Fast Linear Discriminant Analysis using QR Decomposition and Regularization

Haesun Park, Barry L. Drake, Sangmin Lee, Cheong Hee Park

College of Computing, Georgia Institute of Technology, 266 Ferst Drive, Atlanta, GA 30332, U.S.A. (hpark@cc.gatech.edu, bldrake@cc.gatech.edu, min0528@cc.gatech.edu) The work of the first three authors was supported in part by the National Science Foundation grants ACI-0305543 and CCF-0621889. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

Dept. of Computer Science and Engineering, Chungnam National University, 220 Gung-dong, Yuseong-gu, Daejeon, 305-763, Korea (cheonghee@cnu.ac.kr). The work of this author was supported by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD)(KRF-2006-331-D00510).

Abstract

Linear Discriminant Analysis (LDA) is among the most optimal dimension reduction methods for classification, which provides a high degree of class separability for numerous applications from science and engineering. However, problems arise with this classical method when one or both of the scatter matrices is singular. Singular scatter matrices are not unusual in many applications, especially for high-dimensional data. For high-dimensional undersampled and oversampled problems, the classical LDA requires modification in order to solve a wider range of problems. In recent work the generalized singular value decomposition (GSVD) has been shown to mitigate the issue of singular scatter matrices, and a new algorithm, LDA/GSVD, has been shown to be very robust for many applications in machine learning. However, the GSVD inherently has a considerable computational overhead. In this paper, we propose fast algorithms based on the QR decomposition and regularization that solve the LDA/GSVD computational bottleneck. In addition, we present fast algorithms for classical LDA and regularized LDA utilizing the framework based on LDA/GSVD and preprocessing by the Cholesky decomposition. Experimental results are presented that demonstrate substantial speedup in all of classical LDA, regularized LDA, and LDA/GSVD algorithms without any sacrifice in classification performance for a wide range of machine learning applications.

Index Terms

Dimension reduction, Linear Discriminant Analysis, Regularization, QR decomposition.

I. Introduction

Dimension reduction is an important preprocessing step which is encountered in many applications such as data mining, pattern recognition and scientific visualization. Discovering intrinsic data structure embedded in high dimensional data can give a low dimensional representation preserving essential information in the original data. Among several traditional methods, Linear Discriminant Analysis (LDA) [5] has been known to be one of the most optimal dimension reduction methods for classification. LDA aims to find an optimal linear transformation that maximizes the class separability. However, in undersampled problems, where the number of data samples is smaller than the dimension of the data space, it is difficult to apply classical LDA due to the singularity of the scatter matrices caused by high dimensionality. In order to make the LDA applicable in a wider range of applications, several generalizations of the LDA have been proposed. Among them, a method utilizing the generalized singular value decomposition

(GSVD), called LDA/GSVD, has been proposed recently [11], [12] and applied successfully in various application areas such as text categorization and face recognition. However, one disadvantage of the classical LDA and its variations for underdetermined systems is their high computational costs.

In this paper, we address these issues by proposing fast algorithms for classical LDA and generalizations of LDA such as regularized LDA [4] and LDA/GSVD by taking advantage of QR decomposition preprocessing and regularization in the framework of the LDA/GSVD algorithm.

Given a data matrix $A \in \mathbb{R}^{m\times n}$, where n columns of A represent n data items in an m dimensional space, the problem we consider is that of finding a linear transformation $G^T \in \mathbb{R}^{l\times m}$ that maps a vector x in the m dimensional space to a vector y in the l dimensional space, where l is an integer with $l \ll m$, to be determined, based on A :

$$G^T : x \in \mathbb{R}^{m\times 1} \rightarrow y \in \mathbb{R}^{l\times 1}. \tag{1}$$

Of special interest in this paper is the case when the data set is already clustered. Our goal is to find a dimension reducing linear transformation such that the cluster separability of the full dimensional data matrix A is preserved in the reduced dimensional space. For this purpose, we first need to define a measure of cluster quality. When the cluster quality is high, a clustering result has a tight within-cluster relationship while the between-cluster relationship is remote. In order to quantify this, in discriminant analysis [5], [21], within-cluster, between-cluster, and total scatter matrices must be defined. Throughout this paper, for simplicity of discussion, we will assume that the given data matrix $A \in \mathbb{R}^{m\times n}$ is partitioned into k clusters as

$$A = [A_1 \quad A_2 \quad \cdots \quad A_k] \quad \text{where} \quad A_i \in \mathbb{R}^{m\times n_i}, \quad \sum_{i=1}^{k} n_i = n,$$

N_i denotes the set of column indexes that belong to the cluster i, $c^{(i)}$ the centroid of each cluster, and c the global centroid.

The rest of the paper is organized as followed. In Section II, a brief review of linear discriminant analysis is presented and the formulation of LDA/GSVD is motivated. In Section III, the generalized singular value decomposition (GSVD) due to Paige and Saunders [15], which provides a foundation for the fast algorithms we present in this paper, is reviewed. Then the fast LDA algorithms designed based on the LDA/GSVD framework for undersampled problems utilizing the QR decomposition preprocessing and regularization, as well as for oversampled problems utilizing the Cholesky decomposition preprocessing [8], are presented in Section IV. Finally, some substantial experimental results are presented in Section V which illustrates more than an order of magnitude speedup of the proposed algorithm on many data sets from text categorization and face recognition.

II. Dimension Reduction by LDA

Linear Discriminant Analysis (LDA) is a well known method in the pattern recognition community for classification and dimension reduction. LDA maximizes the *conceptual* ratio of the between-cluster scatter (variance) versus the within-cluster scatter of the data. Here we provide a brief overview of the basic ideas. For a more in-depth treatment please see [6], [7].

