Technical Report

Department of Computer Science
and Engineering
University of Minnesota
4-192 EECS Building
200 Union Street SE
Minneapolis, MN 55455-0159 USA

TR 03-026

A new optimization criterion for generalized discriminant analysis on undersampled problems

Jieping Ye, Ravi Janardan, Cheonghee Park, and Haesun Park

June 10, 2003

A new optimization criterion for generalized discriminant analysis on undersampled problems

Jieping Ye* Ravi Janardan* Cheong Hee Park* Haesun Park*

Abstract

We present a new optimization criterion for discriminant analysis. The new criterion extends the optimization criteria of the classical linear discriminant analysis (LDA) by introducing the pseudo-inverse when the scatter matrices are singular. It is applicable regardless of the relative sizes of the data dimension and sample size, overcoming a limitation of the classical LDA. Recently, a new algorithm called LDA/GSVD for structure-preserving dimension reduction has been introduced, which extends the classical LDA to very high-dimensional undersampled problems by using the generalized singular value decomposition (GSVD). The solution from the LDA/GSVD algorithm is a special case of the solution for our generalized criterion in this paper, which is also based on GSVD.

We also present an approximate solution for our GSVD-based solution, which reduces computational complexity by finding sub-clusters of each cluster, and using their centroids to capture the structure of each cluster. This reduced problem yields much smaller matrices of which the GSVD can be applied efficiently. Experiments on text data, with up to 7000 dimensions, show that the approximation algorithm produces results that are close to those produced by the exact algorithm.

Keywords: Clustering, dimension reduction, generalized singular value decomposition, linear discriminant analysis, text mining.

1 Introduction

Many interesting data mining problems involve data sets represented in very high dimensional spaces. We consider dimension reduction of high-dimensional, undersampled data, where the dimension of the data points is higher than the number of data points. The high-dimensional, undersampled problems frequently occur in many applications including information retrieval [14, 21], facial recognition [11] and microarray analysis [1].

One application area of interest in this paper is vector space based-information retrieval. The dimension of the document vectors is typically very high, due to a large number of terms that appear in the collection of the documents. In the vector space based-model, documents are represented as column vectors in a term-document matrix. For an $m \times n$ term-document matrix $A = (a_{ij})$, its (i, j)-th term a_{ij} represents the weighted frequency of term i in document j. Several weighting schemes have been developed for encoding the document collection in a term-document matrix [13, 15]. (We applied the commonly used *tf-idf* weighting scheme [20, 24] for all our experiments.) An advantage of the vector space based-

*Department of Computer Science & Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A. {jieping,janardan,chpark,hpark}@cs.umn.edu. Research of J. Ye and R. Janardan is sponsored, in part, by the Army High Performance Computing Research Center under the auspices of the Department of the Army, 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. Research of C. Park and H. Park has been supported in part by the National Science Foundation Grant No. CCR-0204109. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation (NSF).

method is that once the collection of documents is represented as columns of the term-document matrix in a high dimensional space, the algebraic structure of the related vector space can then be exploited.

When the documents are already clustered, we would like to find a dimension-reducing transformation that preserves the cluster structure of the original full space even after the dimension reduction. Throughout the paper, the input documents are assumed to have been already clustered before the dimension reduction step. When the documents are not clustered, then efficient clustering algorithms such as K-Means [4, 12] can be applied before the dimension reduction step. We seek a reduced representation of the document vectors, which best preserves the structure of the original document vectors.

Latent Semantic Indexing has been widely used for dimension reduction of text data [2, 3]. It is based on lower rank approximation of the term-document matrix from the singular value decomposition (SVD) [8]. Although the SVD provides the optimal reduced rank approximation of the matrix when the difference is measured in the L_2 or Frobenius norm, it has limitations in that it does not consider cluster structure in the data and is expensive to compute. Moreover, the choice of the optimal reduced dimension is difficult to determine theoretically.

The Orthogonal Centroid Method has been introduced [18], as a dimension reduction method that maximizes the separation between clusters. It was also used for class visualization of the high dimensional data [5]. The main advantage of this method is its computational efficiency since a dimension-reducing transformation based on the symmetric eigenvalue decomposition can be computed by simple orthogonal decomposition of the matrix that involves only the centroids of the clusters [10]. A disadvantage of the Orthogonal Centroid method is that it does not take into account the so-called within-cluster distance.

The linear discriminant analysis (LDA) method has been applied for decades for dimension reduction (feature extraction) of clustered data in pattern recognition [7]. It is classically formulated as an optimization problem on covariance matrices. A serious disadvantage of the LDA is that its objective function requires that at least one of the covariance matrices be non-singular. In many modern data mining problems such as information retrieval, facial recognition, and microarray data analysis, all of the covariance matrices in question can be singular since the data items are from a very high-dimensional space and in general the number of sample data points does not exceed this dimension. Recently, a generalization of LDA based on the generalized singular value decomposition (GSVD) has been developed [9, 10], which is applicable regardless of the data dimension, and, therefore can be used for the undersampled problems. The classical LDA solution becomes a special case of this LDA/GSVD method. In [9, 10], the solution from LDA/GSVD is justified to preserve the cluster structure in the original full space after dimension reduction. However, no explicit global objective function has been presented.

In this paper, we present a new generalized optimization criterion for discriminant analysis. Our class-preserving projections are tailored for extracting the class structure of high dimensional data, and are closely related to the classical linear discriminant analysis. The main advantage of this proposed algorithm is that the new criterion is applicable to undersampled problems. A detailed mathematical derivation of the proposed new optimization problem is presented. The GSVD technique is the key component for the derivation. The solution from the LDA/GSVD algorithm is a special case of the solution for this new criterion. Since there is no approximation involved in the proposed algorithm, we call it the *exact algorithm*, to distinguish from the approximation algorithm introduced below.

One limitation of the GSVD-based method is its high computational complexity in handling large matrices. We propose an approximation algorithm based on sub-clustering of clusters to reduce the cost of computing the SVD involved in the computation of GSVD. Each cluster is further sub-clustered so that the overall structure of each cluster can be represented by the set of centroids

corresponding to each sub-clusters. As a result, only a few vectors are needed to define the scatter matrices, thus reducing the computational complexity. Experimental results show that the approximation algorithm produces results close to those produced by the exact one.

To compare our proposed algorithms with other dimension reduction algorithms, we use the K-Nearest neighbors (KNN) method [6] based on the Euclidean distance for classification. We use 10-fold cross-validation is used for estimating the misclassification rate. In 10-fold cross-validation, we divide the data into 10 subsets of (approximately) equal size. Then we do the training and testing 10 times, each time leaving out one of the subsets from training, and using only the omitted subset for testing. The misclassification rate is the average from the 10 runs. The misclassification rates on different data sets for all the dimension reduction algorithms discussed in this paper are reported for comparison. The results in Section 6 show that our proposed exact and approximation algorithms outperform other dimension reduction algorithms discussed in this paper. More interestingly, the results produced by the approximation algorithm are close to those produced by the exact algorithm, while the approximation algorithm deals with matrices of much smaller sizes than those in the exact algorithm, hence has lower computational complexity.

The main contributions of this paper include:

1. A generalization of the classical discriminant analysis to small sample size data using a new criterion, where the non-singularity of the scatter matrices is not required.

2. Mathematical derivation of the solution for the new optimization criterion.

3. An efficient approximation algorithm for the optimization problem.

4. Detailed experimental results for our exact and approximation algorithms and comparisons with competing algorithms.

The rest of the paper is organized as follows: Classical discriminant analysis is reviewed in Section 2. A generalization of the classical discriminant analysis using the new criterion is presented in Section 3. An efficient approximation algorithm is presented in Section 4 and a comparison between the proposed method and other dimension reduction algorithms is made in Section 5. Several experimental results are presented in Section 6 and concluding discussions are presented in Section 7.

2 Classical discriminant analysis

Given a term-document matrix $A \in R^{m\times n}$, we consider finding a linear transformation $G^T \in R^{\ell\times m}$ that maps each column a_i, for $1 \le i \le n$, of A in the m-dimensional space to a column y_i in the ℓ-dimensional space:

$$G^T : a_i \in R^{m\times 1} \rightarrow y_i \in R^{\ell\times 1}. \qquad (1)$$

Assume the original data is already clustered. The goal here is to find the transformation G^T such that cluster structure of the original full high-dimensional space is preserved in the reduced dimensional space. Let the document matrix A be partitioned into k clusters as

$$A = [A_1 \quad A_2 \quad \cdots \quad A_k],$$

where $A_i \in R^{m\times n_i}$, and $\sum_{i=1}^{k} n_i = n$.

Let N_i be the set of column indices that belong to the ith cluster, i.e., a_j, for $j \in N_i$, belongs to the ith cluster.

