogonal.) The original problem on computing G is equivalent to computing W. Since

$$G^TS_bG = W^T(Q^TS_bQ)W,$$
$$G^TS_wG = W^T(Q^TS_wQ)W,$$

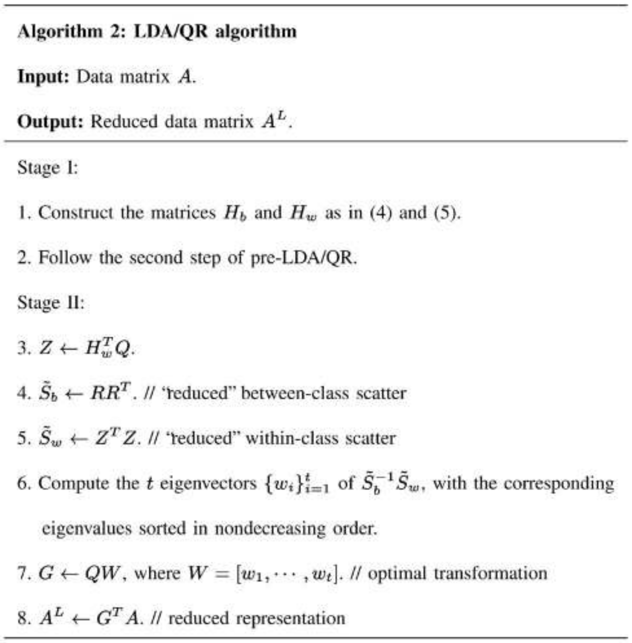
the original optimization problem on finding optimal G is equivalent to finding optimal W, with $\tilde{S}_b = Q^TS_bQ$ and $\tilde{S}_w = Q^TS_wQ$ as the "reduced" between-class and within-class scatter matrices, respectively.

The optimal W can be computed by solving the following optimization problem:

$$W = \arg\min_W \mathrm{trace}\Big((W^T\tilde{S}_bW)^{-1}(W^T\tilde{S}_wW)\Big). \tag{8}$$

Note that $\tilde{S}_b$ is nonsingular and has much smaller size than the original scatter matrix S_b.

The optimization problem in (8) can be solved using the similar method for classical LDA. That is, we compute optimal W by applying the eigen-decomposition to $\tilde{S}_b^{-1}\tilde{S}_w$. The pseudocode for this algorithm is given in **Algorithm 2**. Note that the eigenvalues are ordered in nondecreasing order in Line 6 of the LDA/QR algorithm, since the inversion is applied to the "reduced" between-class scatter $\tilde{S}_b$.



```
Algorithm 2: LDA/QR algorithm
Input: Data matrix $A$.
Output: Reduced data matrix $A^L$.
Stage I:
1. Construct the matrices $H_b$ and $H_w$ as in (4) and (5).
2. Follow the second step of pre-LDA/QR.
Stage II:
3. $Z \leftarrow H_w^TQ$.
4. $\tilde{S}_b \leftarrow RR^T$. // "reduced" between-class scatter
5. $\tilde{S}_w \leftarrow Z^TZ$. // "reduced" within-class scatter
6. Compute the $t$ eigenvectors $\{w_i\}_{i=1}^t$ of $\tilde{S}_b^{-1}\tilde{S}_w$, with the corresponding
   eigenvalues sorted in nondecreasing order.
7. $G \leftarrow QW$, where $W = [w_1, \cdots, w_t]$. // optimal transformation
8. $A^L \leftarrow G^TA$. // reduced representation
```

4.1 Time Complexity of LDA/QR Algorithm

The time complexity of the LDA/QR algorithm can be analyzed as follows: Line 2 takes $O(k^2n)$ time for QR decomposition with column pivoting [10]. Line 3 takes $O(Nnk)$ time for multiplication of two matrices. It takes $O(t^2k)$ time in Line 4, where $t = \mathrm{rank}(H_b)$ is less than or equal to $k-1$; hence, Line 4 takes $O(k^3)$. Similarly, the complexity for Line 5 is $O(k^2N)$. Line 6 computes the eigen-decomposition of a k by k matrix, hence takes $O(k^3)$ [10]. The matrix multiplication in Line 7 takes $O(nt^2) = O(nk^2)$. Finally, in Line 8, it takes $O(tnN) = O(knN)$ time for matrix multiplication.

Recall that the number of dimensions (n) and the total number of points (N) are usually much larger than the number of classes (k). Hence, the most expensive steps in the second stage of the LDA/QR algorithm are Lines 3 and 8, which take $O(Nnk)$. Therefore, the total (training) complexity of the LDA/QR algorithm is the same as pre-LDA/QR, i.e., linear on the number of data points, as well as the number of dimensions. In the test stage, the complexity of reducing the dimension of a new coming datum is $O(nk)$, which is the same as other LDA methods.

4.2 Scalability of LDA/QR Algorithm

Scalability of a dimension reduction algorithm is highly desirable for large data sets. Most algorithms, such as PCA, PCA+LDA, LDA/GSVD, and RLDA, require the entire data in main memory for the SVD or GSVD computation, and are thus not scalable. LDA/QR is highly scalable under the assumption that the number of classes is small enough such that all class centroids can reside in main memory. This is not a restrictive assumption for most applications. With this assumption, the scalability of LDA is mainly achieved by the incremental matrix computations in Lines 3 and 8. By incremental matrix computation, we refer to the processing of one data-stream at a time in matrix computation.