Using the notations introduced in the previous section, the within-cluster scatter matrix S_w, between-cluster scatter matrix S_b, and the total (or mixture) scatter matrix S_t are defined as

$$S_w = \sum_{i=1}^{k} \sum_{j \in N_i} (a_j - c^{(i)})(a_j - c^{(i)})^T, \tag{2}$$

$$S_b = \sum_{i=1}^{k} \sum_{j \in N_i} (c^{(i)} - c)(c^{(i)} - c)^T, \tag{3}$$

and

$$S_t = \sum_{j=1}^{n} (a_j - c)(a_j - c)^T, \tag{4}$$

respectively. It is easy to show [5], [9] that the scatter matrices have the relationship

$$S_t = S_w + S_b. \tag{5}$$

The scatter matrices are analogous to covariance matrices, and are positive semi-definite. Thus, they can be factored into their "square-root" factors analogous to the case in many signal processing problems where the covariance matrix is factored into a product of a data matrix and its transpose [23], [24]. These factors can be used in subsequent processing and circumvent the condition number squaring problem, since $\kappa(AA^T) = \kappa^2(A)$.

Thus, we can define the matrices,

$$H_w = [A_1 - c^{(1)}e^{(1)^T}, A_2 - c^{(2)}e^{(2)^T}, \ldots, A_k - c^{(k)}e^{(k)^T}] \in \mathbb{R}^{m \times n}, \tag{6}$$

$$H_b = [\sqrt{n_1}(c^{(1)} - c), \sqrt{n_2}(c^{(2)} - c), \ldots, \sqrt{n_k}(c^{(k)} - c)] \in \mathbb{R}^{m \times k}, \tag{7}$$

and

$$H_t = [a_1 - c, \ldots, a_n - c] = A - ce^T \in \mathbb{R}^{m \times n}, \tag{8}$$

where $e^{(i)} \in \mathbf{R}^{n_i \times 1}$ and $e \in \mathbf{R}^{n \times 1}$ are vectors where all components are 1's. Then the scatter matrices can be expressed as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T, \quad \text{and} \quad S_t = H_t H_t^T. \tag{9}$$

Note that another way to define H_b is

$$H_b = [(c^{(1)} - c)e^{(1)^T}, (c^{(2)} - c)e^{(2)^T}, \ldots, (c^{(k)} - c)e^{(k)^T}] \in \mathbb{R}^{m \times n},$$

but the compact form shown in Eqn.(7) reduces the storage requirements and computational complexity [10], [11].

The measure of the within-cluster scatter can be defined as,

$$trace(S_w) = \sum_{i=1}^{k} \sum_{j \in N_i} \|a_j - c^{(i)}\|_2^2, \tag{10}$$

which is defined over all k clusters, and, similarly, define the measure for between-cluster scatter as

$$trace(S_b) = \sum_{i=1}^{k} \sum_{j \in N_i} \|c^{(i)} - c\|_2^2. \tag{11}$$

When items within each cluster are located tightly around their own cluster centroid, then trace(S_w) will have a small value. On the other hand, when the between-cluster relationship is remote, and hence the centroids of the clusters are remote, trace(S_b) will have a large value. Using the values trace(S_w), trace(S_b), and Eqn. (5), the cluster quality can be measured. There are several measures of the overall cluster quality which involve the three scatter matrices [5], [12], [21], including the maximization of

$$J_1 = trace(S_w^{-1} S_b), \tag{12}$$

or

$$J_2 = trace(S_w^{-1} S_t). \tag{13}$$

Note that both of the above criteria require S_w to be nonsingular, or equivalently, require H_w to have full rank. For more measures of cluster quality, their relationships, and their extension to text document data, see [5], [12].

In the lower dimensional space obtained from the linear transformation $G^T \in \mathbf{R}^{l\times m}$, the within-cluster, between-cluster, and total scatter matrices become

$$S_w^Y = G^T S_w G, \quad S_b^Y = G^T S_b G, \quad S_t^Y = G^T S_t G,$$

where the superscript Y denotes the scatter matrices in the l dimensional space transformed by G^T. Given k clusters in the full dimensional space, the linear transformation G^T that best maintains cluster separability in the reduced dimensional space should maximize trace(S_b^Y) and minimize trace(S_w^Y). Specifically, the problem is formulated as

$$\text{Find} \quad G^T \in \mathbb{R}^{l\times m} \quad s.t. \quad \max trace(G^T S_b G) \quad \text{and} \quad \min trace(G^T S_w G). \tag{14}$$

This simultaneous optimization problem can be approximated using the measures shown in Eqns. (12) - (13), by finding the matrix G that maximizes

$$J_1(G) = \text{trace}((G^T S_w G)^{-1}(G^T S_b G)),$$

or

$$J_2(G) = \text{trace}((G^T S_w G)^{-1}(G^T S_t G)).$$

Assuming $S_w = H_w H_w^T$ is nonsingular, it can be shown that [5], [10]

$$trace((S_w^Y)^{-1} S_b^Y) \leq trace(S_w^{-1} S_b)$$

and the upper bound on $J_1(G)$ is achieved as

$$trace((S_w^Y)^{-1} S_b^Y) = trace(S_w^{-1} S_b)$$

when $G \in \mathbb{R}^{m\times l}$ consists of l eigenvectors of $S_w^{-1} S_b$ corresponding to the l largest eigenvalues in the eigenvalue problem

$$S_w^{-1} S_b x = \lambda x. \tag{15}$$

Therefore, if we choose $l = k - 1$, dimension reduction results in no loss of cluster quality as measured by J_1 since S_b has at most $k - 1$ nonzero eigenvalues.

One limitation of the criteria $J_1(G)$ and $J_2(G)$ in many applications is that the matrix S_w must be nonsingular. When $m > n$, i.e., when the problem is undersampled, S_w is always singular. One way to alleviate this problem and generalize the classical LDA for the undersampled problem is to make S_w nonsingular by adding a small regularization parameter γI_m to the matrix S_w.

This method has been presented as regularized LDA. For more details, see [4] and also Section IV-B where we show that the regularized LDA solution can be obtained much faster by initial preprocessing of the data matrix with the QR decomposition.

Expressing λ as α^2/β^2, and using Eqns. (9), Eqn. (15) can be rewritten as

$$\beta^2 H_b H_b^T x = \alpha^2 H_w H_w^T x. \tag{16}$$

This problem can be solved using the GSVD [8], [15], [22], as described in the next section, and reformulation of the problem derived in Eqns. (15) and (16), which is the key component in the LDA/GSVD proposed in [9], [10]. This reformulation and its solution via the GSVD circumvents the nonsingularity restriction on the scatter matrices in classical LDA. The LDA/GSVD is a generalization of classical LDA and it is applicable regardless of the relative sizes between the problem dimension m and the number of data items n. Next we briefly review the GSVD.

