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 03-015

Matrix Rank Reduction for Data Analysis and Feature Extraction

Haesun Park and Lars Elden

February 28, 2003

Matrix Rank Reduction for Data Analysis and Feature Extraction

Haesun Park * and Lars Eldén †

February 28, 2003

Abstract

Numerical techniques for data analysis and feature extraction are discussed using the framework of matrix rank reduction. The singular value decomposition (SVD) and its properties are reviewed, and the relation to Latent Semantic Indexing (LSI) and Principal Component Analysis (PCA) is described. Methods that approximate the SVD are reviewed. A few basic methods for linear regression, in particular the Partial Least Squares (PLS) method, are presented, and analyzed as rank reduction methods. Methods for feature extraction, based on centroids and the classical Linear Discriminant Analysis (LDA), as well as an improved LDA based on the generalized singular value decomposition (LDA/GSVD) are described. The effectiveness of these methods are illustrated using examples from information retrieval, and 2 dimensional representation of clustered data.

1 Introduction

The concepts of *rank reduction*, *matrix factorization*, *data compression*, *dimension reduction*, and *feature selection/extraction* are very closely related. The key link between the concepts is the *Wedderburn rank reduction theorem* from 1934 (comprehensive bibliographies are given in [6, 25]).

THEOREM 1.1 (WEDDERBURN [48]) *Suppose $A \in \mathbb{R}^{m \times n}$, $f \in \mathbb{R}^{n \times 1}$, and $g \in \mathbb{R}^{m \times 1}$. Then*

$$\text{rank}(A - \omega^{-1} A f g^T A) = \text{rank}(A) - 1$$

if and only if $\omega = g^T A f \neq 0$. ■

Let $A^{(1)} = A$, and define a sequence of matrices $\{A^{(i)}\}$ from

$$A^{(i+1)} = A^{(i)} - \omega_i^{-1} A^{(i)} f^{(i)} g^{(i)T} A^{(i)} \quad (1)$$

for any vectors $f^{(i)} \in \mathbb{R}^{n \times 1}$ and $g^{(i)} \in \mathbb{R}^{m \times 1}$ such that

$$\omega_i = g^{(i)T} A^{(i)} f^{(i)} \neq 0. \quad (2)$$

*Department of Computer Science and Engineering, University of Minnesota, Minneapolis, MN 55455, U.S.A. (e-mail: hpark@cs.umn.edu). The work of this author was supported in part by the National Science Foundation grant 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).

†Department of Mathematics, Linköping University, S-581 83 Linköping, Sweden, (e-mail: laeld@math.liu.se). This paper will appear as a chapter in Handbook of parallel computing and statistics, Marcel Dekker, Ed. E.J. Kontogiorgos.

Then the sequence defined in (1) terminates in $\gamma = \text{rank}(A)$ steps since each time the rank of the matrix decreases by exactly one. This process is called a *rank-reducing process* and the matrices $A^{(i)}$ are called Wedderburn matrices. For details, see [6]. This finite terminating process gives a matrix *rank reducing decomposition* which can be defined as

$$A = \hat{F}\Omega^{-1}\hat{G}^T \quad (3)$$

where

$$\hat{F} = [\hat{f}_1, \dots, \hat{f}_\gamma] \in \mathbb{R}^{m \times \gamma} \quad \text{with} \quad \hat{f}_i = A^{(i)} f^{(i)} \quad (4)$$

$$\Omega = \text{diag}(\omega_1, \dots, \omega_\gamma) \in \mathbb{R}^{\gamma \times \gamma}, \quad \text{and} \quad (5)$$

$$\hat{G} = [\hat{g}_1, \dots, \hat{g}_\gamma] \in \mathbb{R}^{n \times \gamma} \quad \text{with} \quad \hat{g}_i = A^{(i)T} g^{(i)}. \quad (6)$$

There are many choices of the vectors $f^{(i)}$ and $g^{(i)}$ that satisfy the condition (2). Therefore, various rank reducing decompositions (3) are possible. Theorem 1.1 has been used to explain many matrix decompositions and it will be the basis for the rank reduction procedures discussed in this chapter.

Theorem 1.1 can be generalized to the case where the reduction of rank is larger than one, as shown in the next theorem.

THEOREM 1.2 (GUTTMAN [19]) *Suppose $A \in \mathbb{R}^{m \times n}$, $F \in \mathbb{R}^{n \times k}$, and $G \in \mathbb{R}^{m \times k}$. Then*

$$\text{rank}(A - AFR^{-1}G^T A) = \text{rank}(A) - \text{rank}(AFR^{-1}G^T A) \quad (7)$$

if and only if $R = G^T AF \in \mathbb{R}^{k \times k}$ is nonsingular. ■

In [6] the authors discuss rank reduction from the point of view of solving linear systems of equations. It is shown that several standard matrix factorizations in numerical linear algebra are instances of the Wedderburn formula. A complementary view is taken in [25], where the Wedderburn formula and matrix factorizations are considered as tools for data analysis: “The major purpose of a matrix factorization in this context is to obtain some form of lower rank approximation to A (...) for understanding the structure of the data matrix...”.