TABLE 2
Complexity Comparison: N Is the Number of Training Data Points, n Is the Number of Dimensions, and k Is the Number of Classes

Method	Time complexity	Space complexity
PCA	$O(N^2n)$	$O(nN)$
PCA+LDA	$O(N^2n)$	$O(nN)$
pre-LDA/QR	$O(nNk)$	$O(nk)$
LDA/QR	$O(nNk)$	$O(nk)$
LDA/GSVD	$O((N+k)^2n)$	$O(nN)$
RLDA	$O(N^2n)$	$O(nN)$

Next, we give more detailed analysis on the scalability of the LDA/QR algorithm. It first computes the k class centroids m_i and the global centroid m by scanning the whole data set once and stores them in a temporary array. By subtracting each local centroid m_i by the global centroid m, we keep the matrix H_b in main memory. QR decomposition with column pivoting is then applied to H_b. As mentioned above, we can process one data-stream at a time to do the matrix multiplication in Line 3 by scanning the whole data set one more time. Lines 4, 6, and 7 only involve small matrices. Recall that $\tilde{S}_b, \tilde{S}_w$ and $\tilde{S}_b^{-1}\tilde{S}_w$ are all of size $t \times t$, where t, the rank of H_b, is smaller than the number of classes. Line 5 involves the matrix $Z \in \mathbb{R}^{N\times k}$. The multiplication Z^TZ can be computed efficiently if the matrix Z can be kept in main memory. Otherwise, the matrix multiplication can be done incrementally by the following observation: If

$$Z = \begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix},$$

then

$$Z^TZ = \left(Z_1^T, Z_2^T\right)\begin{pmatrix} Z_1 \\ Z_2 \end{pmatrix} = Z_1^TZ_1 + Z_2^TZ_2.$$

Similar to Line 3, the computation in Line 8 can be done by scanning the whole data set one more time.

Table 2 lists the time/space complexity of the dimension reduction algorithms discussed in this paper. We can observe that LDA/QR and pre-LDA/QR are distinctly more efficient than other methods.

4.3 Equivalence between LDA/QR and Pseudoinverse LDA

As discussed in Section 3.1, classical LDA computes the optimal transformation matrix by computing the eigen-decomposition on $S_w^{-1}S_b$, if S_w is nonsingular, or $S_b^{-1}S_w$, if S_b is nonsingular. A natural extension of classical LDA for singular scatter matrices is to compute the eigen-decomposition on $S_b^+S_w$ or $S_w^+S_b$, as discussed in Section 3.2.1.

The eigen-decomposition on $S_b^+S_w$ is closely related to the LDA/QR algorithm, as stated in the following theorem.

Theorem 2. *Let G be the optimal transformation matrix obtained from the LDA/QR algorithm. Then, the columns of G are eigenvectors of $S_b^+S_w$ corresponding to the nonzero eigenvalues.*

Proof. Let x be an eigenvector of $S_b^+S_w$ corresponding to the nonzero eigenvalue λ, i.e., $S_b^+S_wx = \lambda x$. Let

$$H_b = [Q, \tilde{Q}]\begin{pmatrix} R \\ 0 \end{pmatrix}\Pi$$

be the QR decomposition of H_b with column pivoting, where $[Q, \tilde{Q}] \in \mathbb{R}^{n\times n}$ is orthogonal, R is upper triangular, and Π is a permutation matrix. It follows from Proposition 1 that

$$S_b^+ = (H_bH_b^T)^+ = [Q, \tilde{Q}]\begin{pmatrix} (RR^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix}[Q, \tilde{Q}]^T.$$

Hence,

$$S_b^+S_wx = [Q, \tilde{Q}]\begin{pmatrix} (RR^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix}[Q, \tilde{Q}]^TH_wH_w^Tx = \lambda x,$$

or

$$\begin{pmatrix} (RR^T)^{-1} & 0 \\ 0 & 0 \end{pmatrix}\begin{pmatrix} Q^T \\ \tilde{Q}^T \end{pmatrix}H_wH_w^Tx = \lambda\begin{pmatrix} Q^T \\ \tilde{Q}^T \end{pmatrix}x.$$

It follows that

$$\begin{pmatrix} (RR^T)^{-1} \\ 0 \end{pmatrix}Q^TH_wH_w^T[Q, \tilde{Q}]\begin{pmatrix} Q^Tx \\ \tilde{Q}^Tx \end{pmatrix} = \lambda\begin{pmatrix} Q^Tx \\ \tilde{Q}^Tx \end{pmatrix}.$$

It is easy to check that $\tilde{Q}^Tx = 0$. Hence,

$$(RR^T)^{-1}(Q^TH_wH_w^TQ)Q^Tx = \lambda Q^Tx,$$

which implies that Q^Tx is an eigenvector of

$$(RR^T)^{-1}Q^TH_wH_w^TQ,$$

the same matrix used in Line 6 of the LDA/QR algorithm. This completes the proof of the theorem. □

Theorem 2 shows the relationship between LDA/QR and pseudoinverse LDA. More specifically, LDA/QR is shown to be a special case of pseudoinverse LDA with the pseudoinverse applied to the between-class scatter matrix S_b, and the LDA/QR algorithm proposed in this paper provides an efficient way for computing the eigen-decomposition of $S_b^+S_w$.

4.4 LDA/QR and PCA+LDA: Approximations of LDA/GSVD

In [32], the equivalence between pseudoinverse LDA and LDA/GSVD was presented. More specifically, it was shown that the solution to LDA/GSVD can be obtained by computing the eigen-decomposition on the matrix $S_t^+S_b$. That is, LDA/GSVD is a special case of pseudoinverse LDA, where the pseudoinverse is applied to the total scatter matrix S_t. It is then straightforward to show that LDA/GSVD can be decomposed into two stages: an intermediate dimension reduction stage using the eigen-decomposition on S_t followed by LDA. More specifically, let $S_t = U\Sigma U^T$ be the eigen-decomposition, where $U \in \mathbb{R}^{n\times r}$ has orthonormal columns, $\Sigma \in \mathbb{R}^{r\times r}$ is diagonal with positive diagonal entries, and $r = \mathrm{rank}(S_t)$. Then, the "reduced" scatter matrices after the first stage are

$$\tilde{S}_w = (U^T H_w)(U^T H_w)^T,$$
$$\tilde{S}_b = (U^T H_b)(U^T H_b)^T,$$
$$\tilde{S}_t = U^T S_t U = \Sigma.$$

Since $\tilde{S}_t$ is nonsingular, the second stage computes the optimal transformation by applying the eigen-decomposition to $(\tilde{S}_t)^{-1}\tilde{S}_b$, as discussed in Section 3.1.

The decomposition of LDA/GSVD into two stages naturally leads to its connection with LDA/QR and PCA+LDA since both LDA/QR and PCA+LDA apply two-stage dimension reduction. In the first stage of LDA/QR, the eigen-decomposition is applied to S_b. Note that S_b is an approximation of S_t, when all data points from the same class are replaced by the centroid. This can be observed from the following equality:

$$S_t - S_b = S_w = \frac{1}{N}\sum_{i=1}^{k}\sum_{x \in A_i}(x - m_i)(x - m_i)^T,$$

where m_i is the centroid of the ith class and A_i contains the data points from the ith class. Hence, LDA/QR can be considered as an approximation of LDA/GSVD, where the eigen-decomposition is applied to S_b, instead of S_t, in its first stage.