III. Generalized Singular Value Decomposition and LDA/GSVD

In this section we briefly review the generalized singular value decomposition (GSVD) algorithm due to Paige and Saunders [15]. The proposed fast algorithms for linear discriminant analysis are computationally equivalent to the GSVD-based algorithm, which can be defined for any two matrices with the same number of columns. For a more restricted but a simpler form of the GSVD, see [8], [22]. Before introducing a GSVD algorithm, we first state the GSVD theorem:

Theorem 1 *Suppose any two matrices $K_b \in \mathbb{R}^{m\times n}$ and $K_w \in \mathbb{R}^{p\times n}$, with the same number of columns, are given. Then there exist orthogonal matrices $U \in \mathbb{R}^{m\times m}$, $V \in \mathbb{R}^{p\times p}$, and a nonsingular matrix $X \in \mathbb{R}^{n\times n}$ such that*

$$U^T K_b X = (\Sigma_b, 0) \quad \textit{and} \quad V^T K_w X = (\Sigma_w, 0), \tag{17}$$

where

$$\underset{m\times t}{\Sigma_b} = \begin{pmatrix} I_b & & \\ & D_b & \\ & & 0_b \end{pmatrix}, \quad \underset{p\times t}{\Sigma_w} = \begin{pmatrix} 0_w & & \\ & D_w & \\ & & I_w \end{pmatrix}, t = rank \begin{pmatrix} K_b \\ K_w \end{pmatrix}. \tag{18}$$

The matrices

$$I_b \in \mathbb{R}^{r\times r} \quad \textit{and} \quad I_w \in \mathbb{R}^{(t-r-s)\times(t-r-s)}$$

are identity matrices, where

$$r = rank \left(K_b K_w \right) - rank(K_w), s = rank(K_b) + rank(K_w) - rank \begin{pmatrix} K_b \\ K_w \end{pmatrix}, \tag{19}$$

$$0_b \in \mathbb{R}^{(m-r-s)\times(t-r-s)} \quad \textit{and} \quad 0_w \in \mathbb{R}^{(p-t+r)\times r}$$

are zero matrices with possibly no rows or no columns, and

$$D_b = diag(\alpha_{r+1}, \ldots, \alpha_{r+s}) \quad \textit{and} \quad D_w = diag(\beta_{r+1}, \ldots, \beta_{r+s})$$

satisfy

$$1 > \alpha_{r+1} \geq \cdots \geq \alpha_{r+s} > 0, \quad 0 < \beta_{r+1} \leq \cdots \leq \beta_{r+s} < 1, \tag{20}$$

and

$$\alpha_i^2 + \beta_i^2 = 1 \quad \textit{for} \;\; i = r+1, \ldots, r+s. \qquad \square$$

From Eqn. (17), we have

$$K_b^T K_b X = X^{-T} \begin{pmatrix} \Sigma_b^T \Sigma_b & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad K_w^T K_w X = X^{-T} \begin{pmatrix} \Sigma_w^T \Sigma_w & 0 \\ 0 & 0 \end{pmatrix}. \tag{21}$$

Defining

$$\alpha_i = 1, \;\; \beta_i = 0 \quad \text{for} \;\; i = 1, \ldots, r$$

and

$$\alpha_i = 0, \;\; \beta_i = 1 \quad \text{for} \;\; i = r+s+1, \ldots, t,$$

we have,

$$\beta_i^2 K_b^T K_b x_i = \alpha_i^2 K_w^T K_w x_i, \tag{22}$$

for $1 \leq i \leq t$, where x_i represents the ith column of X. For the remaining $n - t$ columns of X, both $K_b^T K_b x_i$ and $K_w^T K_w x_i$ are zero, and Eqn. (22) is satisfied for arbitrary values of α_i and β_i when $t + 1 \leq i \leq n$. The columns of X are the generalized right singular vectors for the matrix pair K_b and K_w. Equivalently, the columns of X are the generalized eigenvectors for the matrix pair $K_b^T K_b$ and $K_w^T K_w$. In terms of the generalized singular values, or the α_i/β_i quotients, r of them are infinite, s are finite and nonzero, and $t - r - s$ are zero. From Eqn. (19) we see that

$$rank(K_b) = r + s \quad \text{and} \quad rank(K_w) = t - r.$$

	α_i	β_i	$\lambda_i = \frac{\alpha_i}{\beta_i}$	x_i belongs to
$1 \le i \le r$	1	0	∞	null$(S_w) \cap$ null$(S_b)^c$
$r+1 \le i \le r+s$	$1 > \alpha_i > 0$	$0 < \beta_i < 1$	$\infty > \lambda_i > 0$	null$(S_w)^c \cap$ null$(S_b)^c$
$r+s+1 \le i \le t$	0	1	0	null$(S_w)^c \cap$ null(S_b)
$t+1 \le i \le m$	any value	any value	any value	null$(S_w) \cap$ null(S_b)

TABLE I

GENERALIZED EIGENVALUES λ_i'S AND EIGENVECTORS x_i'S FROM THE GSVD. THE SUPERSCRIPT c DENOTES THE COMPLEMENT.

```
Algorithm 1 LDA/GSVD
Given a data matrix $A \in \mathbb{R}^{m \times n}$ where the columns are partitioned into $k$ clusters, this algorithm computes the the dimension reducing transformation $G \in \mathbb{R}^{m \times (k-1)}$. For any vector $x \in \mathbb{R}^{m \times 1}$, $y = G^T x \in \mathbb{R}^{(k-1) \times 1}$ gives a $(k-1)$ dimensional representation of $x$.
  1) Compute $H_b \in \mathbb{R}^{m \times k}$ and $H_w \in \mathbb{R}^{m \times n}$ from $A$ according to Eqns. (7) and (6), respectively.
  2) Compute the complete orthogonal decomposition of $K = \begin{pmatrix} H_b^T \\ H_w^T \end{pmatrix} \in \mathbb{R}^{(k+n) \times m}$, i.e.,
     $P^T K V = \begin{pmatrix} R & 0 \\ 0 & 0 \end{pmatrix}$, where $P \in \mathbb{R}^{(k+n) \times (k+n)}$ and $V \in \mathbb{R}^{m \times m}$ are orthogonal,
     and $R$ is a square matrix with $rank(K) = rank(R)$.
  3) Let $t = rank(K)$.
  4) Compute W from the SVD of $P(1:k, 1:t)$, i.e., $U^T P(1:k, 1:t) W = \Sigma$.
  5) Compute the first $k-1$ columns of $V \begin{pmatrix} R^{-1}W & 0 \\ 0 & I \end{pmatrix}$, and assign them to $G$.
```

The GSVD algorithm adopted in this paper is based on the constructive proof of the GSVD presented in Paige and Saunders [15]. The GSVD is a powerful tool not only for algorithm development, but also for analysis. In Table I, four subspaces of interest spanned by the columns of X from the GSVD are characterized.