In general, if each cluster is tightly grouped, but well separated from the other clusters, the quality of the cluster is considered to be high. In discriminant analysis [7], two scatter matrices, within-cluster and between-cluster scatter matrices are defined to quantify the quality of the cluster, as follows:

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

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

$$= \sum_{i=1}^{k} n_i (c^{(i)} - c)(c^{(i)} - c)^T, \quad (2)$$

where the centroid $c^{(i)}$ of the ith cluster is defined as

$$c^{(i)} = \frac{1}{n_i} A_i e^{(i)},$$

where $e^{(i)} = (1, 1, \cdots, 1)^T \in R^{n_i \times 1}$, and the global centroid c is defined as

$$c = \frac{1}{n} Ae, \text{ where } e = (1, 1, \cdots, 1)^T \in R^{n \times 1}. \quad (3)$$

Define the matrices

$$\begin{aligned} H_w &= [A_1 - c^{(1)}(e^{(1)})^T, \cdots, A_k - c^{(k)}(e^{(k)})^T] \\ &\in R^{m \times n}, \\ H_b &= [\sqrt{n_1}(c^{(1)} - c), \cdots, \sqrt{n_k}(c^{(k)} - c)] \\ &\in R^{m \times k}. \end{aligned} \quad (4)$$

Then the scatter matrices S_w and S_b can be expressed as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T. \quad (5)$$

The traces of the two scatter matrices can be computed as follows,

$$\begin{aligned} \text{trace } (S_w) &= \sum_{i=1}^{k} \sum_{j \in N_i} (a_j - c^{(i)})^T (a_j - c^{(i)}) \\ &= \sum_{i=1}^{k} \sum_{j \in N_i} ||a_j - c^{(i)}||^2 \\ \text{trace } (S_b) &= \sum_{i=1}^{k} n_i (c^{(i)} - c)^T (c^{(i)} - c) \\ &= \sum_{i=1}^{k} n_i ||c^{(i)} - c||^2. \end{aligned} \quad (6)$$

Hence, trace (S_w) measures the closeness of the vectors within the clusters, while trace (S_b) measures the separation between clusters.

In the lower-dimensional space resulting from the linear transformation G^T, the within-cluster and between-cluster matrices become

$$\begin{aligned} S_w^L &= (G^T H_w)(G^T H_w)^T = G^T H_w H_w^T G \\ &= G^T S_w G, \\ S_b^L &= (G^T H_b)(G^T H_b)^T = G^T H_b H_b^T G \\ &= G^T S_b G. \end{aligned} \quad (7)$$

An optimal transformation G^T would maximize trace (S_b^L) and minimize trace (S_w^L). Common optimizations in classical discriminant analysis include

$$\begin{aligned} &\max_{G^T} \left\{ \text{trace}((S_w^L)^{-1} S_b^L) \right\} \text{ and} \\ &\min_{G^T} \left\{ \text{trace}((S_b^L)^{-1} S_w^L) \right\}. \end{aligned} \quad (8)$$

In the following, we will focus on the criterion of maximizing

$$F_0(G) = \text{trace}\left(\left(G^T S_w G \right)^{-1} \left(G^T S_b G \right) \right). \quad (9)$$

If we switch between S_w and S_b, the problem becomes a minimization problem. In classical discriminant analysis, S_w is assumed to be nonsingular, hence symmetric positive definite. It follows from linear algebra [8] that there is a nonsingular matrix $X \in R^{m \times m}$ such that

$$X^T S_b X = \Lambda = \text{diag}\,(\lambda_1, \cdots, \lambda_m)$$

and

$$X^T S_w X = I_m,$$

where

$$\lambda_1 \geq \cdots \geq \lambda_q > \lambda_{q+1} = \cdots = \lambda_m = 0.$$

We have

$$\begin{aligned} F_0(G) &= \text{trace}\left(\left(G^T S_w G \right)^{-1} \left(G^T S_b G \right) \right) \\ &= \text{trace}\left(\left(\tilde{G} \tilde{G}^T \right)^{-1} \left(\tilde{G} \Lambda \tilde{G}^T \right) \right), \end{aligned}$$

where $\tilde{G} = \left(X^{-1} G \right)^T$. Let $\tilde{G}^T = QR$ be the reduced QR factorization of $\tilde{G}^T$, where $Q \in R^{m \times \ell}$ has orthonormal columns and R is nonsingular (Note G^T has full row rank). Using the fact that trace(AB) = trace(BA), for any matrices A and B, we have

$$\begin{aligned} F_0(G) &= \text{trace}\left(\left(R^T R \right)^{-1} \left(R^T Q^T \Lambda Q R \right) \right) \\ &= \text{trace}\left(R^{-1} Q^T \Lambda Q R \right) \\ &= \text{trace}\left(Q^T \Lambda Q R R^{-1} \right) \\ &= \text{trace}\left(Q^T \Lambda Q \right) \leq \lambda_1 + \cdots + \lambda_q, \end{aligned}$$

where the inequality becomes equality for

$$Q = \begin{pmatrix} I_\ell \\ 0 \end{pmatrix} \text{ or } G = X \begin{pmatrix} I_\ell \\ 0 \end{pmatrix} R,$$

when the reduced dimension $\ell \geq q$. Note R is an arbitrary nonsingular matrix, hence the transformation G is not unique.

Note that a limitation of classical discriminant analysis in many applications involving small sample data, including text processing in information retrieval is that the matrix S_w must be nonsingular. In the next section, a new criterion is introduced that generalizes the classical discriminant and overcomes this limitation.

3 Generalization of discriminant analysis

Classical discriminant analysis expresses its solution by solving a generalized eigenvalue problem when S_b or S_w is nonsingular. However, for a general document matrix A, the number of document n may be smaller than its dimension m, then matrix H_w and H_b are not of full column rank, hence matrix S_w and S_b are both singular. In this paper, we define a new criterion F_1 below, where the non-singularity of the matrix S_w or S_b is not required. The new criterion aims to minimize the within-class distance, trace (S_w^L), and maximize the between-class distance, trace (S_b^L).

The new criterion F_1 is a natural extension of the classical one in Equation (8), where the inverse of a matrix is replaced by the pseudo-inverse [8]. While the inverse of a matrix may not exist, the pseudo-inverse of any matrix is always well defined. Moreover, when the matrix is invertible, its pseudo-inverse is the same as its inverse. F_1 is defined as

$$F_1(G) = \text{trace}\left((S_b^L)^+ S_w^L\right). \qquad (10)$$

Hence the trace optimization is to find an optimal transformation matrix G such that $F_1(G)$ is minimum under certain constraint defined in more detail in Section 3.3. We switch the roles between S_b^T and S_w^T in the F_1 criterion, compared with the F_0 criterion in classical discriminant analysis defined in Equation (9), since the value of trace $\left((S_w^L)^+ S_b^L\right)$ can be infinity.

We show how to solve the above minimization problem in Section 3.3. The main technique applied here is the GSVD, briefly introduced in Section 3.1. The constraints on the optimization problem are based on the observations in Section 3.2.

3.1 Generalized singular value decomposition

The Generalized Singular Value Decomposition (GSVD) was first introduced in [23]. A simple algorithm to compute GSVD can be found in [9], where the algorithm is due to [17].

Theorem 3.1 (Generalized Singular Value Decomposition) *Suppose two matrices $A \in R^{m\times n}$ and $B \in R^{p\times n}$ are given. Then for*

$$K = \begin{bmatrix} A \\ B \end{bmatrix} \text{ and } t = rank(K),$$

there exist orthogonal matrices

$$U \in R^{m\times m}, V \in R^{p\times p},$$

and a nonsingular matrix

$$X \in R^{n\times n}$$

such that

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^T KX = \begin{bmatrix} \Sigma_A & 0 \\ \Sigma_B & 0 \end{bmatrix},$$

where

$$\Sigma_A = \begin{bmatrix} I_A & 0 & 0 \\ 0 & D_A & 0 \\ 0 & 0 & 0_A \end{bmatrix},$$

$$\Sigma_B = \begin{bmatrix} 0_B & 0 & 0 \\ 0 & D_B & 0 \\ 0 & 0 & I_B \end{bmatrix}.$$

The matrices

$$I_A \in R^{r\times r}$$

and

$$I_B \in R^{(t-r-s)\times(t-r-s)}$$

are identity matrices, where

$$r = rank(K) - rank(B),$$

and

$$s = rank(A) + rank(B) - rank(K),$$

$$0_A \in R^{(m-r-s)\times(t-r-s)} \text{ and } 0_B \in R^{(p-t+r)\times r}$$

are zero matrices, and

$$D_A = diag(\alpha_{r+1}, \cdots, \alpha_{r+s}) \text{ and}$$

$$D_B = diag(\beta_{r+1}, \cdots, \beta_{r+s})$$

satisfy

$$1 > \alpha_{r+1} \geq \cdots \geq \alpha_{r+s} > 0,$$

$$0 < \beta_{r+1} \leq \cdots \leq \beta_{r+s} < 1,$$

and $\alpha_i^2 + \beta_i^2 = 1$ *for* $i = r+1, \cdots, r+s$. ∎

The GSVD on the matrix pair (H_b^T, H_w^T), will give orthogonal matrices $U \in R^{k\times k}$, $V \in R^{n\times n}$, and a nonsingular matrix $X \in R^{m\times m}$, such that

$$\begin{bmatrix} U & 0 \\ 0 & V \end{bmatrix}^T KX = \begin{bmatrix} \Sigma_1 & 0 \\ \Sigma_2 & 0 \end{bmatrix}. \quad (11)$$

where

$$\Sigma_1 = \begin{bmatrix} I_b & 0 & 0 \\ 0 & D_b & 0 \\ 0 & 0 & 0_b \end{bmatrix}, \quad \Sigma_2 = \begin{bmatrix} 0_w & 0 & 0 \\ 0 & D_w & 0 \\ 0 & 0 & I_w \end{bmatrix}.$$