In the first stage of PCA+LDA, the total scatter $S_t = U\Sigma U^T$ is approximated by $S_t \approx U_p \Sigma_p U_p^T$, where U_p consists of the first p columns of U, and Σ_p is the pth principal submatrix of Σ. Hence, PCA+LDA can also be considered as an approximation of LDA/GSVD, where S_t is approximated by its optimal rank-p approximation in its first stage. It is worthwhile to note that when p is equal to the rank of S_t, PCA+LDA is equivalent to LDA/GSVD.

Therefore, both LDA/QR and PCA+LDA are approximations of LDA/GSVD. The main difference is that LDA/QR applies pre-LDA/QR in the first stage, while PCA+LDA applies PCA in the first stage. However, LDA/QR is much more efficient than PCA+LDA, as shown in Table 2. Detailed comparative studies are given in the next section.

5 Experiments

We evaluate the effectiveness of the LDA/QR algorithm in this section. It contains four parts. The data sets for our performance study are presented in Section 5.1. In Section 5.2, we compare LDA/QR with other LDA methods, including PCA+LDA, LDA/GSVD, and RLDA, in terms of classification accuracy. (We also report the result on PCA, in unsupervised implementation, for each data set.) We use the K-Nearest Neighbor (KNN) algorithm [6] as the classifier. The classification accuracies are estimated by 10-fold cross-validation [6].

In Section 5.3, we study the efficiency of the LDA/QR algorithm and compare it with other competing algorithms. Our hardware configuration is 1.80GHz CPU and 1G RAM. An important observation from this study is that LDA/QR and pre-LDA/QR have distinctly less computational time than other dimension reduction algorithms. Finally, we study the scalability of the LDA/QR algorithm in Section 5.4.

5.1 Data Sets

We have three types of data sets for our performance evaluation: 1) synthetic data (2D and 50D), 2) face images (PIX, ORL, and AR), and 3) text documents (tr41, re0, and re1), as shown below:

- 2D synthetic data set. It contains 200 data points from two classes (each has 100 points) in the 2D space. Data in the first class is generated from a Gaussian whose mean is $[0, 0]$, and data in the second class is generated from a mixture of two Gaussians: The first one has 30 points with the mean $[2, 2] - [\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]\mu$, and the second one has 70 points with the mean $[2, 2] + [\frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2}]\mu$ (for some μ). All these Gaussians have covariance $0.5I_2$, where $I_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix.
- 50D synthetic data set. It contains 450 data points from three classes in the 50D space. The three classes contain 100, 150, and 200 points, respectively, which are generated from Gaussians of different means: $\mathbf{0}_{1\times 50}$, $\mathbf{1}_{1\times 50}+0.2[\mathbf{1}_{1\times 25},-\mathbf{1}_{1\times 25}]$, and $\mathbf{1}_{1\times 50}-0.2[\mathbf{1}_{1\times 25},-\mathbf{1}_{1\times 25}]$, where $\mathbf{0}_{1\times u}$ (respectively, $\mathbf{1}_{1\times u}$) is a vector consisting of u zeros (respectively, ones). All these Gaussians have covariance $0.5I_{50}$, where $I_{50} \in \mathbb{R}^{50\times 50}$ is the identity matrix.
- ORL face data set.[1] It contains 400 face images of 40 individuals. The image size is 92×112. The face images are perfectly centralized. The major challenge on this data set is the variation of the face pose. There is no lighting variation with minimal facial expression variations and no occlusion. We use the whole image as an instance (i.e., the dimension of an instance is $92 \times 112 = 10,304$).
- PIX face data set.[2] It contains 300 face images of 30 individuals. The image size is 512×512. We subsample the images with sample step 5×5, and the dimension of each instance is reduced to $100 \times 100 = 10,000$.
- AR face data set.[3] It is a large face image data set [22]. The instance of each face may contain significantly large areas of occlusion, due to the presence of sun glasses and scarves. The existence of occlusion dramatically increases the within-class variations of AR face image data. In this study, we use a subset of AR containing 1,638 face images of 126 individuals. Its image size is 768×576. We first crop the image from the row 100 to 500 and the column 200 to 550, and then subsample the cropped images with sample step 4×4. The dimension of each instance is reduced to $101 \times 88 = 8,888$.
- tr41 text data set. It is derived from the TREC-5, TREC-6, and TREC-7 collections [29].
- re0 and re1 text data sets. They are derived from *Reuters-21578* text categorization test collection Distribution 1.0 [18].

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

[2] http://peipa.essex.ac.uk/ipa/pix/faces/manchester/test-hard/.

[3] http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html.

TABLE 3
Statistics of Our Real Test Data Sets

Dataset	Size	# of dimensions	# of classes
ORL	400	10304	40
PIX	300	10000	30
AR	1638	8888	126
tr41	878	7454	10
re0	1504	2886	13
re1	1657	3758	25

For all three text data sets, we use a stop-list to remove common words, and the words are stemmed using Porter's suffix-stripping algorithm [23]. Moreover, any term that occurs in fewer than two documents is eliminated as in [34]. We use the *tf-idf* weighting scheme [25], [34] for all the documents. Finally, all document vectors are normalized to have unit length. More information on the three text data sets can be found in [34].