The GSVD has been applied to the matrix pair (H_b^T, H_w^T) to find the dimension reducing transformation for LDA/GSVD, which was first introduced in [10], [11]. The LDA/GSVD method does not depend on the scatter matrices being nonsingular. The algorithm is summarized in

Algorithm 2 LDA/QR-GSVD

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ where the columns are partitioned into k clusters, this algorithm computes the the dimension reducing transformation $G \in \mathbb{R}^{m\times(k-1)}$. For any vector $x \in \mathbb{R}^{m\times 1}$, $y = G^T x \in \mathbb{R}^{(k-1)\times 1}$ gives a $(k-1)$ dimensional representation x.

1) Compute the reduced QRD of A, i.e.,

$$A = Q_1 R$$

where $Q_1 \in \mathbb{R}^{m\times n}$ has orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is upper triangular.

2) Compute $\hat{H}_b \in \mathbb{R}^{n\times k}$ and $\hat{H}_w \in \mathbb{R}^{n\times n}$ from R according to Eqns. (25) and (24), respectively.

3) Compute the complete orthogonal decomposition of $\hat{K} = \begin{pmatrix} \hat{H}_b^T \\ \hat{H}_w^T \end{pmatrix} \in \mathbb{R}^{(k+n)\times n}$, i.e,

$\hat{P}^T \hat{K} \hat{V} = \begin{pmatrix} \hat{R} & 0 \\ 0 & 0 \end{pmatrix}$, where $\hat{P} \in \mathbb{R}^{(k+n)\times(k+n)}$, $\hat{V} \in \mathbb{R}^{n\times n}$ is orthogonal,

and $\hat{R}$ is a square matrix with $rank(\hat{K}) = rank(\hat{R})$.

4) Let $t = rank(\hat{K})$.

5) Compute $\hat{W}$ from the SVD of $\hat{P}(1:k, 1:t)$, i.e., $\hat{U}^T \hat{P}(1:k, 1:t)\hat{W} = \hat{\Sigma}$.

6) Compute the first $k-1$ columns of$\hat{V} \begin{pmatrix} \hat{R}^{-1}\hat{W} & 0 \\ 0 & I \end{pmatrix}$, and assign them to $\hat{G} \in \mathbb{R}^{n\times(k-1)}$.

7) $G = Q_1 \hat{G}$.

Algorithm 1 LDA/GSVD. For details see [10], [11].

One of the disadvantages of the LDA/GSVD is the high computational cost. Several methods have been proposed to mitigate this and provide a faster LDA/GSVD algorithm [16], [17]. As can be seen in Algorithm 1, LDA/GSVD involves the computation of two SVD's. In the following, we present fast algorithms for LDA for both undersampled and oversampled cases. By utilizing the QR or Cholesky decompositions and regularization in a space with a much smaller dimension than that of the original LDA space, we have designed fast algorithms for LDA for both undersampled as well as for oversampled problems.

IV. Efficient LDA Algorithms with QR Decomposition and Regularization

In the following subsections, three algorithms are presented that solve the LDA/GSVD more efficiently while retaining its advantages in the likely event of singular scatter matrices. For undersampled problems, i.e. $m \geq n$, QR decomposition preprocessing is proposed to reduce the problem size without loss of data information. Furthermore, using a regularized version of QRD preprocessing has the same advantages as that of the GSVD for singular scatter matrices. Oversampled problems, $m < n$ are solved by manipulating the "square-root" factors of the scatter matrices. The dimension reduction can then be applied via the Cholesky decomposition. Three algorithms are presented to illustrate these computations. Their complexity analysis is also provided.

A. QR Decomposition Preprocessing of Undersampled Data

In this section, we assume that the data set is undersampled, i.e., $A \in \mathbb{R}^{m\times n}$ where $m \geq n$ and show that QR decomposition preprocessing can reduce the problem size without loss of essential information in the data for the LDA formulation [11]. For any matrix $A \in \mathbb{R}^{m\times n}, m \geq n$, there exists an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$ and an upper triangular matrix $R \in \mathbb{R}^{n\times n}$ such that

$$A = Q \begin{pmatrix} R \\ 0 \end{pmatrix} = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R \\ 0 \end{pmatrix} = Q_1 R$$

where $Q_1 \in \mathbb{R}^{m\times n}$ and $Q_2 \in \mathbb{R}^{m\times(m-n)}$.

The columns of $A \in \mathbb{R}^{m\times n}$ in m dimensional space are represented in an n dimensional space as

$$Q_1^T A = R \tag{23}$$

where Q_1^T is the dimension reducing transformation. Then based on the n dimensional representation of n data points which are now the columns of R and its k clusters $[R_1 \;\cdots\; R_k]$ where $R_i = Q_1^T A_i \in \mathbb{R}^{n\times n_i}$ for $i = 1, ..., k$, we form the matrices

$$\hat{H}_w = [R_1 - \hat{c}^{(1)}e^{(1)^T}, R_2 - \hat{c}^{(2)}e^{(2)^T}, \ldots, R_k - \hat{c}^{(k)}e^{(k)^T}] \in \mathbb{R}^{n\times n}, \tag{24}$$