Here $I_b \in R^{r\times r}$ is an identity matrix with

$$r = \text{rank}(K) - \text{rank}(H_w^T), \quad (12)$$

$$D_b = \text{diag}(\alpha_{r+1}, \cdots, \alpha_{r+s}), \text{ and}$$

$$D_w = \text{diag}(\beta_{r+1}, \cdots, \beta_{r+s}) \in R^{s\times s}$$

are diagonal matrices with

$$s = \text{rank}(H_b) + \text{rank}(H_w) - \text{rank}(K), \quad (13)$$

satisfying

$$1 > \alpha_{r+1} \geq, \cdots, \geq \alpha_{r+s} > 0,$$

$$0 < \beta_{r+1} \leq, \cdots, \leq \beta_{r+s} < 1,$$

and $\alpha_i^2 + \beta_i^2 = 1$ for $i = r+1, \cdots, r+s$.

From (11), we have

$$H_b^T X = U \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix}, H_w^T X = V \begin{bmatrix} \Sigma_2 & 0 \end{bmatrix}$$

hence

$$\begin{aligned} X^T H_b H_b^T X &= \begin{bmatrix} \Sigma_1^T \\ 0 \end{bmatrix} U^T U \begin{bmatrix} \Sigma_1 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_1^T \Sigma_1 & 0 \\ 0 & 0 \end{bmatrix} \equiv D_1, \\ X^T H_w H_w^T X &= \begin{bmatrix} \Sigma_2^T \\ 0 \end{bmatrix} V^T V \begin{bmatrix} \Sigma_2 & 0 \end{bmatrix} \\ &= \begin{bmatrix} \Sigma_2^T \Sigma_2 & 0 \\ 0 & 0 \end{bmatrix} \equiv D_2. \end{aligned}$$

Therefore

$$\begin{aligned} S_b^L &= G^T S_b G = G^T H_b H_b^T G \\ &= G^T (X^{-1})^T (X^T H_b H_b^T X) X^{-1} G \\ &= (X^{-1}G)^T D_1 X^{-1} G = \tilde{G} D_1 \tilde{G}^T, \\ S_w^L &= G^T S_w G = G^T H_w H_w^T G \\ &= G^T (X^{-1})^T (X^T H_w H_w^T X) X^{-1} G \\ &= (X^{-1}G)^T D_2 X^{-1} G = \tilde{G} D_2 \tilde{G}^T, \quad (14) \end{aligned}$$

where the matrix

$$\tilde{G} = (X^{-1}G)^T. \quad (15)$$

We will use the above representations for S_b^L and S_w^L in Section 3.3 for the minimizations of F_1.

3.2 Linear subspace spanned by the centroids

Let

$$\mathcal{C} = \text{span}\{c^{(1)}, \cdots, c^{(k)}\} \quad (16)$$

be a subspace in R^m spanned by the k centroids of the document vectors. In the lower dimensional space transformed by G^T, the linear subspace spanned by the k centroids in the reduced space is

$$\mathcal{C}_L = \text{span}\{c_L^{(1)}, \cdots, c_L^{(k)}\}, \quad (17)$$

where $c_L^{(i)} = G^T c^{(i)}$, for $i = 1, \cdots, k$.

In this section, we study the relation between the dimension of the subspace $\mathcal{C}$ and the rank of the matrix H_b, as well as the corresponding ones in the reduced space.

The main result is as follows:

Lemma 3.1 *Let $\mathcal{C}$ and $\mathcal{C}_L$ be defined as in (16) and (17) and H_b be defined as in (4). Let $dim(\mathcal{C})$ and $dim(\mathcal{C}_L)$ denote the dimensions of the subspaces $\mathcal{C}$, $\mathcal{C}_L$ respectively. Then*

1. *$dim(\mathcal{C}) = rank(H_b) + 1$, or $rank(H_b)$, and $dim(\mathcal{C}_L) = rank(G^T H_b) + 1$, or $rank(G^T H_b)$;*
2. *If $dim(\mathcal{C}) = rank(H_b)$, then $dim(\mathcal{C}_L) = rank(G^T H_b)$. If $dim(\mathcal{C}_L) = rank(G^T H_b) + 1$, then $dim(\mathcal{C}) = rank(H_b) + 1$.*

Proof Consider the matrix

$$S = \left[c^{(1)} - c, \cdots, c^{(k)} - c, c\right],$$

where the global centroid c as defined in (3) can be rewritten as

$$c = \sum_{i=1}^{k} \frac{n_i}{n} c^{(i)}.$$

It is easy to check $\text{rank}(S) = \dim(\mathcal{C})$. On the other hand, the rank of the matrix S equals to $\text{rank}(H_b)$, if c lies in the space spanned by $\{c^{(1)} - c, \cdots, c^{(k)} - c\}$, or $\text{rank}(H_b) + 1$, otherwise. Hence $\dim(\mathcal{C}) = \text{rank}(H_b) + 1$, or $\text{rank}(H_b)$. Similarly, we can prove $\dim(\mathcal{C}_L) = \text{rank}(G^T H_b) + 1$, or $\text{rank}(G^T H_b)$. The completes the proof for part 1.

If $\dim(\mathcal{C}) = \text{rank}(H_b)$, from the above argument, $c = \sum_{i=1}^{k} \gamma_i (c^{(i)} - c)$ for some coefficients γ_i. It follows that

$$c_L = G^T c = \sum_{i=1}^{k} \gamma_i (G^T c^{(i)} - G^T c) = \sum_{i=1}^{k} \gamma_i (c_L^{(i)} - c_L),$$

where c_L is the centroid in the lower dimensional space. Again by the above argument, $\dim(\mathcal{C}_L) = \text{rank}(G^T H_b)$.

Similarly, if $\dim(\mathcal{C}_L) = \text{rank}(G^T H_b) + 1$, we can show $\dim(\mathcal{C}) = \text{rank}(H_b) + 1$. ∎

3.3 Generalized discriminant analysis using F_1 measure

We start with a more general optimization problem as follows,

$$\min_G F_1(G) \text{ subject to } \text{rank}(G^T H_b) = \delta, \quad (18)$$

for some $\delta > 0$. The optimization in Equation (18) depends on the value of δ. The optimal transformation G we are looking for in this section is a special case of the above formulation, where we set $\delta = \text{rank}(H_b)$. This choice of δ guarantees that the dimension of the linear space spanned by the centroids in the original high dimension space and the corresponding one in the transformed lower dimensional space as defined in Equation (16) and Equation (17) are kept close to each other, as shown in the following Proposition 3.1.

To solve the minimization problem in (18), we need the following three lemmas, where the proof of the first two lemmas are straightforward from the definition of the pseudo-inverse [8].

Lemma 3.2 *For any matrix $A \in R^{m \times n}$, we have $trace(AA^+) = rank(A)$.*

Lemma 3.3 *For any matrix $A \in R^{m \times n}$, we have $(AA^T)^+ = (A^+)^T A^+$.*

The following Lemma is critical for our main result,

Lemma 3.4 *Let $\Sigma = diag(\sigma_1, \cdots, \sigma_u)$ be any diagonal matrix with $\sigma_1 \geq \cdots \geq \sigma_u > 0$. Then for any matrix $M \in R^{v \times u}$ with $rank(M) = \delta$, the following inequality holds.*

$$trace\left(\left(M\Sigma M^T\right)^+ MM^T\right) \geq \sum_{i=1}^{\delta} \frac{1}{\sigma_i}$$

Furthermore, the equality holds if and only if

$$M = U \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix},$$

for some orthogonal matrix $U \in R^{v \times v}$ and matrix $D = \Sigma^1 Q \Sigma^2 \in R^{\delta \times \delta}$, where $Q \in R^{\delta \times \delta}$ is orthogonal and $\Sigma^1, \Sigma^2 \in R^{\delta \times \delta}$ are diagonal matrices with positive diagonal entries.

Proof It is easy to check that

$$\begin{aligned}
&\left(M\Sigma M^T\right)^+ MM^T \\
&= \left(M\Sigma^{1/2}\Sigma^{1/2}M^T\right)^+ M\Sigma^{1/2}\Sigma^{-1}\Sigma^{1/2}M^T \\
&= \left(ZZ^T\right)^+ Z\Sigma^{-1}Z^T \\
&= (Z^+)^T Z^+ Z\Sigma^{-1}Z^T, \qquad (19)
\end{aligned}$$

where $Z = M\Sigma^{1/2}$ and the last equality follows from Lemma 3.3.