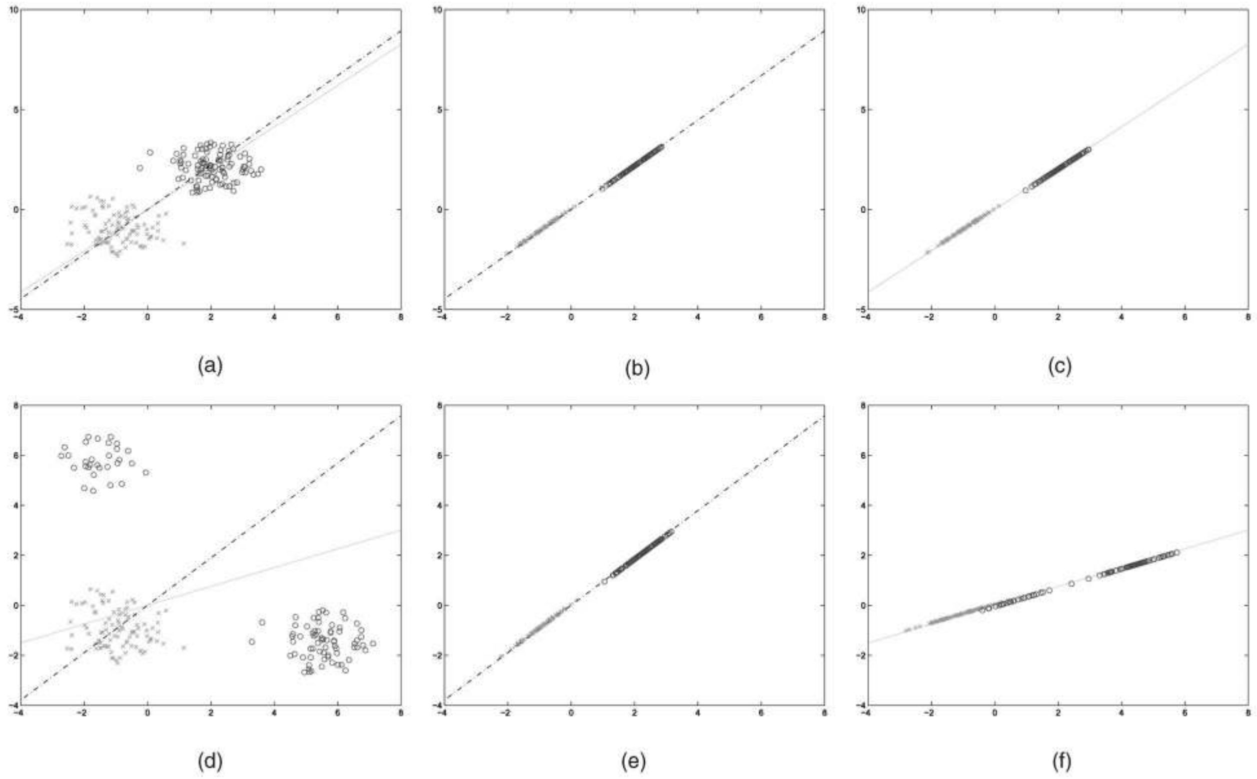
Table 3 summarizes the statistics of our real test data sets.

The two synthetic data sets are used to visually evaluate the performance of LDA/QR in comparison with LDA/GSVD. Note that LDA/GSVD is equivalent to classical LDA with nonsingular scatter matrices, which is the case for our two synthetic data sets.

Visualization of LDA/QR. We first consider the 2D synthetic data set with $\mu = 0$ (Fig. 1a). LDA/GSVD and LDA/QR are applied to the 2D synthetic data set, respectively. The two projection lines of LDA/GSVD and LDA/QR are shown in Fig. 1a. Figs. 1b and 1c) show the projections of all points onto the two projection lines of LDA/GSVD and LDA/QR, respectively. We can observe that the transformation (projection line) of LDA/QR is close to that of LDA/GSVD and, thus, the separability of the projected data in both cases are also similar to each other (see Figs. 1b and 1c).

Next, we consider the 50D synthetic data set. Its 2D projection via LDA/GSVD and LDA/QR are shown in Fig. 2. (Recall that we have three classes here.) We can observe the similar separability of data in these two cases.

Centroid sensitivity of LDA/QR. Note that the first stage of LDA/QR is essentially the orthogonalization of the k class centroids. If the centroids do not configure the decision boundary well, LDA/QR tends to fail, as shown in Figs. 1d, 1e, and 1f, where we increase μ from 0 to 5 (i.e., the distance between the centroids of two Gaussians in the second class increases). In this scenario, there is a significant disagreement between LDA/QR and LDA/GSVD, as shown in the different angles of two transformations/lines in Fig. 1d. We can observe from Fig. 1f that the projections of two classes via LDA/QR overlap. On the other hand, LDA/GSVD considers the maximum discrimination between different classes and is able to separate these two classes, as shown in Fig. 1e.



(a) (b) (c)

(d) (e) (f)

Fig. 1. Visualization of 2D synthetic data and its projections via LDA/GSVD (dashdot lines) and LDA/QR (solid lines). The first and second rows correspond to $\mu = 0$ and $\mu = 5$, respectively. (b), (e) LDA/GSVD, (c), (f) LDA/QR.

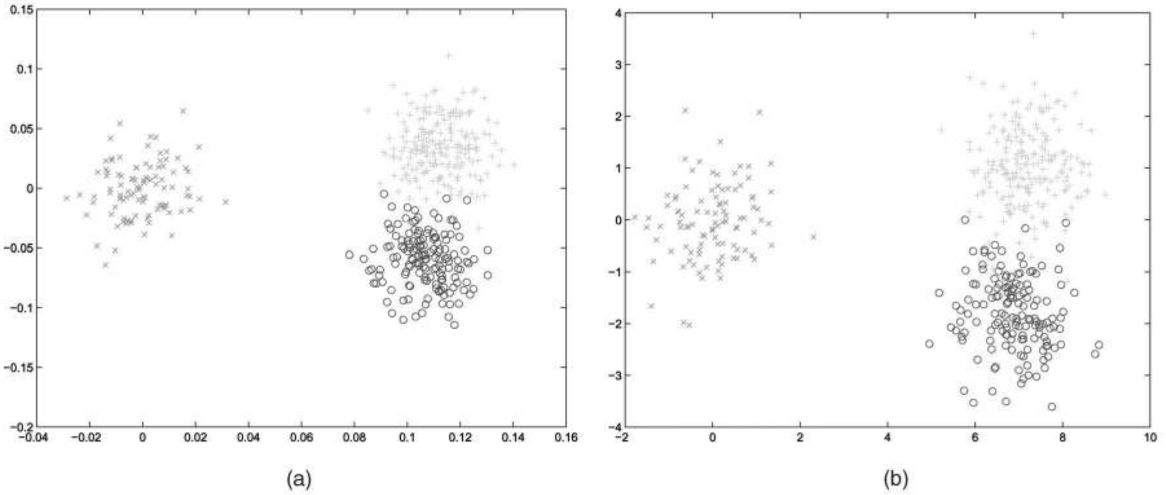


Fig. 2. Projections of 50D synthetic data onto the 2D planes via LDA/GSVD and LDA/QR. (a) LDA/GSVD, (b) LDA/QR.