$$\hat{H}_b = [\sqrt{n_1}(\hat{c}^{(1)} - \hat{c}), \sqrt{n_2}(\hat{c}^{(2)} - \hat{c}), \ldots, \sqrt{n_k}(\hat{c}^{(k)} - \hat{c})] \in \mathbb{R}^{n\times k}, \tag{25}$$

where $\hat{c}^{(k)} = Q_1^T c^{(k)} \in \mathbb{R}^{n\times 1}$ and $\hat{c} = Q_1^T c \in \mathbb{R}^{n\times 1}$. Then it is easy to see that

$$\hat{H}_w = Q_1^T H_w \quad and \quad \hat{H}_b = Q_1^T H_b$$

With the scatter matrices

$$\hat{S}_w = \hat{H}_w \hat{H}_w^T = Q_1^T H_w H_w^T Q_1 = Q_1^T S_w Q_1 \text{ and} \tag{26}$$

$$\hat{S}_b = \hat{H}_b \hat{H}_b^T = Q_1^T H_b H_b^T Q_1 = Q_1^T S_b Q_1, \tag{27}$$

suppose we find a matrix $\hat{X}$ that minimizes $trace(\hat{X}^T \hat{S}_w \hat{X})$ and maximizes $trace(\hat{X}^T \hat{S}_b \hat{X})$. Then since

$$\hat{X}^T \hat{S}_w \hat{X} = \hat{X}^T \hat{H}_w \hat{H}_w^T \hat{X} = \hat{X}^T Q_1^T H_w H_w Q_1 \hat{X} \tag{28}$$

$$\hat{X}^T \hat{S}_b \hat{X} = \hat{X}^T \hat{H}_b \hat{H}_b^T \hat{X} = \hat{X}^T Q_1^T H_b H_b Q_1 \hat{X} \tag{29}$$

$X = Q_1 \hat{X}$ provides the solution for minimizing $trace(G^T S_w G)$ and maximizing $trace(G^T S_b G)$, which is the LDA/GSVD solution shown in Algorithm 1. The discussion above shows that the preprocessing step by the reduced QR decomposition of A enables one to solve LDA/GSVD by manipulating much smaller matrices of size $n \times n$ rather than $m \times n$ in the case of under-sampled problems.

Specifically, efficiency in LDA/QR-GSVD (see Algorithm 2) is obtained in step 3 where the time complexity of $O(n^3)$ is sufficient for the complete orthogonal decomposition of the $n \times n$ matrix, while LDA/GSVD requires $O(m^2 n)$ with an $m \times n$ matrix in step 2. For undersampled problems where the data dimension, m, could be up to several thousands, QR decomposition preprocessing can produce significant time and memory savings as demonstrated in Section V.

B. Regularization and QRD Preprocessing

When S_w is singular or ill conditioned, in regularized LDA a diagonal matrix γI with $\gamma > 0$, is added to S_w to make it nonsingular. Regularized LDA has been commonly used for dimension reduction of high dimensional data in many application areas. However, high dimensionality can make the time complexity and memory requirements very expensive, since computing the SVD of the scatter matrices is required for regularized LDA.

We now show that if we apply regularization *after* preprocessing by the QR decomposition on the data set, the process of regularization becomes simpler since we work with the within-cluster scatter matrix, which is transformed into a much smaller dimensional space ($n \times n$ compared to $m \times n$). In addition, we will show that though regularization is preceded by preprocessing of the data matrix A, it is equivalent to regularized LDA without preprocessing. When this observation is combined within the framework of the LDA/GSVD algorithm, then it also makes the LDA/GSVD algorithm much simpler since regularization of S_w results in full rank of the corresponding matrix K, and therefore, eliminates the need of a rank revealing decomposition in Step 3 of the LDA/QR-GSVD algorithm.

Suppose regularization of the within-cluster scatter matrix $\hat{S}_w$ is performed, which is obtained after the QR decomposition preprocessing as in Eqn. (26) and obtain $\hat{S}_w + \gamma I$. Also, suppose we look for a solution $\hat{X}$ that maximizes

$$\mathrm{trace}((\hat{X}^T(\hat{S}_w + \gamma I)^{-1}\hat{X})(\hat{X}^T\hat{S}_b\hat{X})). \tag{30}$$

We now show that this is equivalent to a problem of finding X that maximizes

$$\mathrm{trace}((X^T(S_w + \gamma I)X)^{-1}(X^TS_bX)). \tag{31}$$

Note that

$$\hat{S}_w + \gamma I = \begin{pmatrix} \hat{H}_w & \sqrt{\gamma}I \end{pmatrix} \begin{pmatrix} \hat{H}_w^T \\ \sqrt{\gamma}I \end{pmatrix}.$$

Since $\hat{H}_w\hat{H}_w^T = Q_1^TH_wH_w^TQ_1$ and $\gamma I = \gamma Q_1^TQ_1$, we have

$$\hat{H}_w\hat{H}_w^T + \gamma I = Q_1^T(H_wH_w^T + \gamma I)Q_1$$

and

$$\hat{X}^T(\hat{S}_w + \gamma I)\hat{X} = \hat{X}^TQ_1^T(H_wH_w^T + \gamma I)Q_1\hat{X}.$$

Together with

$$\hat{X}^T\hat{S}_b\hat{X} = \hat{X}^TQ_1^TH_bH_b^TQ_1\hat{X}, \tag{32}$$

the above shows that the solution obtained from regularization, after QR preprocessing, is equivalent to the original regularized LDA. In the context of the LDA/GSVD or LDA/QR-GSVD algorithms, regularization has the beneficial effect of simplifying the algorithm due to the fact that $\hat{S}_w + \gamma I$ is nonsingular. Specifically, when $\hat{S}_w$ is replaced by $\hat{S}_w + \gamma I$, the matrix $\hat{K}$ in Algorithm LDA/QR-GSVD is changed to

$$\hat{K}_\gamma = \begin{pmatrix} \hat{H}_b^T \\ \hat{H}_w^T \\ \sqrt{\gamma}I \end{pmatrix} \in \mathbb{R}^{(k+2n)\times n}$$

and $rank(\hat{K}_\gamma)$ is full for any value $\gamma > 0$. For this reason we do not have to go through the process of revealing the rank as in Step 3 of Algorithm 2. Instead, the reduced QR decomposition of $\hat{K}_\gamma$ will suffice. The new algorithm is presented in Algorithm 3.