It is clear that $\mathrm{rank}(Z) = \mathrm{rank}(M) = \delta$, since Σ is non-singular. Let

$$Z = U \begin{pmatrix} \tilde{\Sigma} & 0 \\ 0 & 0 \end{pmatrix} V^T$$

be the SVD of Z. Then by the definition of pseudo-inverse,

$$Z^+ = V \begin{pmatrix} \tilde{\Sigma}^{-1} & 0 \\ 0 & 0 \end{pmatrix} U^T.$$

Define the matrix $Y = Z^+Z$, then

$$Y = V \begin{pmatrix} I_\delta & 0 \\ 0 & 0 \end{pmatrix} V^T,$$

where $I_\delta \in R^{\delta\times\delta}$ is an identity matrix.

Using the fact that $\mathrm{trace}(AB) = \mathrm{trace}(BA)$, for any two matrices A and B, together with (19), we have

$$\begin{aligned}
&\mathrm{trace}\left(\left(M\Sigma M^T\right)^+ MM^T\right) \\
=\ &\mathrm{trace}\left((Z^+)^T Z^+ Z\Sigma^{-1}Z^T\right) \\
=\ &\mathrm{trace}\left(Z^+Z\Sigma^{-1}(Z^+Z)^T\right) \\
=\ &\mathrm{trace}\left(Y\Sigma^{-1}Y^T\right) \\
=\ &\mathrm{trace}\left(\begin{pmatrix} I_\delta & 0 \\ 0 & 0 \end{pmatrix} V^T\Sigma^{-1}V \begin{pmatrix} I_\delta & 0 \\ 0 & 0 \end{pmatrix}\right) \\
=\ &\mathrm{trace}\left(\begin{pmatrix} V_\delta^T \\ 0 \end{pmatrix} \Sigma^{-1} \begin{pmatrix} V_\delta & 0 \end{pmatrix}\right), \\
=\ &\mathrm{trace}\left(\begin{pmatrix} V_\delta^T\Sigma^{-1}V_\delta & 0 \\ 0 & 0 \end{pmatrix}\right), \\
=\ &\mathrm{trace}\left(V_\delta^T\Sigma^{-1}V_\delta\right) \\
\geq\ &\sum_{i=1}^{\delta} \frac{1}{\sigma_i}, \qquad (20)
\end{aligned}$$

where V_δ contains the first δ columns of the orthogonal matrix V. The last inequality in (20) follows, since the diagonal elements of the diagonal matrix Σ^{-1} is nondecreasing. Moreover, the minimum of trace $\left(V_\delta^T\Sigma^{-1}V_\delta\right)$ is obtained if and only if

$$V_\delta = \begin{pmatrix} Q_\delta \\ 0 \end{pmatrix},$$

for some orthogonal matrix $Q_\delta \in R^{\delta\times\delta}$. Hence

$$\begin{aligned}
M &= Z\Sigma^{-1/2} = U \begin{pmatrix} \tilde{\Sigma} & 0 \\ 0 & 0 \end{pmatrix} V^T\Sigma^{-1/2} \\
&= U \begin{pmatrix} \tilde{\Sigma}Q_\delta\Sigma_\delta^{-1/2} & 0 \\ 0 & 0 \end{pmatrix} \\
&= Q \begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix}.
\end{aligned}$$

where Σ_δ is the δth principle sub-matrix of Σ, and $D = \tilde{\Sigma}Q_\delta\Sigma_\delta^{-1/2}$. ∎

We are now ready to present our main result for this section. To solve the minimization problem in (18), we first give a lower bound on the objective function F_1 in Theorem 3.2. A sufficient condition on G, under which the lower bound computed in Theorem 3.2 is obtained, is presented in Corollary 3.1. A simple solution is then presented in Corollary 3.2.

The optimal solution for our generalized discriminant analysis presented in this paper is a special case of the solution in Corollary 3.2, where we set $\delta = \mathrm{rank}(H_b)$.

Theorem 3.2 *Assume the transformation matrix $G^T \in R^{\ell\times m}$ satisfies rank$(G^TH_b) = \delta$, for some integer $\delta > 0$, then the following inequality holds,*

$$\begin{aligned}
F_1 &= trace\left((S_b^L)^+(S_w^L)\right) \\
&\geq \begin{cases} \sum_{i=r+1}^{\delta}\left(\frac{\beta_i}{\alpha_i}\right)^2 & if\ \delta \geq r+1 \\ 0 & if\ \delta < r+1, \end{cases} \qquad (21)
\end{aligned}$$

where $S_b^L = G^tS_bG$ and $S_w^L = G^TS_wG$ are defined in (7).

Proof First consider the easy case when $\delta < r+1$. Since both $(S_b^L)^+$ and S_w^L are semi-positive definite,

$$\text{trace}\left((S_b^L)^+(S_w^L)\right) \geq 0.$$

Next consider the case when $\delta \geq r+1$.

Recall from Equation (14) that by the GSVD,

$$S_b^L = \tilde{G} D_1 \tilde{G}^T, \quad S_w^L = \tilde{G} D_2 \tilde{G}^T,$$

where $\tilde{G}$ is defined in (15). Partition $\tilde{G}$ into

$$\tilde{G} = \begin{pmatrix} G_1, & G_2, & G_3, & G_4 \end{pmatrix},$$

such that $G_1 \in R^{\ell \times r}$, $G_2 \in R^{\ell \times s}$, $G_3 \in R^{\ell \times (t-r-s)}$, and $G_4 \in R^{\ell \times (m-t)}$.

Let $G_{123} = \begin{pmatrix} G_1, & G_2, & G_3 \end{pmatrix}$. Since D_1 and D_2 are diagonal matrices and the last $m-t$ diagonal entries are zero, it follows that

$$\begin{aligned} S_b^L &= G_{123}\Sigma_1^T\Sigma_1 G_{123}^T, \\ S_w^L &= G_{123}\Sigma_2^T\Sigma_2 G_{123}^T \end{aligned} \tag{22}$$

Since $\alpha_i^2 + \beta_i^2 = 1$, for $i = 1, 2, \cdots, t$, we have

$$\Sigma_1^T\Sigma_1 + \Sigma_2^T\Sigma_2 = I_t,$$

here $I_t \in R^{t \times t}$ is an identity matrix. Therefore,

$$S_w^L + S_b^L = G_{123}G_{123}^T = G_{12}G_{12}^T + G_3G_3^T. \tag{23}$$

Define $G_{12} = \begin{pmatrix} G_1, & G_2 \end{pmatrix}$. It follows from Equation (22) and Equation (23) that

$$\begin{aligned} & \text{trace}\left((S_b^L)^+(S_w^L + S_b^L)\right) \\ = \ & \text{trace}\left((S_b^L)^+(G_{123}G_{123}^T)\right) \\ = \ & \text{trace}\left(\left(G_{123}\Sigma_1\Sigma_1^T G_{123}^T\right)^+ G_{123}G_{123}^T\right) \\ = \ & \text{trace}\left(\left(G_{12}\begin{pmatrix} I_b & 0 \\ 0 & D_b^2 \end{pmatrix} G_{12}^T\right)^+ G_{123}G_{123}^T\right) \\ \geq \ & \text{trace}\left(\left(G_{12}\Sigma G_{12}^T\right)^+ G_{12}G_{12}^T\right) \\ \geq \ & \sum_{i=1}^{r} 1 + \sum_{i=r+1}^{\delta} \frac{1}{\alpha_i^2}, \end{aligned} \tag{24}$$

where

$$\Sigma = \begin{pmatrix} I_b & 0 \\ 0 & D_b^2 \end{pmatrix}.$$

The first inequality follows, since $G_3G_3^T$ is positive semi-definite, and the equality holds if $G_3 = 0$. The second inequality follows from Lemma 3.4 and the equality holds if and only if

$$G_{12} = U\begin{pmatrix} D & 0 \\ 0 & 0 \end{pmatrix} \in R^{\ell \times r+s}, \tag{25}$$

for some orthogonal matrix $U \in R^{\ell \times \ell}$ and some matrix

$$D = \Sigma^1 Q \Sigma^2 \in R^{\delta \times \delta},$$

where

$$\Sigma^1, \Sigma^2 \in R^{\delta \times \delta}$$

are diagonal matrices with positive diagonal entries and $Q \in R^{\delta \times \delta}$ is orthogonal. Especially it is true if both U and D are identity matrices.

By the property of the pseudo-inverse in Lemma 3.2,

$$\begin{aligned} \text{trace}((S_b^L)^+ S_b^L) &= \text{rank}(S_b^L) = \text{rank}(G^T S_b G) \\ &= \text{rank}(G^T H_b) = \delta. \end{aligned} \tag{26}$$

Therefore,

$$\begin{aligned} F_1 &= \text{trace}((S_b^L)^+ S_w^L) \\ &= \text{trace}((S_b^L)^+(S_w^L + S_b^L)) - \text{trace}((S_b^L)^+ S_b^L) \\ &\geq \sum_{i=1}^{r} 1 + \sum_{i=r+1}^{\delta} \frac{1}{\alpha_i^2} - \delta \\ &= \sum_{i=r+1}^{\delta} \left(\frac{\beta_i}{\alpha_i}\right)^2, \end{aligned} \tag{27}$$

where the inequality follows from (24). ∎

Theorem 3.2 gives a lower bound on F_1, when δ is fixed. From the arguments above, all the inequalities become equality if G_{12} has the form as in (25), and $G_3 = 0$, especially when U and D are identity matrices, which is used in our implementation later. The result is summarized in the following Corollary.