TABLE 4
Classification Accuracies (%) of Different Dimension Reduction Algorithms on the Three Face Image Data Sets: ORL, PIX, and AR

Data set	KNN	PCA	PCA+LDA	Pre-LDA/QR	LDA/QR	LDA/GSVD	RLDA
ORL	1	97.25(1.42)	95.00(3.12)	97.75(1.84)	98.25(1.69)	94.00(3.76)	94.75(3.62)
	3	94.50(3.07)	94.75(3.43)	95.25(2.75)	98.00(2.58)	94.00(3.76)	94.75(3.62)
	5	92.25(2.99)	95.50(2.58)	94.75(2.49)	98.25(2.06)	94.00(3.76)	94.75(3.62)
	10	81.25(5.03)	93.75(3.58)	87.00(5.11)	96.75(2.37)	94.00(3.76)	94.50(3.50)
	15	70.75(5.41)	93.00(3.29)	82.75(5.33)	94.75(3.81)	94.00(3.76)	94.50(3.50)
PIX	1	97.67(3.16)	97.69(3.06)	97.67(3.16)	98.67(2.33)	97.33(2.63)	98.00(2.33)
	3	95.33(4.77)	97.69(3.06)	95.67(3.87)	97.00(2.92)	97.33(2.63)	98.00(2.33)
	5	93.67(5.54)	97.69(3.06)	95.33(4.22)	97.33(3.06)	97.33(2.63)	98.00(2.33)
	10	88.00(5.92)	95.67(3.06)	91.00(6.49)	95.33(5.26)	97.33(2.63)	98.00(2.33)
	15	82.67(5.84)	94.00(3.06)	86.67(7.37)	95.33(4.77)	97.33(2.63)	97.67(3.16)
AR	1	65.30(2.63)	92.45(1.22)	68.59(2.38)	92.24(1.26)	92.60(1.16)	93.24(1.04)
	3	59.05(2.10)	90.72(1.17)	62.75(1.47)	90.63(1.63)	92.60(1.16)	93.24(1.04)
	5	57.49(2.04)	88.50(1.17)	62.23(1.47)	89.53(1.63)	92.60(1.16)	93.24(1.04)
	10	44.70(2.49)	85.63(1.84)	59.63(2.77)	86.93(2.44)	92.60(1.16)	92.12(1.46)
	15	52.69(3.20)	84.49(2.65)	55.75(2.76)	85.95(2.88)	92.60(1.16)	92.12(1.46)

The mean and standard deviation (in parenthesis) of accuracies from ten runs are shown.

5.2 Classification Accuracy

In this experiment, we evaluate the LDA/QR algorithm in terms of classification accuracy and compare it with other algorithms, including PCA, PCA+LDA, Pre-LDA/QR, LDA/GSVD, and RLDA. The relevant parameters are as follows: $p = 100$ principal components in PCA and the PCA stage of PCA+LDA (except the AR data set, where $p = 150$) and $\sigma = 0.5$ in RLDA. For LDA algorithms, the output dimension is $k - 1$, where k is the number of classes in the data set, as the k centroids in all data sets are linearly independent.

Table 4 shows the classification accuracy results of different dimension reduction algorithms on three face image data sets: ORL, PIX, and AR. The most interesting result lies in the classification accuracy results on the AR data set. We observe that LDA/QR, PCA+LDA, LDA/GSVD, and RLDA distinctly outperform the other two dimension reduction algorithms. Recall that the images in the AR data set contain large areas of occlusion whose direct consequence is the large within-class variation of each individual. The effort of minimizing the within-class variation achieves distinct success in this situation. Neither

TABLE 5
Classification Accuracies (%) of Different Dimension Reduction Algorithms on the Three Text Document Data Sets: tr41, re0, and re1

Data set	KNN	PCA	PCA+LDA	Pre-LDA/QR	LDA/QR	LDA/GSVD	RLDA
tr41	1	94.28(3.01)	94.65(2.21)	95.32(2.06)	95.33(2.60)	95.67(1.86)	95.10(2.65)
	10	92.47(3.63)	95.75(3.45)	94.98(2.76)	95.21(2.79)	95.67(1.86)	95.33(2.72)
	15	91.68(3.67)	96.35(2.76)	94.98(3.01)	95.21(2.68)	95.21(1.69)	94.64(3.28)
	30	86.55(3.11)	94.27(3.08)	92.59(3.88)	92.94(3.51)	94.19(2.96)	92.59(3.69)
	50	82.57(4.14)	92.14(3.34)	90.66(3.78)	91.34(4.02)	91.12(2.51)	91.12(3.86)
re0	1	82.96(2.21)	85.42(2.97)	83.85(2.46)	83.56(2.03)	85.67(2.32)	84.30(2.59)
	10	80.88(2.97)	85.57(2.75)	84.75(2.39)	84.82(2.54)	85.67(2.49)	85.42(2.51)
	15	80.80(3.70)	85.05(2.95)	85.34(2.18)	84.97(2.19)	85.12(2.35)	84.82(2.15)
	30	78.42(2.98)	85.13(2.21)	84.52(2.35)	84.00(2.80)	84.69(2.19)	84.82(2.11)
	50	76.04(3.17)	85.93(3.06)	83.63(2.96)	83.11(2.88)	84.72(2.84)	83.56(2.71)
re1	1	81.05(2.63)	84.43(2.52)	85.70(2.68)	85.09(2.35)	85.02(2.28)	85.40(4.06)
	10	82.07(1.71)	84.12(3.51)	86.12(3.05)	85.57(2.22)	86.23(3.28)	84.91(4.08)
	15	84.85(2.26)	83.18(3.29)	85.46(2.86)	84.37(3.19)	84.35(2.22)	84.13(3.52)
	30	83.64(3.17)	81.98(3.75)	82.92(3.18)	82.68(4.11)	83.31(3.27)	81.59(3.88)
	50	80.31(3.38)	79.03(3.72)	80.32(3.89)	80.57(2.93)	80.65(3.63)	77.37(3.44)

The mean and standard deviation (in parenthesis) of accuracies from ten runs are shown.

PCA nor pre-LDA/QR has the effort in minimizing the within-class variation, which predicts their poor performance in this situation.