In this chapter rank reduction will be discussed mainly from the perspective of data analysis, but it will be seen that data analysis concepts and methods are often the same as those appearing in connection with the solution of linear systems of equations, although different terminology and notation are used. By formulating a specific criterion for the data analysis, a rank reduction is obtained as a particular instance of the matrix rank reduction formula presented in Theorems 1.2 and 1.1. If it is required that the data matrix A be well approximated by a low rank matrix, then one is lead to rank reduction using the singular value decomposition (SVD) or, equivalently, principal component analysis. This is referred to here as rank reduction without constraints. In Section 2, some fast methods for approximating the SVD are described. The application of singular value analysis to information retrieval, LSI, is discussed.

In multiple regression, the rank reduction is coupled to the reduction in norm of a residual vector, which is the (stochastic and modeling) error in a linear model. A common method here is principal components regression, described in Section 3.1. An alternative, and often very flexible method is partial least squares, PLS, discussed in Section 3.2. In some cases the rank reduction aims at facilitating feature extraction; this is discussed in Section 3.3. These procedures can be considered as constrained rank reduction methods.

Throughout this chapter the data matrix is denoted as A with dimension $m \times n$ and it is assumed that A is real. The Euclidean vector norm and its subordinate matrix norm are written as $\|\cdot\|_2$, and $\|A\|_F$ is the Frobenius norm of the matrix A [16, Chapter 2.2-2.3].

2 Rank Reduction without Constraints

In this section, the singular value decomposition (SVD) is presented, which gives an *optimal* lower rank approximation to a given matrix, where optimality is measured in the 2-norm or the Frobenius norm. Two applications of the SVD, Latent Semantics Indexing (LSI) from information retrieval and Principal Component Analysis (PCA) from statistical pattern recognition, are discussed. Then several methods including orthogonal decomposition with pivoting and the complete orthogonal decomposition are discussed that are often used in practice to approximate the SVD since the SVD is expensive to compute and update.

2.1 Singular Value Decomposition

The following theorem defines the SVD. For a proof of the theorem, see [5, 16], for example.

THEOREM 2.1 *For any matrix $A \in \mathbb{R}^{m \times n}$, there exist matrices*

$$U = [u_1, \dots, u_m] \in \mathbb{R}^{m \times m}, \quad \text{and} \quad V = [v_1, \dots, v_n] \in \mathbb{R}^{n \times n},$$

with $U^T U = I$ and $V^T V = I$, such that

$$U^T A V = \Sigma = \text{diag}(\sigma_1, \dots, \sigma_p) \in \mathbb{R}^{m \times n}, \quad p = \min(m, n), \quad (8)$$

where $\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_p \geq 0$. ■

The σ_i are the singular values, and the columns of U and V are left and right singular vectors, respectively. Suppose for an index $\gamma \leq \min(m, n)$, $\sigma_\gamma > 0$ and $\sigma_{\gamma+1} = 0$. Then

$$A = U \Sigma V^T = \begin{pmatrix} U_\gamma & \hat{U}_\gamma \end{pmatrix} \begin{pmatrix} \Sigma_\gamma & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_\gamma^T \\ \hat{V}_\gamma^T \end{pmatrix} = U_\gamma \Sigma_\gamma V_\gamma^T, \quad (9)$$

where $U_\gamma \in \mathbb{R}^{m \times \gamma}$, $\Sigma_\gamma = \text{diag}(\sigma_1, \dots, \sigma_\gamma) \in \mathbb{R}^{\gamma \times \gamma}$, and $V_\gamma \in \mathbb{R}^{n \times \gamma}$.

The SVD is one of the most powerful decompositions since it provides all four *fundamental* spaces related to the matrix A as

- U_γ gives an orthogonal basis for $\text{Range}(A)$,
- $\hat{V}_\gamma$ gives an orthogonal basis for $\text{Null}(A)$,
- V_γ gives an orthogonal basis for $\text{Range}(A^T)$,
- $\hat{U}_\gamma$ gives an orthogonal basis for $\text{Null}(A^T)$,

where Range and Null denote the range space and the null space of the matrix, respectively. In addition, the SVD provides the rank of the matrix, which is the number of positive singular values, γ .