Corollary 3.1 *Let G^T, $\tilde{G}$ be defined as in Theorem 3.2 and $\delta > r+1$, then the equality*

$$F_1 = \textit{trace}((S_b^L)^+ S_w^L) = \sum_{i=r+1}^{\delta} \left(\frac{\beta_i}{\alpha_i}\right)^2$$

holds, under the condition that the partition of

$$\tilde{G} = \begin{pmatrix} G_1, & G_2, & G_3, & G_4 \end{pmatrix}$$

satisfies

$$G_{12} = \begin{pmatrix} I_\delta & 0 \\ 0 & 0 \end{pmatrix}$$

and

$$G_3 = 0,$$

where

$$G_{12} = \begin{pmatrix} G_1, & G_2 \end{pmatrix}.$$

Theorem 3.2 does not mention the connection between ℓ, the row dimension of the transformation matrix G^T and δ. We can choose the transformation matrix G^T with large row dimension ℓ and still satisfies the condition stated in Corollary 3.1. However we are more interested in a lower dimensional representation of the original, while keeping the same information from the original data.

The following Corollary says the smallest possible value for ℓ is δ, and more importantly we can find a transformation G^T with its row dimension equal δ, which also satisfies the condition stated in Corollary 3.1.

Corollary 3.2 *For every $\delta \leq r + s$, there exists a transformation $G^T \in R^{\delta \times m}$, such that the equality in (21) holds, i.e. the minimum value for F_1 is obtained. Furthermore for any transformation $(G')^T \in R^{\ell \times m}$, such that the assumption in Theorem 3.2 holds, we have $\ell \geq \delta$.*

Proof Construct a transformation $G^T \in R^{\delta \times m}$, such that

$$\tilde{G} = (X^{-1}G)^T = \begin{pmatrix} G_{12}, & G_3, & G_4 \end{pmatrix}$$

satisfies $G_{12} = \begin{pmatrix} I_\delta & 0 \end{pmatrix}$, $G_3 = 0$, and $G_4 = 0$, where $I_\delta \in R^{\delta \times \delta}$ is an identity matrix. Hence $G^T = X_\delta^T$, where $X_\delta \in R^{\delta \times m}$ contains the first δ columns of the matrix X. By Corollary 3.1, the equality in (21) holds under the above transformation G^T. This complete the proof for the first part.

For any transformation $(G')^T \in R^{\ell \times m}$, such that $\text{rank}((G')^T H_b) = \delta$, it is clear

$$\delta \leq \text{rank}(G')^T) \leq \ell.$$

Hence $\ell \geq \delta$. ∎

Remark 3.1 Theorem 3.2 shows the minimum value of the objective function F_1 is dependent on the rank of the matrix $G^T H_b$. As shown in Lemma 3.1, the rank of the matrix H_b, which is $r+s$ by GSVD, denotes the degree of linear independence of the k centroids in the original data, while the rank δ of the matrix $G^T H_b$ implies the degree of linear independence of the k centroids in the reduced space by Lemma 3.1. By Corollary 3.2, if we fix the degree of linear independence of the centroids in the reduced space, i.e. if $\text{rank}(G^T H_b) = \delta$ is fixed, then we can always find a transformation matrix $G^T \in R^{\delta \times m}$ with row dimension δ such that the minimum value for F_1 is obtained.

Next we consider the case when δ varies. From the result in Theorem 3.2, the value of the objective function F_1 is smaller for smaller δ. However, as the value of δ decreases (Note δ is always no larger than $r + s$), the degree of linear independence of the k centroids in the reduced space is a lot less than the one in the original high dimensional space, which may lead to information loss of the original data. Hence, we choose $\delta = \text{rank}(H_b)$, its maximum possibility, in our implementation. One nice property of this choice is stated in the following Proposition:

Proposition 3.1 *If $\delta \equiv$ rank$(G^T H_b)$ equals to rank(H_b), then $|\mathcal{C}| - 1 \leq |\mathcal{C}_L| \leq |\mathcal{C}|$.*

Proof The proof follows directly from Lemma 3.1. ∎

Proposition 3.1 implies choosing $\delta = \text{rank}(H_b)$ keeps the same or one less degree of linear independence of the centroids in the reduced space as the one in the original space. With the above choice, the reduced dimension under the transformation G^T is $\text{rank}(H_b)$. We use $\delta^\star = \text{rank}(H_b)$ to denote the optimal reduced dimension for our generalized discriminant analysis (also called exact algorithm) throughout the rest of the paper.

Algorithm 1: Exact algorithm
1. Form the matrices H_b and H_w as in Eq. (4).
2. Compute GSVD on the matrix pair (H_b^T, H_w^T) to obtain the matrix X as in Eq. (11),
3. $\delta^\star \leftarrow \text{rank}(H_b)$.
4. $G^T \leftarrow X_{\delta^\star}^T$, where $X_{\delta^\star}$ contains the first $\delta^\star$ columns of the matrix X.

In this case, we can choose $G^T = X_{\delta^\star}^T$, where $X_{\delta^\star}$ contains the first $\delta^\star$ columns of the matrix X as in Corollary 3.2. The pseudo-code for our main algorithm is shown in **Algorithm 1**.

In principle, it is not necessary to choose the reduced dimension to be exactly $\delta^\star$. Experiments show good performance for all values of reduced dimension close to $\delta^\star$. However the results also show the performance can be very poor for very small value of reduced dimension, probably due to information loss of original data.

4 Approximation algorithm

One of the limitations of the above method is the expensive computation of the generalized singular value decomposition of the matrix $K \in R^{n+k\times m}$. For large text document data, both n and m can be large, hence the above method may not be applicable. In this section, an efficient approximation algorithm is presented to overcome this limitation.

The K-Means algorithm [4, 12] is widely used to capture the structure of the scattered data, by decomposing the whole data set as a disjoint union of small sets, called clusters. The K-Means algorithm aims to minimize the the distance within each cluster, hence the centroid of the each cluster represents well the data points in the same cluster, while the centroids of the resulting clusters give a good approximation of the original data set.

In Section 2, we use the matrix H_w to capture the closeness of the vectors within each cluster. However, the dimension of $H_w \in R^{m\times n}$ is very high, since we use every point in the document data. To simplify the model, we attempt to use the centroids only to approximate the structure of each cluster, by apply K-Means algorithm to each cluster.

More specifically, if $(\pi_1, \pi_2, \cdots, \pi_k)$ are the k clusters in the text document data, with the size of each cluster $|\pi_i| = n_i$, and $\sum_{i=1}^k n_i = n$, K-Means algorithm is applied to each cluster π_i to produce s_i sub-clusters $\{\pi_i^{(j)}\}_{j=1}^{s_i}$, with $\pi_i = \cup_{j=1}^{n_i} \pi_i^{(j)}$ and the size of each sub-cluster $|\pi_i^{(j)}| = n_i^j$. Let $c_i^{(j)}$ be the centroid for each sub-cluster $\pi_i^{(j)}$. The within cluster distance in the ith cluster

$$\sum_{j\in\pi_i} ||a_j - c^{(i)}||^2$$

can be approximated as

$$\sum_{u=1}^{s_i} \sum_{a_j\in\pi_i^{(u)}} ||a_j - c^{(i)}||^2 \sim \sum_{u=1}^{s_i} n_i^u ||c_i^{(u)} - c^{(i)}||^2,$$

by approximating every point $a_j$in the sub-cluster $\pi_i^{(u)}$ by its centroid $c_i^{(u)}$.

Hence the matrix H_w can be approximated as

$$\begin{aligned}[\sqrt{n_1^1}(c_1^{(1)} - c^{(1)}), \quad \cdots \quad , \sqrt{n_1^{s_1}}(c_1^{(s_1)} - c^{(1)}), \cdots, \\ \sqrt{n_k^1}(c_k^{(1)} - c^{(k)}), \quad \cdots \quad , \sqrt{n_k^{s_k}}(c_k^{(s_k)} - c^{(k)})] \\ \in \quad R^{m\times M}, \end{aligned} \tag{28}$$

where $M = \sum_{i=1}^k s_i$ is the total number of centroids, which is typically much smaller than n, the total number of data points in the original text document data, thus reducing the complexity of the GSVD computation dramatically. The main steps for our approximation algorithm in summarized in **Algorithm 2**. For simplicity, in our implementation we chose all the s_i's to have the same value s'. We discuss below the choice for s'.

To test its efficacy, we have applied the approximation algorithm to numerous data sets. Experiments show the approximation algorithm produces similar results to the exact ones.