Besides the major observation mentioned above, other important observations on image data sets include:

- KNN with $K = 1$ usually performs the best by all algorithms on all three image data sets. Except LDA/GSVD and RLDA, there is a clear trend of decrease in accuracy for each data set as K increases.
- On ORL and PIX, the best accuracies are around 99 percent. Several algorithms can achieve this accuracy. This is mainly due to the relatively small within-class variations in these data. Recall that ORL face images contain small pose variations, and PIX face images contain facial expression variations only.

Next, let us shift the performance study to the text data. Table 5 shows the classification accuracy results on three text data sets: tr41, re0, and re1. The main observation on the text data is that pre-LDA/QR becomes competitive with other LDA algorithms. This may be related to the fact that text data sets have relatively small within-class variation.

Besides the main observation, two important observations on text data are:

- The accuracy of different LDA methods keeps increasing up to $K = 10$. This phenomenon does not occur on image data. This is mainly due to the large number of instances contained in each class of text data. For example, each class in tr41 has 90 instances in average.
- We observe that the best accuracy on tr41 is around 96 percent, which is distinctly higher than the best accuracy on the other two (86 percent). (Recall that the dimension, 7,454, of the first data set, is much higher than those of the other two, which are around 3,000.)

5.3 Efficiency

In this experiment, we study the efficiency of the algorithms, measured by the CPU time (in log scale). The results are summarized in Fig. 3. We observe that, except pre-LDA/QR, the computational time of LDA/QR is distinctly less than others. Since the number of principal components used in PCA is usually small, ($p = 150$ for AR and $p = 100$ for other data sets), the computational times of PCA and PCA+LDA are close to each other.

5.4 Scalability

We study two aspects of scalability in this section: the scalability with respect to the number of original/input dimensions and the scalability with respect to the number of training data points. We analyze the scalability using the AR image data set. The results for PCA+LDA are also shown for comparison. Following the analysis in [13], we use the total response time for generating the optimal transformation as the parameter for measurement.

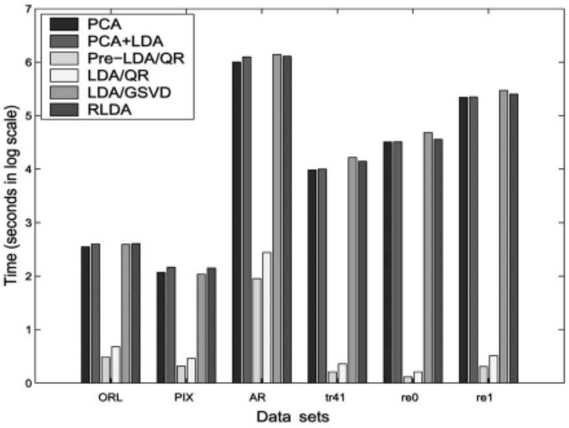
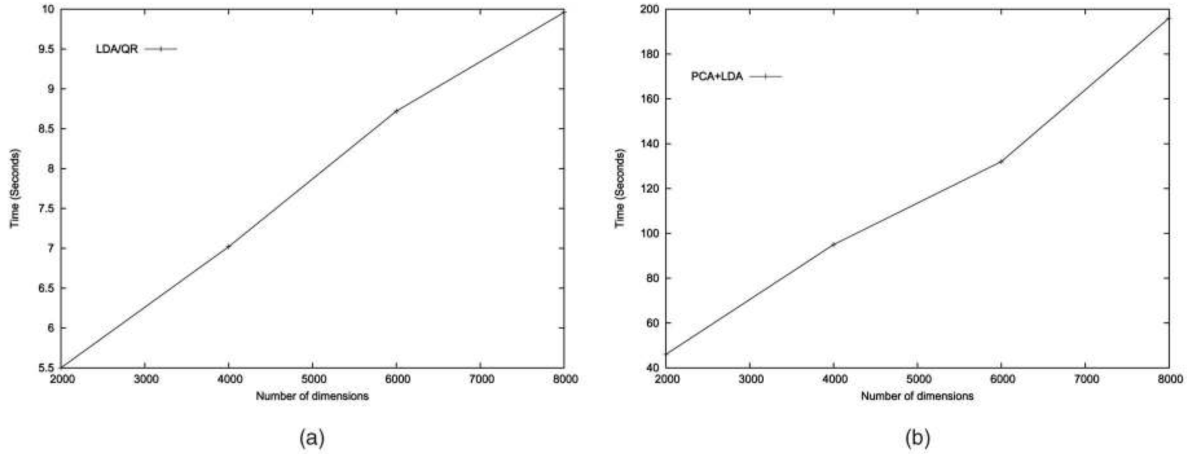


Fig. 3. Comparison of different dimension reduction algorithms on the CPU time (measured in seconds in log scale) in computing the reduced representations.

In the first experiment, we consider the case where the whole data matrix is kept in main memory. This is the case for our AR data set. We fix the number of training data points and vary the number of original/input dimensions from 2,000 to 8,000 by removing the remaining dimensions in the original full-dimensional space. The results are shown in Fig. 4, where the horizontal axis denotes the number of original/input dimensions and the vertical axis denotes the total response time. Here, the time for reading the data matrix is omitted since the whole data set is scanned once only. Fig. 4 shows that the total response times of both LDA/QR and PCA+LDA are linear on the number of dimensions for a fixed number of training data points. However, the increasing rate of LDA/QR is much lower than that of PCA+LDA.