Regularization provides another advantage by alleviating the overfitting problem inherent in LDA/GSVD. When S_w is singular and, accordingly, $dim(null(S_w)) \geq 1$, Table III and

$$X^T K_b^T K_b X = \begin{pmatrix} \Sigma_b^T \Sigma_b & 0 \\ 0 & 0 \end{pmatrix} \quad \text{and} \quad X^T K_w^T K_w X = \begin{pmatrix} \Sigma_w^T \Sigma_w & 0 \\ 0 & 0 \end{pmatrix} \tag{33}$$

show that the largest generalized eigenvalue is ∞. The leading generalized eigenvectors that consist of the first r columns of X give

$$X_1^T S_w X_1 = 0$$

where $X_1 \in \mathbb{R}^{n \times r}$. This means that the within-cluster scatter matrix, after the dimension reducing transformation by X_1, becomes zero and accordingly all data points in A in each cluster are transformed to the same point. Then the optimal dimension reducing transformation $[X_1 \quad X_2]$ from LDA/GSVD tends to map every point within the same cluster onto an extremely narrow region of a single point, resulting in $\left\|[X_1 \quad X_2]^T S_w [X_1 \quad X_2]\right\|_2 \approx 0$. Since the goal of LDA is to find a dimension reducing transformation that minimizes the within-cluster relationship and maximizes the between-cluster relationship, the small norm of the within-cluster scatter matrix in the reduced dimensional space may seem to be a desirable result. However, such a small norm of the within-cluster scatter matrix causes difficulties in generalizing test data, and having very small regions for each cluster will prove detrimental for classification. In order to resolve this problem, we need to make S_w nonsingular by adding a regularization term to S_w, such as $S_w + \gamma I$ where γ is a small positive number. This idea of applying regularization to LDA/GSVD may seem to be identical to regularized LDA. However, within the LDA/GSVD algorithm framework, QR decomposition preprocessed data and regularization, an algorithm obtained (Algorithm 3) is significantly more efficient as shown in the numerical experiments, Section V.

C. Fast LDA Algorithm for Oversampled Problems

When the number of data points exceeds the data dimension, QR decomposition preprocessing becomes useless since the dimension of the upper triangular matrix R will be the same as the dimension of the original data. The solution to this is arrived at by manipulating the matrix

Algorithm 3 LDA/QR-regGSVD

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with $m \geq n$ where the columns are partitioned into k clusters and a regularization parameter $\gamma > 0$, this algorithm computes the the dimension reducing transformation $G \in \mathbb{R}^{m\times(k-1)}$. For any vector $x \in \mathbb{R}^{m\times 1}$, $y = G^T x \in \mathbb{R}^{(k-1)\times 1}$ gives a $(k-1)$ dimensional representation x.

1) Compute the reduced QRD of A, i.e.,

$$A = Q_1 R$$

where $Q_1 \in \mathbb{R}^{m\times n}$ has orthonormal columns and $R \in \mathbb{R}^{n\times n}$ is upper triangular.

2) Compute $\hat{H}_b \in \mathbb{R}^{n\times k}$ and $\hat{H}_w \in \mathbb{R}^{n\times n}$ from R according to Eqns. (25) and (24), respectively.

3) Compute the reduced QRD of $\hat{K}_\gamma = \begin{pmatrix} \hat{H}_b^T \\ \hat{H}_w^T \\ \sqrt{\gamma} I \end{pmatrix} \in \mathbb{R}^{(k+2n)\times n}$, i.e., $\hat{P}_\gamma^T K = \hat{R}_\gamma$, where $\hat{P}_\gamma \in \mathbb{R}^{(k+2n)\times n}$ has orthonormal columns and $\hat{R}_\gamma \in \mathbb{R}^{n\times n}$ is upper triangular.

4) Compute $\hat{W}_\gamma$ from the SVD of $\hat{P}_\gamma(1:k, 1:n)$, i.e., $\hat{U}_\gamma^T \hat{P}_\gamma(1:k, 1:n)\hat{W}_\gamma = \hat{\Sigma}_\gamma$.

5) Solve the triangular system $\hat{R}_\gamma \hat{G}_\gamma = \hat{W}_\gamma(:, 1:k-1)$ for $\hat{G}_\gamma$.

6) $G = Q_1 \hat{G}_\gamma$

H_w instead of the original data matrix A. Our goal here is to reduce the dimension of the data to increase the speed of LDA/GSVD without losing information in the original data. In the LDA/GSVD algorithm, H_w is used to form the within cluster scatter matrix S_w, where $S_w = H_w H_w^T$. As long as an H_w is found that satisfies this relationship with S_w, this H_w can be used in the algorithm. Therefore, we want to find a matrix that is equivalent to H_w but has smaller dimension. Such a matrix can be found when we compute the Cholesky decomposition of S_w as

$$\underbrace{S_w}_{m\times m} = \underbrace{H_w}_{m\times n}\underbrace{H_w^T}_{n\times m} = \underbrace{C_w^T}_{m\times m}\underbrace{C_w}_{m\times m} \quad (34)$$

Algorithm 4 LDA/Chol

Given a data matrix $A \in \mathbb{R}^{m\times n}$ with $m < n$ where the columns are partitioned into k clusters, this algorithm computes the the dimension reducing transformation $G \in \mathbb{R}^{m\times(k-1)}$. For any vector $x \in \mathbb{R}^{m\times 1}$, $y = G^T x \in \mathbb{R}^{(k-1)\times 1}$ gives a $(k-1)$ dimensional representation x.