4.1 The value for s'

The number of sub-clusters s' within each cluster π_i will determine the complexity of the approximation algorithm. If s' is too large, the approximation algorithm will produce results close to the

Algorithm 2: Approximation algorithm

1. Form the matrix H_b as defined in Eq. (4).
2. Run K-Means algorithm on each π_i with $s_i = s'$.
3. Form the matrix $\tilde{H}_w$ as defined in Eq. (28) to approximate H_w.
4. Compute GSVD on the matrix pair $(H_b^T, \tilde{H}_w^T)$ to obtain the matrix X as in Eq. (11).
3. $\delta^\star \leftarrow \text{rank}(H_b)$.
4. $G^T \leftarrow X_{\delta^\star}^T$, where $X_{\delta^\star}$ contains the first $\delta^\star$ columns of the matrix X.

one using all the points, while the computation of the GSVD will still be expensive. For our problem, we only apply the K-Means algorithm to the data points belonging to the same cluster of the original document set, which are already close to each other. Indeed, in our experiments, we found that small values of s' worked well; in particular, choosing s' around 6 to 10 gave good results.

5 Comparison between different dimension reduction methods

Latent Semantic Indexing (LSI) is a common dimension reduction method for text document data. Using truncated SVD algorithm, the document data is presented in a lower dimensional "topic" space: the documents are characterized by some underlining hidden concepts referred to by the terms. The choice of reduced dimension N is a serious problem for LSI. Experiments in next section show for small N, the results can be very bad.

We also consider the extreme case, when we use the full document matrix, i.e., the reduced dimension N is the maximum possible one. We use "Full" to distinguish this from other methods. While there is no decomposition time for this method, its query time $O(nmK)$ and storage requirement $O(nm)$ is much higher than other methods, when using the K nearest neighbors for query.

The orthogonal centroid method (OCM) in [18], maximize the separation between clusters. It solves the following optimization problem,

$$\text{Find } G^T \in R^{\ell \times m} \text{ with orthonormal columns}$$
$$\text{such that trace}(G^T S_b G) \text{ is maximum.}$$

As shown in [18], the above optimization problem can be solved efficiently by computing QR decomposition of the centroid matrix $C = [c^1 \quad c^2 \quad \cdots \quad c^k] \in R^{m \times k}$, where c^i, for $i = 1, \cdots, k$ are the centroids of the k clusters in the document data as introduced in section 2. More specifically, G^T is chosen to be Q^T, where $C = QR$ is the QR-decomposition of the centroid matrix C. Hence the reduced dimension $\ell = \text{rank}(C) = k$, if we assume the k centroids are linearly independent. The decomposition time for this method is $O(k^2 m)$, since C is $m \times k$ matrix.

The exact algorithm introduced in Section 3 computes a generalized singular value decomposition, of an $(n+k) \times m$ matrix. If we assume $n + k < m$, the decomposition time would be $O(n^2 m)$ [9], which has the same complexity as LSI when the SVD of the full matrix is computed. However the query time and storage are much less than those for LSI. Specifically, the optimal reduced dimension $\delta^\star = \text{rank}(H_b) \leq k - 1$, hence a single query using the K nearest neighbors only takes $O(n\delta^\star K) = O(nkK)$ time. More importantly, the experiments in next section show this method produces much better result than LSI.

To reduce the complexity for the decomposition step, an approximation algorithm is presented in Section 4. The ith cluster is represented by the s_i centroids. Hence the number of rows in the approximate H_w matrix is

$$k + (s_1 + \cdots + s_k) = k + ks' = O(ks').$$

The complexity of the GSVD computation using the approximate H_w matrix is reduced to $O\left((k + ks')^2 m\right) = O(k^2 s'^2 m)$. Note, the K-Means algorithm for the clustering is very fast. Within every iteration, the computation complexity is linear on the number of vectors and the number of clusters, while the algorithm converges within few iteration. Hence the complexity

Method	decom. time	query time	Storage
Full	0	$O(nmK)$	$O(mn)$
LSI	$O(n^2m)$	$O(nNK)$	$O(Nn)$
OCM	$O(k^2m)$	$O(nkK)$	$O(kn)$
Exact	$O(n^2m)$	$O(nkK)$	$O(kn)$
Appr.	$O(nm)$	$O(nkK)$	$O(kn)$

Table 1: Comparison of decomposition, query time and storage requirement

of the K-Means step for cluster i is $O(s_i n_i m) = O(s' n_i m)$, where n_i is the size of the ith cluster. Therefore, the total time for all the k clusterings is

$$O\left(\sum_{i=1}^{k} s' n_i m\right) = O\left(s'\left(\sum_{i=1}^{k} n_i\right) m\right) = O(s' nm).$$

Hence the total complexity for our approximation algorithm is $O(k^2 s'^2 m + s' nm)$. Since s' is usually chosen to be a small number and k is much smaller than n, the complexity is simplified to be $O(nm)$.

Note, since the K-Means clustering for each cluster is independent to each other, we can take advantage of this by applying K-Means for all the clusters in parallel, hence further reducing the running time complexity. Moreover for the situation when the memory can't hold all the data points simultaneously, the above approximation algorithm can process every cluster individually, thus also reduce the memory requirement.

Table 1 lists the complexity of different methods discussed above. K-nearest neighbor algorithm is used for document query. In the experiments in next section, we choose three different numbers for K, i.e. $K = 1$, 7, and 15. Note, for LSI, we compute SVD for the full matrix, hence the complexity of the decomposition step is $O(n^2 m)$. However, if the reduced dimension N is small, the complexity can be lower.

6 Experimental results

6.1 Datasets

In the following experiments, we used three different datasets, summarized in Table 2. For all datasets, we used a stop-list to remove common words, and the words were stemmed using Porter's suffix-stripping algorithm [19]. Moreover, any term that occurs in fewer than two documents was eliminated as in [24]. Dataset 1 is derived from the TREC-5, TREC-6, and TREC-7 collections [22]. It consist of 210 documents in a space of dimension of 7454, with 7 classes. Each class has 30 documents. Datasets 2 and 3 are from *Reuters-21578* text categorization test collection Distribution 1.0 [16]. Dataset 2 contains 4 classes, with each containing 80 elements. The dimension of the second document set is 2887. Dataset 3 has 5 classes, each with 98 elements. The dimension of the document set is 3759. (These datasets are available at `www.cs.umn.edu/~jieping/Research.html` .)

For all the examples, we used the *tf-idf* weighting scheme [20, 24]. More specifically, let

$$d_{tf} = (tf_1, tf_2, \cdots, tf_m)$$

be the term-frequency vector for one of the documents, where tf_i is the frequency of the ith term in the document. Then the *tf-idf* representation of this document is

$$d_{tf\text{-}idf} = (tf_1 \log(N/df_1), \cdots, tf_m \log(N/df_m)),$$

where N is the total number of documents in the collection, and df_i is the number of documents that contains the ith term, i.e. the document frequency. Finally, to account for documents of different lengths, the length of each document vector is normalized so that it is of unit length.

6.2 Experimental methodology

To evaluate the proposed methods in this paper, we compared them with the other dimension reduction methods discussed in Section 5 on the three datasets. The K-Nearest Neighbor algorithm (for $K = 1, 7, 15$) [6] was applied to evaluate the quality of different dimension-reduction

Data Set	1	2	3
Source	TREC	Reuters-21578	
# of documents	210	320	490
# of terms	7454	2887	3759
# of classes	7	4	5

Table 2: Summary of datasets used for evaluation

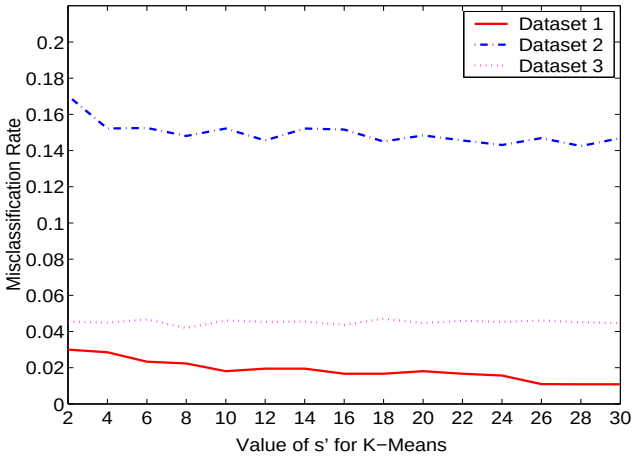

Figure 1: Effect of s' on the approximation algorithm using Datasets 1–3

algorithms as in [18, 9]. For each method, we applied 10-fold cross validation to compute misclassification rate. For LSI, the results depend on the choice of reduced dimension N. In the following experiments, we applied LSI on two different values of N. One of them equals the optimal reduced dimension used in the exact algorithm. While LSI does not work well if the reduced dimension N is small, and the optimal dimension in our exact algorithm is typically very small, we also apply LSI on a much larger reduced dimension $N = 100$.

The clustering using the K-Means in the approximation algorithm is sensitive to the choices of the initial centroids. To mitigate this, we ran the algorithm 10 times, and the initial centroids for each run were generated randomly. The final result is the average over the 10 different runs.