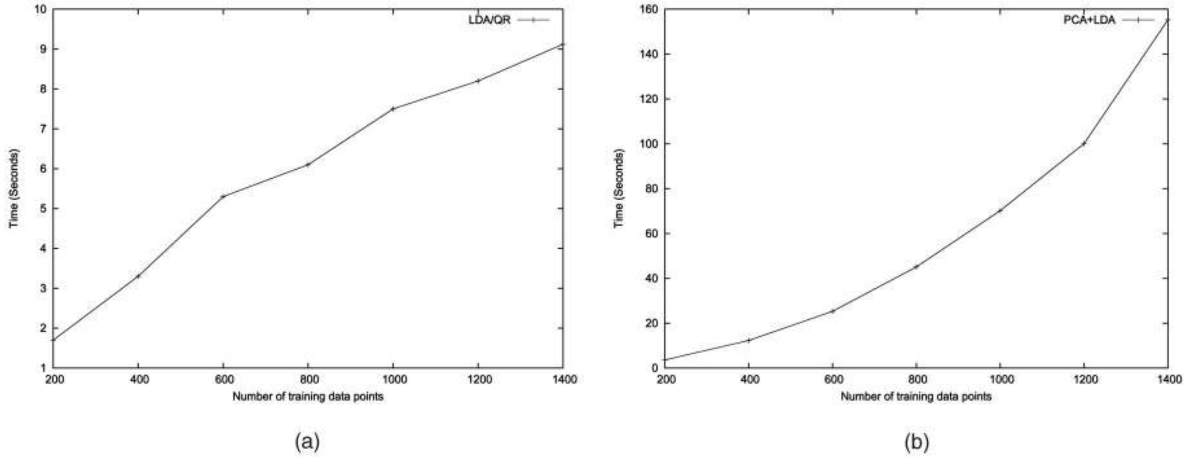
Next, we fix the number of original/input dimensions and vary the number of training data points from 200 to 1,600. The results are shown in Fig. 5. We can observe that the total response time of LDA/QR is still linear on the number of training data points, whereas the total response time of PCA+LDA is quadratic on the number of training data points. These results confirm the theoretical complexity estimation in Table 2.

In the second experiment, we simulate the scalability of the LDA/QR algorithm by reading 400 data points each time. (Note that PCA+LDA is not scalable in this case.) The results are shown in Fig. 6, where the horizontal axis



(a) (b)

Fig. 4. Scalability of LDA/QR and PCA+LDA with respect to the number of original/input dimensions, using the AR data set. The horizontal axis is the number of dimensions, and the vertical axis is the total response time. (a) LDA/QR, (b) PCA+LDA.



(a) (b)

Fig. 5. Scalability of LDA/QR and PCA+LDA with respect to the number of training data points, using the AR data set. The horizontal axis is the number of training data points, and the vertical axis is the total response time. (a) LDA/QR, (b) PCA+LDA.

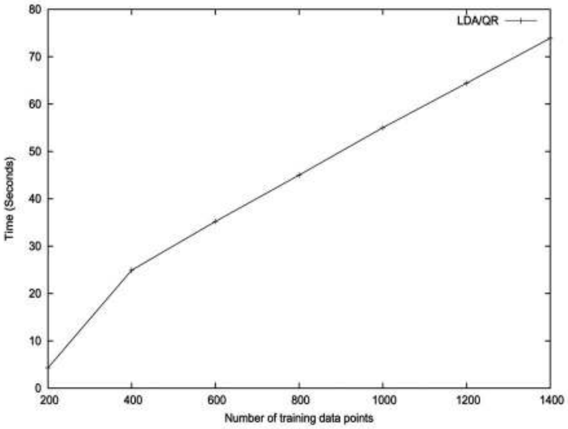


Fig. 6. Simulation of scalability of LDA/QR with respect to the number of training data points, using the AR data set. The horizontal axis is the number of training data points, and the vertical axis is the total response time.

denotes the number of training data points and the vertical axis denotes the total response time. It is clear from Fig. 6 that the total response time of LDA/QR is linear on the number of training data points, except when the number of data points reaches 400, where there is a jump. Note that we read 400 data points each time, which is much fewer than the total number of data points in the AR data set. Hence, the whole data matrix gets scanned multiple times in the LDA/QR algorithm (see the scalability analysis in Section 5.4).

6 Discussion

6.1 LDA/QR versus Pre-LDA/QR

From the perspective of linear algebra, the solution to pre-LDA/QR is a special case of the LDA/QR algorithm when the within-class scatter matrix S_w is set to be an identity matrix. Or equivalently, LDA/QR can be considered as an extension of pre-LDA/QR by incorporating the within-class information. When the within-class variation in a data set is large, the pre-LDA/QR algorithm is not guaranteed to perform well. LDA/QR overcomes this limitation by incorporating the within-class information at the second stage. This has been justified by the result on the AR data set in Section 5.2.

6.2 LDA/QR versus PCA+LDA

As discussed earlier, both LDA/QR and PCA+LDA apply an intermediate dimension reduction stage before the LDA stage. The main difference is that pre-LDA/QR is applied in LDA/QR, in contrast to PCA in PCA+LDA. Extensive experiments in Section 5.2 show that pre-LDA/QR outperforms PCA, which partly explains why LDA/QR is competitive with PCA+LDA. The superiority of pre-LDA/QR over PCA may be related to the fact that class label information is used in pre-LDA/QR, while PCA is unsupervised. Another interesting observation here is that, even though pre-LDA/QR outperforms PCA, when combined with LDA, PCA+LDA is competitive with LDA/QR. However, when large data sets are involved and efficiency is considered to be an important factor, LDA/QR is preferred, due to its lower time and space complexities compared to PCA+LDA (see Section 5.3).

7 Conclusions and Future Work

In this paper, we propose an extension of discriminant analysis, namely, LDA/QR, which is highly efficient and scalable. It is the QR decomposition that contributes to the efficiency and scalability of the LDA/QR algorithm, which is not only shown by our theoretical analysis, but also strongly supported by our empirical results.

The proposed algorithm is closely related to other LDA methods. More specifically, LDA/QR is shown to be a special case of pseudoinverse LDA with the pseudoinverse applied to the between-class scatter matrix. We also show that both LDA/QR and PCA+LDA are approximations of LDA/GSVD. The main difference is that LDA/QR applies pre-LDA/QR before the LDA stage, while PCA+LDA applies PCA instead.

Our experiments on face image and text data have shown that the accuracy achieved by the LDA/QR algorithm is competitive with the ones achieved by other LDA algorithms. Among all experiments, the results on the AR image data set (that contains large within-class variation) justify the effort of LDA/QR in minimizing the within-class distance in its second stage.

With efficiency and scalability, LDA/QR is promising in applications involving extremely high-dimensional data, such as video, which is one of our future work.

Acknowledgments

The authors would like to thank the associate editor and reviewers for helpful comments that greatly improved the paper. This research is sponsored, in part, by the Army High Performance Computing Research Center under the auspices of the Department of the Army, USA, Army Research Laboratory cooperative agreement number DAAD19-01-2-0014, the content of which does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred. The research of J. Ye is also supported by fellowships from Guidant Corporation and from the Department of Computer Science and Engineering at the University of Minnesota.

References

[1] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," *Neural Computation,* vol. 12, no. 10, pp. 2385-2404, 2000.