1) Compute $H_b \in \mathbb{R}^{m\times k}$ and $H_w \in \mathbb{R}^{m\times n}$ from A according to Eqns. (25) and (24), respectively.
2) Compute $S_w = H_w H_w^T \in \mathbb{R}^{m\times m}$ and its Cholesky decomposition, i.e.,
$$S_w = C_w^T C_w$$
3) Compute the reduced QR decomposition of $K = \begin{pmatrix} H_b^T \\ C_w \end{pmatrix} \in \mathbb{R}^{(k+m)\times m}$,
i.e. $P^T K = F$ where $P \in \mathbb{R}^{(k+m)\times m}$ has orthonormal columns and $F \in \mathbb{R}^{n\times n}$ is upper triangular.
4) Compute W from the SVD of $P(1:k, 1:m)$, which is $U^T P(1:k,1:m)W = \Sigma_A$.
5) Compute the first $k-1$ columns of $X = F^{-1}W$, and assign them to G.

where $m < n$. Now, we may use C_w^T instead of H_w to construct the matrix K in Step 3 of Algorithm 3. The rest of the procedure is followed as in Algorithm 3 with the modified matrix K. Since the dimension of K is reduced from $m\times(n+k)$ to $m\times(m+k)$ and $m < n$, the LDA/GSVD process becomes faster. This result is summarized in algorithm 4. In the classical LDA, the within-cluster scatter matrix S_w is considered nonsingular and its inverse is utilized. Under this assumption, C_w will also be nonsingular and in Algorithm 4, no rank revealing decomposition is used in Step 3. However, it is not always true that S_w is nonsingular for oversampled problems. In this case, we propose computing the reduced QR decomposition of $H_w^T = Q_w R_w$ where $R_w \in \mathbb{R}^{m\times m}$ and forming K in Step 3 of Algorithm 4 as $K = \begin{pmatrix} H_b^T \\ R_w \\ \gamma I \end{pmatrix}$. With this modification, Algorithm 4 has another advantage in that it allows classical LDA to handle both singular and nonsingular scatter matrices for oversampled cases.

TABLE II

DESCRIPTION OF THE DATA SETS USED. THE FIRST FOUR DATA SETS CORRESPOND TO UNDERSAMPLED CASES AND THE LAST THREE ARE DATA SETS FOR OVERSAMPLED CASES.

Data set	number of data	dimension	number of clusters	training	test
Text	210	5896	7	168	42
Yale	165	77760	15	135	30
AT&T	400	10304	40	320	80
Feret	130	3000	10	100	30
ImageSeg	2310	19	7	2100	210
Optdigit	5610	64	10	3813	1797
Isolet	7797	617	26	6238	1559

V. EXPERIMENTAL RESULTS

We have tested the proposed algorithms on undersampled problems as well as oversampled problems. For the undersampled case, regularized LDA, LDA/GSVD, LDA/QR-GSVD, and LDA/QR-regGSVD are tested using data sets for text categorization and face recognition. For the oversampled case, LDA/GSVD, LDA/Chol, and classical LDA are tested for various benchmark data sets from the UCI machine learning repository. A detailed description of the data sets is given in Table II. All the experiments were run using MATLAB on Windows XP with 1.79 GHz CPU and 1 GB memory.

The first data set denoted as Text in Table II is for the text categorization problem and was downloaded from http://www-users.cs.umn.edu/∼karypis/cluto/download.html. This data set consists of a total 210 instances of documents and 5896 terms. The data set was composed of 7 clusters, each cluster contains 30 data samples. From this data set, $4/5$'s were used for training, and the remaining for testing. From the face recognition data, three data sets were used, Yale, AT&T and Feret face data. The Yale face data consists of 165 images, each image contained 77760 pixels. The data set is made up of 15 clusters and each cluster contains 11 images of the same person. Out of 165 images, 135 were used for training and 30 were used for testing. The AT&T and Feret face data, were similarly composed and used in the experiments.

Tables III and IV compare the time complexities and classification accuracies of those algorithms tested on undersampled problems, respectively. For all the methods, k nearest neighbor

TABLE III

COMPARISON OF CPU TIME IN SECONDS FOR UNDERSAMPLED PROBLEMS. FOR AT&T AND YALE FACE DATA, LDA/GSVD AND REGULARIZED LDA (REGLDA) RAN OUT OF MEMORY, THEREFORE NO RESULTS ARE AVAILABLE.

Data set	LDA/GSVD	LDA/QR-GSVD	LDA/QR-regGSVD	regLDA
Text	48.834	0.141	0.033	42.220
Feret	10.938	0.033	0.009	9.300
AT&T	-	0.956	0.217	-
Yale	-	0.066	0.017	-

(kNN) classification was used in the dimension reduced space. For different k values in kNN classification, accuracies were varied only slightly with differences less than 1 $\%$, and the results from the nearest neighbor classification with $k = 1$ are shown in Table IV. Although modification of the within-cluster scatter matrix by regularization can affect the accuracy of the classification, we restrict the value of the regularization variable, γ, to small values keeping its influence on the classification results to a minimum. The regularization parameter γ was tested for several small values such as $\gamma = 10^{-2}$,$\gamma = 10^{-4}$,$\gamma = 10^{-6}$, and $\gamma = 10^{-8}$, and it was found that the accuracy for the classification did not change significantly. Accuracies shown in Table IV were measured with $\gamma = 10^{-2}$ for LDA/QR-regGSVD and regularized LDA.

As shown in Table III and IV, while classification accuracies for the three algorithms, LDA/GSVD, LDA/QR-GSVD, and LDA/QR-regGSVD, and regularized LDA were similar, the computing time for each method was quite different. In particular, the time complexity of LDA/QR-regGSVD was reduced greatly. Time complexity reduction is obtained by eliminating the rank revealing procedure within the orthogonal decomposition of $K = \begin{pmatrix} H_b^T \\ H_w^T \end{pmatrix}$. Since the regularized within-cluster scatter matrix is full rank, we did not compute the SVD of K but the QR decomposition, which completes in a finite number of steps.

Unlike undersampled problems, for the oversampled case the QR preprocessing does not reduce the dimension of the input data. Instead, the Cholesky decomposition can speed up the process by reducing the dimension of the within-cluster scatter matrix to $C_w \in \mathbb{R}^{m \times m}$ as in the algorithm LDA/Chol. As in the undersampled case, we may need to regularize the within-cluster scatter matrix S_w to make it positive definite. Three data sets were used, which are

TABLE IV

COMPARISON OF CLASSIFICATION ACCURACIES (%) IN UNDERSAMPLED PROBLEMS.