6.3 Results

6.3.1 Effect of s' on the approximation algorithm

As mentioned in Section 4, our approximation algorithm worked well for small values of s'. We tested our approximation algorithm on Datasets 1–3 for different values of s', ranging from 2 to 30, and computed the misclassification rates. As seen from Figure 1, the rates did not fluctuate very much within the range. In our experiments, we chose $s' = 8$. (All figures in this paper may be viewed in color at `www.cs.umn.edu/~jieping/Research.html` .)

6.3.2 Comparison of misclassification rates

We made a comparison of our exact and approximation algorithms with other competing algorithms based on the misclassification rates, using the three datasets in Table 2.

The results for Datasets 1–3 are summarized in Figure 2–Figure 4, respectively, where the x-axis is three different choices of K ($K = 1, 7, 15$) used in KNN for classification, and the y-axis is the misclassification rate. For each K, the misclassification rates for different methods ("FULL", "LSI-1","LSI-2", "OCM", "EXACT", and "APPR") are ordered from left to right. Here "FULL" is the method without any dimension algorithm, "LSI-1" is the Latent Semantic Indexing algorithm with reduced dimension $N = \ell$, where ℓ is the optimal reduced dimension in our exact algorithm, and "LSI-2" is the Latent Semantic Indexing algorithm with the reduced dimension $N = 100$. "OCM" is the Orthogonal Centroid Method. "EXACT" and "APPR" are the exact and approximation algorithms proposed in this paper.

In Dataset 1, since the number of terms (7454) in the term document matrix is larger than the number of documents (210), both S_w and S_b are singular and classical discriminant analysis breaks down. However our proposed generalized

discriminant analysis circumvents this problem. Our exact and approximation algorithm reduce the dimension $m = 7454$ to $\delta^\star = 6$, while the orthogonal centroid method (OCM) reduces the dimension to seven. We applied LSI with the reduced dimension $N = 6$ and $N = 100$. As shown in Figure 2, our exact and approximation algorithms work better than the other methods, while the results produced by the approximation algorithm are fairly close to those of the exact one. Figure 2 also shows better performance of OCM over LSI and Full.

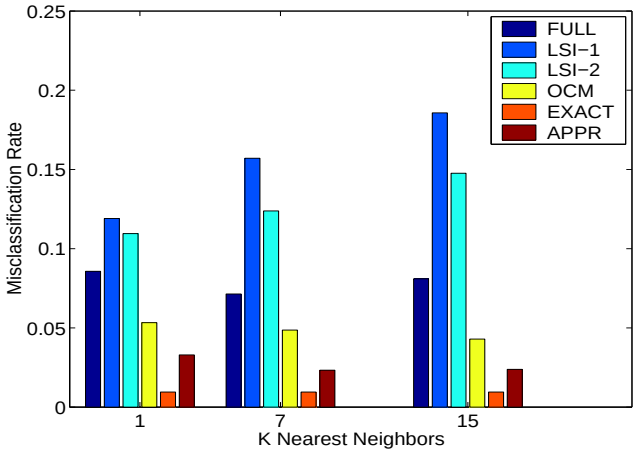

Figure 2: Performance of different dimension reduction methods using Dataset 1

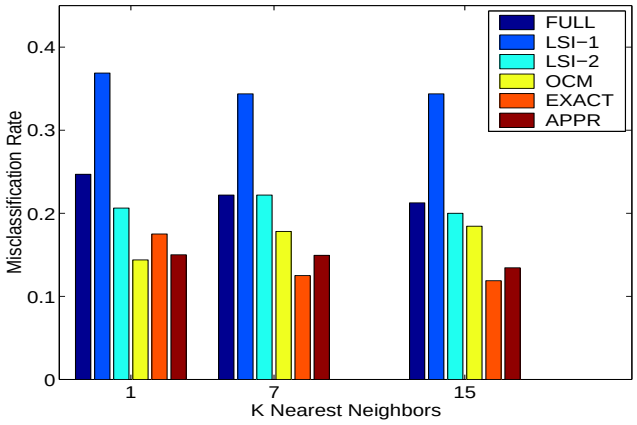

Figure 3: Performance of different dimension reduction methods using Dataset 2

For Datasets 2 and 3, the number of documents

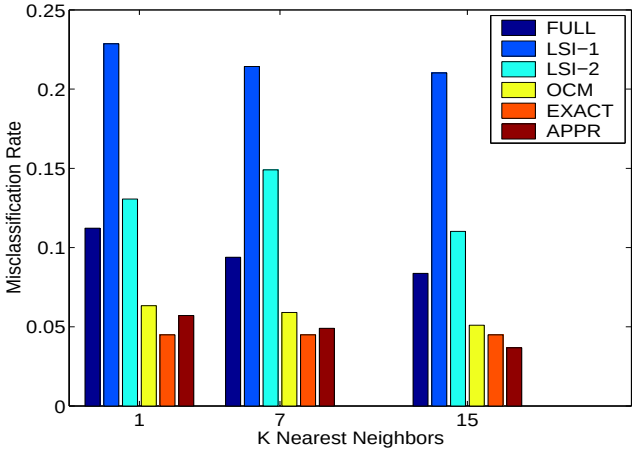

Figure 4: Performance of different dimension reduction methods using Dataset 3

is smaller than the document dimension, hence classical discriminant analysis again breaks down. Again, the results (Figure 3 and Figure 4) show that our exact and approximation algorithms perform better than the other methods. More interestingly, the approximation algorithm works almost as well as the exact one.

6.3.3 Effect of the reduced dimension on the exact algorithm and on LSI

As is well known, the choice of the reduced dimension is a serious problem for LSI. In Section 3, we show our exact algorithm using generalized discriminant analysis has optimal reduced dimension $\delta^\star$, which equals the rank of the matrix H_b and is typically very small. We also mentioned in Section 3, that our exact algorithm may not work very well if the reduced dimension is chosen to be much smaller than the optimal one.

Figures 5–7 illustrate the effect of different choices for the reduced dimension on our exact algorithm and on LSI. $K = 7$ nearest neighbors have been used for classification. The x-axis is the value for the reduced dimension, and the y-axis is the misclassification rate. For each reduced dimension, the results for our exact algorithm and LSI are ordered from left to right. Our exact algorithm outperforms LSI in almost all cases, especially for reduced dimension around

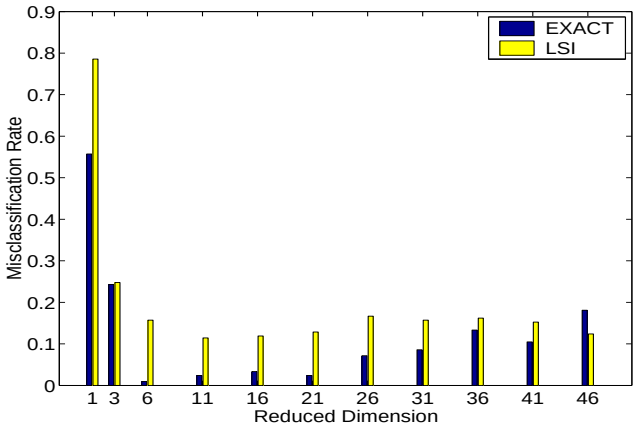

Figure 5: Comparison between our exact algorithm (optimal reduced dimension $\delta^\star = 6$) and LSI on different values of reduced dimension using Dataset 1

$\delta^\star$. The results also show and further confirm our theoretical results on the optimal choice of the reduced dimension for our exact algorithm as discussed in Section 3.

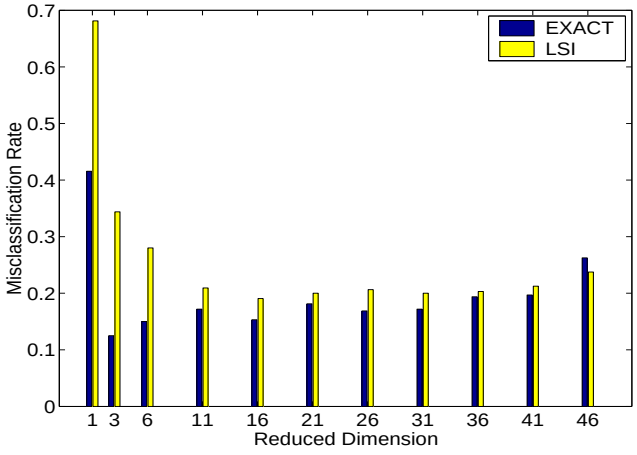

Figure 6: Comparison between our exact algorithm (optimal reduced dimension $\delta^\star = 3$) and LSI on different values of reduced dimension using Dataset 2

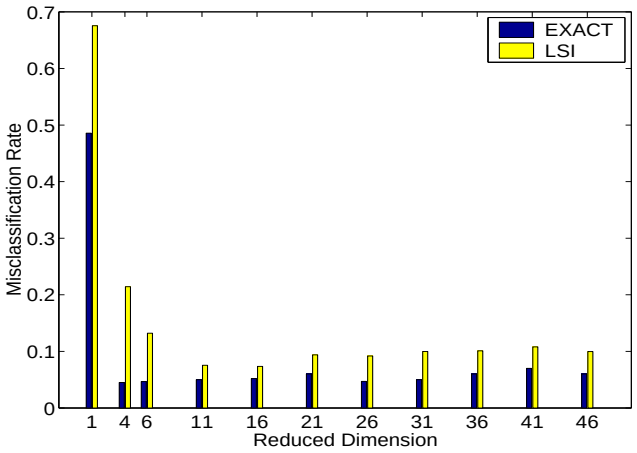

Figure 7: Comparison between our exact algorithm (optimal reduced dimension $\delta^\star = 4$) and LSI on different values of reduced dimension using Dataset 3

6.3.4 Class visualization of high-dimensional data in 2D

Our final experiment is to visualize the high-dimensional data by projecting onto the 2-dimensional plane as done in [5]. We extracted 240 data points, from three distinct classes in Dataset 2. Our exact and approximation dimension reduction algorithms and LSI are compared. We did not consider the Orthogonal Centroid Method, since it projects the data onto 3-dimensional space. The result is shown in Figures 8–10. Note that the three classes in Figure 9 and Figure 10 using our exact and approximation algorithms are better separated than those in Figure 8, where the data were projected using LSI with the reduced dimension equal to 2.