[2] 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,* vol. 19, no. 7, pp. 711-720, July 1997.

[3] H. Cevikalp, M. Neamtu, M. Wilkes, and A. Barkana, "Discriminative Common Vectors for Face Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 1, pp. 4-13, Jan. 2005.

[4] S. Chakrabarti, S. Roy, and M. Soundalgekar, "Fast and Accurate Text Classification via Multiple Linear Discriminant Projections," *Very Large Databases J.,* vol. 12, no. 2, pp. 170-185, 2003.

[5] L.F. Chen, H.Y.M. Liao, J.C. Lin, M.D. Kao, and G.J. Yu, "A New LDA-Based Face Recognition System which Can Solve the Small Sample Size Problem," *Pattern Recognition,* vol. 33, no. 10, pp. 1713-1726, 2000.

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

[7] S. Dudoit, J. Fridlyand, and T.P. Speed, "Comparison of Discrimination Methods for the Classification of Tumors Using Gene Expression Data," *J. Am. Statistical Assoc.*, vol. 97, no. 457, pp. 77-87, 2002.

[8] J.H. Friedman, "Regularized Discriminant Analysis," *J. Am. Statistical Assoc.*, vol. 84, no. 405, pp. 165-175, 1989.

[9] K. Fukunaga, *Introduction to Statistical Pattern Classification.* San Diego, Calif.: Academic Press, 1990.

[10] G.H. Golub and C.F. Van Loan, *Matrix Computations,* third ed. The Johns Hopkins Univ. Press, 1996.

[11] P. Howland, M. Jeon, and H. Park, "Structure Preserving Dimension Reduction for Clustered Text Data Based on the Generalized Singular Value Decomposition," *SIAM J. Matrix Analysis and Applications,* vol. 25, no. 1, pp. 165-179, 2003.

[12] W.-S. Hwang and J. Weng, "Hierarchical Discriminant Regression," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 11, pp. 1277-1293, Nov. 2000.

[13] H. Jin, B.C. Ooi, H.T. Shen, C. Yu, and A.Y. Zhou, "An Adaptive and Efficient Dimensionality Reduction Algorithm for High-Dimensionality Indexing," *Proc. Int'l Conf. Data Eng.*, pp. 87-98, 2003.

[14] I.T. Jolliffe, *Principal Component Analysis.* Springer-Verlag, 1986.

[15] K.V.R. Kanth, D. Agrawal, A.E. Abbadi, and A. Singh, "Dimensionality Reduction for Similarity Searching in Dynamic Databases," *Computer Vision and Image Understanding: CVIU,* vol. 75, nos. 1-2, pp. 59-72, 1999.

[16] 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,* vol. 44, pp. 101-115, 1995.

[17] S. Kumar, J. Ghosh, and M.M. Crawford, "Hierarchical Fusion of Multiple Classifiers for Hyperspectral Data Analysis," *Pattern Analysis and Applications,* vol. 5, no. 2, pp. 210-220, 2002.

[18] D.D. Lewis, "Reuters-21578 Text Categorization Test Collection Distribution 1.0," http://www.research.att.com/~lewis, 1999.

[19] C. Liu and H. Wechsler, "Enhanced Fisher Linear Discriminant Models for Face Recognition," *Proc. Int'l Conf. Pattern Recognition,* pp. 1368-1372, 1998.

[20] J. Lu, K.N. Plataniotis, and A.N. Venetsanopoulos, "Face Recognition Using Kernel Direct Discriminant Analysis Algorithms," *IEEE Trans. Neural Networks,* vol. 14, no. 1, pp. 117-126, 2003.

[21] A. Martinez and A. Kak, "PCA versus LDA," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, pp. 228-233, 2001.

[22] A.M. Martinez and R. Benavente, "The AR Face Database," Technical Report No. 24, CVC, 1998.

[23] M.F. Porter, "An Algorithm for Suffix Stripping," *Program,* vol. 14, no. 3, pp. 130-137, 1980.

[24] S. Raudys and R.P.W. Duin, "On Expected Classification Error of the Fisher Linear Classifier with Pseudoinverse Covariance Matrix," *Pattern Recognition Letters,* vol. 19, nos. 5-6, pp. 385-392, 1998.

[25] G. Salton, *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer.* Addison-Wesley, 1989.

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

[27] D.L. Swets and J. Weng, "Using Discriminant Eigenfeatures for Image Retrieval," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 8, pp. 831-836, Aug. 1996.

[28] F.D.L. Torre and M. Black, "Robust Principal Component Analysis for Computer Vision," *Proc. Int'l Conf. Computer Vision,* vol. I, pp. 362-369, 2001.

[29] TREC, *Proc. Text Retrieval Conf.*, http://trec.nist.gov, 1999.

[30] M. Turk and A. Pentland, "Eigenfaces for Recognition," *J. Cognitive Neuroscience,* vol. 3, pp. 71-86, 1991.

[31] C.F. Van Loan, "Generalizing the Singular Value Decomposition," *SIAM J. Numerical Analysis,* vol. 13, pp. 76-83, 1976.

[32] J. Ye, R. Janardan, C.H. Park, and H. Park, "An Optimization Criterion for Generalized Discriminant Analysis on Undersampled Problems," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 8, pp. 982-994, Aug. 2004.

[33] H. Yu and J. Yang, "A Direct LDA Algorithm for High-Dimensional Data with Applications to Face Recognition," *Pattern Recognition,* vol. 34, pp. 2067-2070, 2001.

[34] Y. Zhao and G. Karypis, "Empirical and Theoretical Comparisons of Selected Criterion Functions for Document Clustering," *Machine Learning,* vol. 55, no. 3, pp. 311-331, 2004.

Jieping Ye received the BS degree in mathematics from Fudan University, Shanghai, China, in 1997. He is currently a PhD student at the Department of Computer Science and Engineering at the University of Minnesota. He 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. His research interests include data mining, machine learning, pattern recognition, bioinformatics, and geometric modeling. He is a student member of the IEEE and ACM.

Qi Li received the BS degree from the Department of Mathematics at Zhongshan University, China, in 1993, and a master's degree from the Department of Computer Science at the University of Rochester in 2002. He is currently a PhD candidate in the Department of Computer and Information Sciences at the University of Delaware. His current research interests include pattern recognition, data mining, and machine learning. He is a student member of the IEEE.