Data set	LDA/GSVD	LDA/QR-GSVD	LDA/QR-regGSVD	regLDA
Text	98.33	98.33	97.86	96.67
Feret	92.67	92.67	92.17	95.17
AT&T	-	94.37	93.75	-
Yale	-	97.33	97.33	-

described in the last three rows in Table II, in order to test the three algorithms, LDA/GSVD and LDA/Chol and classical LDA. For oversampled problems, the classical LDA is applicable when the within-cluster scatter matrix is nonsingular. Image segmentation data, optical recognition for handwritten digits, and Isolet spoken letter recognition data sets were obtained from the UCI Machine Learning repository. Tables V and VI show time complexity and classification accuracy, respectively. As discussed in Section IV-C, LDA/Chol reduced the time complexity dramatically compared with LDA/GSVD and LDA, while maintaining classification performance competitive to other methods.

VI. CONCLUSION

As demonstrated in this paper, for high-dimensional undersampled and oversampled problems, the classical LDA requires modification in order to solve a wider range of problems. The purpose of modifying LDA/GSVD is not only to prevent overfitting of the data in a reduced dimensional space, but also to reduce the amount of processing time required. These issues were addressed by developing fast algorithms for classical LDA and generalizations of LDA such as regularized LDA and LDA/GSVD by taking advantage of QR decomposition preprocessing and regularization in the framework of the LDA/GSVD algorithm. We have shown that though regularization is preceded by preprocessing of the data matrix A, this is equivalent to regularized LDA without preprocessing. When this observation is combined within the framework of the LDA/GSVD algorithm, then it also makes the LDA/GSVD algorithm much simpler since regularization of S_w results in full rank of the corresponding matrix K, and therefore, eliminates the need of a rank revealing decomposition in Step 3 of the LDA/QR-GSVD algorithm. Reducing the time

TABLE V

COMPARISON OF CPU TIME IN SECONDS FOR OVERSAMPLED PROBLEMS

Data set	LDA/GSVD	LDA/Chol	LDA
ImageSeg	0.842	0.005	0.905
Optdigit	8.966	0.016	9.590
Isolet	98.195	6.695	99.328

TABLE VI

COMPARISON OF CLASSIFICATION ACCURACIES (%)IN OVERSAMPLED PROBLEMS

Data set	LDA/GSVD			LDA/Chol			LDA		
k in kNN	1	15	29	1	15	29	1	15	29
ImageSeg	97.0	95.0	93.7	96.9	94.8	93.8	92.4	92.1	91.6
Optdigit	94.5	94.7	94.4	94.5	94.7	94.4	81.9	77.3	71.7
Isolet	93.2	93.8	93.7	93.2	93.8	93.7	94.8	95.3	95.1

complexity is obtained by eliminating this rank revealing procedure of $K = \begin{pmatrix} H_b^T \\ H_w^T \end{pmatrix}$ and utilizing the QR decomposition for the regularized within-cluster scatter matrix. The algorithms presented in this paper successfully achieve both objectives and are applicable to a wide range of problems, owing to the generalization to problems with potentially singular scatter matrices.

REFERENCES

[1] M.W. Berry and S.T. Dumais and G.W. O'Brien, "Using linear algebra for intelligent information retrieval," SIAM Review, vol. 37, pp. 573-595, 1995.

[2] Å. Bjorck, Numerical Methods for Least Squares Problems, SIAM, Philadelphia, 1996.

[3] W.B. Frakes and R. Baeza-Yates, Information Retrieval: Data Structures and Algorithms, Prentice Hall PTR, 1992.

[4] J.H. Friedman, "Regularized discriminant analysis," Journal of the American statistical association, vol. 84, no. 405, pp. 165-175, 1989.

[5] K. Fukunaga, Introduction to Statistical Pattern Recognition, second edition, Academic Press, Inc., 1990.

[6] R.O. Duda and P.E. Hart and D. G. Stork, Pattern Classification, second edition, John Wiley and Sons, Inc., 2001.

[7] C.M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

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

[9] A.K. Jain, and R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, 1988.

[10] P. Howland and M. Jeon and H. Park, “Structure preserving dimension reduction for clustered text data based on the generalized singular value decomposition,” SIAM Journal on Matrix Analysis and Applications, vol. 25, no. 1, pp. 165-179, 2003.

[11] P. Howland and H. Park, “Equivalence of several two-stage methods for linear discriminant analysis,” Proc. fourth SIAM International Conference on Data Mining, Kissimmee, FL, pp. 69-77, April, 2004.

[12] P. Howland and H. Park, “Generalizing Discriminant Analysis Using the Generalized Singular Value Decomposition,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, no. 8, pp. 995-1006, 2004.

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

[14] C.L. Lawson and R.J. Hanson, Solving Least Squares Problems, SIAM, Philadelphia, 1995.

[15] C.C. Paige and M.A. Saunders, “Towards a generalized singular value decomposition,” SIAM J. Numer. Anal., vol. 18, pp. 398-405, 1981.

[16] C. Park and H. Park and P. Pardalos, “A Comparative Study of Linear and Nonlinear Feature Extraction Methods,” Proc. of 4th ICDM, pp. 495-498, 2004.

[17] C. Park and H. Park, “A relationship between LDA and the generalized minimum squared error solution,” SIAM Journal on Matrix Analysis and Applications, vol. 27, no. 2, pp. 474-492, 2005.

[18] H. Park and M. Jeon and J.B. Rosen, “Lower dimensional representation of text data based on centroids and least squares,” BIT, vol. 43, no. 2, pp. 1-22, 2003.

[19] G. Salton, The SMART Retrieval System, Prentice Hall, 1971.

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

[21] S. Theodoridis and K. Koutroumbas, Pattern Recognition, Academic Press, 1999.

[22] C.F. Van Loan, “Generalizing the singular value decomposition,” SIAM J. Numer. Anal., vol. 13, pp. 76-83, 1976.

[23] J. M. Speiser and C.F. Van Loan, “Signal processing computations using the generalized singular value decomposition,” Proc. SPIE Vol.495, Real Time Signal Processing VII, pp. 47-55, 1984.

[24] C.F. Van Loan and J. M. Speiser, “Computation of the CS Decomposition with Applications to Signal Processing,” Proc. SPIE Vol.696, Advanced Signal Processing Algorithms and Architectures, 1986.