6.4 Analysis

The results from the previous section show several interesting points:

1. In general, the dimension reduction algorithms, like OCM and our exact and approximation algorithms do improve the performance for classification, even for very high dimensional data sets, like those derived from text documents. The dimensional re-

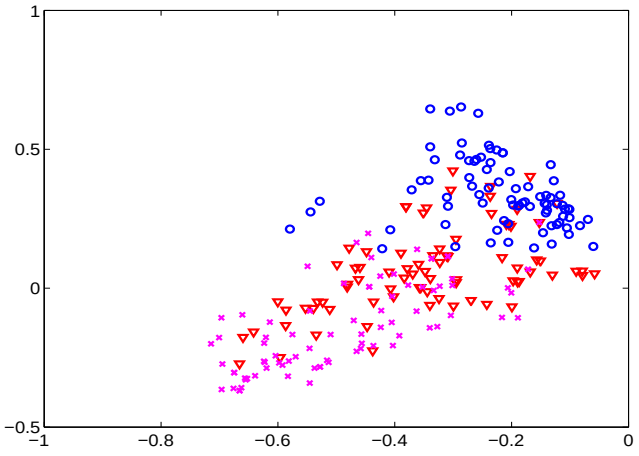

Figure 8: Projection onto 2-dimensions by Latent Semantic Indexing

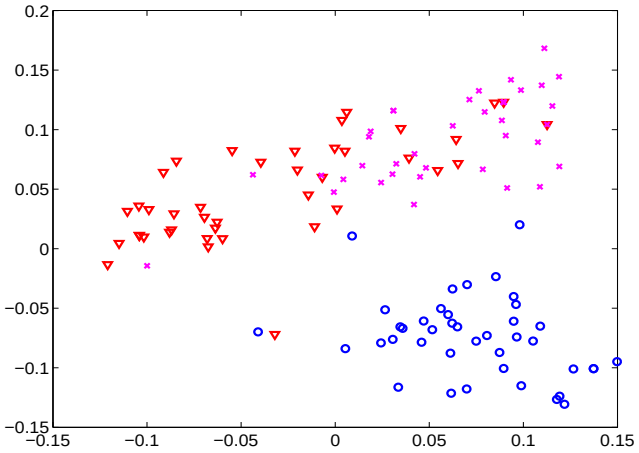

Figure 9: Projection onto 2-dimensions by our exact algorithm

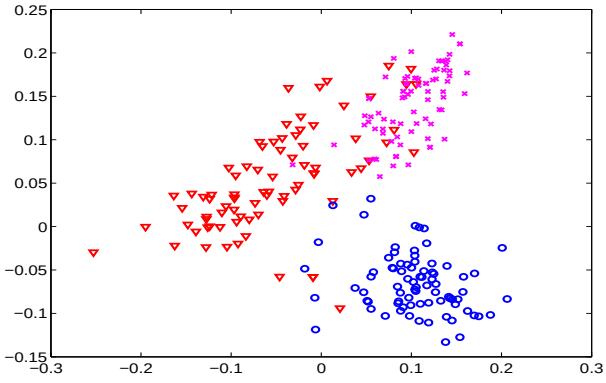

Figure 10: Projection onto 2-dimensions by our approximation algorithm

duction step may be time-consuming, but it dramatically reduces the query time.

2. Algorithms using the label information like our proposed exact and approximation algorithms, and the Orthogonal Centroid Method have better performance than those without using the label information, like LSI. The results also show better performance of our proposed exact and approximation algorithms over the Orthogonal Centroid Method.

3. Our approximation algorithm deals with a much smaller size problem, compared with the one in the exact algorithm. However, the results from all the experiments show they have similar misclassification rates, while the approximation algorithm has much lower running time complexity.

4. In all of the examples in this paper, we found out the optimal reduced dimension of our exact and approximation algorithms $\delta^\star$ equal to $k-1$, where k is the number of clusters in the document set. However, it is possible that $\delta^\star < k-1$, if the k centroids lie in a subspace with dimension less than $k-1$.

7 Conclusions

A new criterion for generalized linear discriminant analysis is presented. The new criterion is applicable to the undersampled problems, thus overcoming the limitation of the classical linear discriminant analysis. A new formulation for the proposed generalized linear discriminant analysis based on the trace optimization is discussed. Generalized singular value decomposition is applied to solve the optimization problem. The solution from the LDA/GSVD algorithm is a special case of the solution for this new optimization problem.

The exact algorithms involve the matrix $H_w \in R^{m\times n}$ with high column dimension n. To reduce the decomposition time for the exact algorithm, an approximation algorithm is presented, which applies K-Means algorithm to each cluster and

replace the cluster by the centroids of the resulting sub-clusters. The column dimension of the matrix H_w is reduced dramatically, therefore reducing the complexity for the computation of GSVD. Experimental results on various real data set show that the approximation algorithm produces results close to those produced by the exact algorithm.

References

[1] P. Baldi and G.W. Hatfield. *DNA Microarrays and Gene Expression: From experiments to data analysis and modeling*, Cambridge, 2002.

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

[3] S. Deerwester, S.T. Dumais, G.W. Furnas, T.K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *J. of the Society for Information Science*, 41, pp. 391–407, 1990.

[4] I.S. Dhillon and D.S. Modha. Concept Decompositions for Large Sparse Text Data using Clustering. *Machine Learning*. 42, pp. 143–175, 2001.

[5] I. S. Dhillon, D. M. Modha, and W. S. Spangler. Class Visualization of High-Dimensional Data with Applications. *Computational Statics & Data Analysis* (Special issue on Matrix Computations & Statistics), vol. 4:1, pp. 59–90, 2002.

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

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

[8] G.H. Golub, and C.F. Van Loan. Matrix Computations, John Hopkins University Press, 3rd edition, 1996.

[9] P. Howland, M. Jeon, and H. Park. Cluster structure preserving dimension reduction based on the generalized singular value decomposition. *SIAM Journal on Matrix Analysis and Applications*, to appear.

[10] P. Howland and H. Park. Generalizing discriminant analysis using the generalized singular value decomposition, submitted to *IEEE Transactions on Pattern Analysis and Machine Learning*, January, 2003.

[11] P. Howland, J. Wang, and H. Park. Generalized discriminant analysis and its applicatino to face recognition, in preparation.

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

[13] Y. Jung, H. Park, and D. Du. An Effective Term-Weighting Scheme for Information Retrieval, Technical Report TR00-008. Department of Computer Science and Engineering, University of Minnesota, Twin Cities, U.S.A., 2000.

[14] H. Kim, P. Howland, and H. Park. Dimension reduction in text classification using support vector machines, Technical Report TR03-014. Department of Computer Science and Engineering, University of Minnesota, Twin Cities, U.S.A., 2003.

[15] G. Kowalski. Information Retrieval System: Theory and Implementation, *Kluwer Academic Publishers*, 1997.

[16] D.D. Lewis. Reuters-21578 text categorization test collection distribution 1.0. *http://www.research.att.com/~lewis*, 1999.

[17] C.C. Paige, and M.A.Saunders. Towards a generalized singular value decomposition, *SIAM Journal on Numerical Analysis*. 18, pp. 398–405, 1981.

[18] H. Park, M. Jeon and J.B. Rosen. Lower dimensional representation of text data based on centroids and least squares. *BIT*, to appear.

[19] M.F Porter. An algorithm for suffic stripping. Program, 14(3), pp. 130–137, 1980.

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

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

[22] TREC. Text Retrieval conference. *http://trec.nist.gov*, 1999.

[23] C. F. Van Loan. Generalizing the singular value decomposition. *SIAM Journal on Numerical Analysis*, 13(1), pp. 76–83, 1976.

[24] Y. Zhao, and G. Karypis. Criterion Functions for Document: Experiments and Analysis. Technical Report TR01-040. Department of Computer Science and Engineering, University of Minnesota, Twin Cities, U.S.A., 2001.