However, in floating point arithmetic, these zero singular values may appear as small numbers. In general, a large relative gap between two consecutive singular values is considered to reflect the numerical rank deficiency of a matrix. It is widely known that noise filtering can be achieved via a *truncated SVD*. If trailing *small* diagonal components of Σ are replaced with zeros, then a rank k approximation A_k for A can be obtained as

$$\begin{aligned} A &= \begin{pmatrix} U_k & \hat{U}_k \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & \hat{\Sigma}_k \end{pmatrix} \begin{pmatrix} V_k^T \\ \hat{V}_k^T \end{pmatrix} \\ &\approx \begin{pmatrix} U_k & \hat{U}_k \end{pmatrix} \begin{pmatrix} \Sigma_k & 0 \\ 0 & 0 \end{pmatrix} \begin{pmatrix} V_k^T \\ \hat{V}_k^T \end{pmatrix} = U_k \Sigma_k V_k^T = A_k, \end{aligned} \quad (10)$$

where $\Sigma_k \in \mathbb{R}^{k \times k}$ and $\|\hat{\Sigma}_k\|_2 < \epsilon$ for a *small* tolerance ϵ . This is summarized in the following theorem, which provides an important application of the SVD, the lower rank approximation. The following theorem is the theoretical foundation of numerous important procedures in science and engineering.

THEOREM 2.2 *Let the SVD of $A \in \mathbb{R}^{m \times n}$ be given as in Theorem 2.1, and assume that an integer k is given with $0 < k \leq \gamma = \text{rank}(A)$. Then*

$$\min_{B \in \mathbb{R}^{m \times n}, \text{rank}(B)=k} \|A - B\|_2 = \|A - A_k\|_2 = \sigma_{k+1}$$

where

$$A_k = U_k \Sigma_k V_k^T = \sum_{i=1}^k \sigma_i u_i v_i^T, \quad (11)$$

U_k consists of the first k columns of U , V_k the first k columns of V , and Σ_k is the $k \times k$ leading principal submatrix of Σ . ■

A similar result is obtained when the Frobenius norm is used.

THEOREM 2.3 (ECKART, YOUNG [9]) *Let the SVD of $A \in \mathbb{R}^{m \times n}$ be given as in Theorem 2.1, and assume that an integer k is given with $0 < k \leq \gamma = \text{rank}(A)$. Then*

$$\min_{B \in \mathbb{R}^{m \times n}, \text{rank}(B)=k} \|A - B\|_F = \|A - A_k\|_F = \sum_{i=k+1}^p \sigma_i$$

where A_k is the same as in Theorem 2.2, and $p = \min(m, n)$. ■

Theorems 2.2 and 2.3 show that the singular values indicate how close a given matrix is to a matrix of lower rank.

The relation between the truncated SVD (11) and the Wedderburn matrix rank reduction process (7) can be demonstrated as follows. In the rank reduction formula, define the error matrix E as

$$E = A - AF(G^T AF)^{-1}G^T A, \quad F \in \mathbb{R}^{n \times k}, G \in \mathbb{R}^{m \times k}.$$

Assume that $k \leq \text{rank}(A) = \gamma$, and consider the problem

$$\min \|E\|_2 = \min_{F \in \mathbb{R}^{n \times k}, G \in \mathbb{R}^{m \times k}} \|A - AF(G^T AF)^{-1}G^T A\|_2.$$

According to Theorem 2.2, the minimum error is obtained when $F = V_k$ and $G = U_k$, which gives

$$\begin{aligned} (AF)(G^T AF)^{-1}(G^T A) &= (AV_k)(U_k^T AV_k)^{-1}(U_k^T A) \\ &= (U_k \Sigma_k)(\Sigma_k)^{-1}(\Sigma_k V_k^T) = U_k \Sigma_k V_k^T. \end{aligned}$$

This same result can be obtained by a stepwise procedure, when k pairs of vectors $f^{(i)}$ and $g^{(i)}$ are to be found, where each pair reduces the matrix rank by 1 in each of k rank reduction problems. Consider the problem

$$\min \|E_1^{(1)}\|_2 = \min_{f^{(1)} \in \mathbb{R}^{n \times 1}, g^{(1)} \in \mathbb{R}^{m \times 1}} \|A - (g^{(1)T} A f^{(1)})^{-1} A f^{(1)} g^{(1)T} A\|_2.$$

Then according to Theorem 2.2, the solution is obtained when $f^{(1)} = v_1$ and $g^{(1)} = u_1$. Now for

$$A^{(2)} = A^{(1)} - A^{(1)}v_1u_1^T A^{(1)}/\sigma_1 = A^{(1)} - \sigma_1 u_1 v_1^T,$$

with $A^{(1)} = A$, the solution for

$$\min \|E_1^{(2)}\|_2 = \min_{f^{(2)} \in \mathbb{R}^{n \times 1}, g^{(2)} \in \mathbb{R}^{m \times 1}} \|A^{(2)} - (g^{(2)T} A^{(2)} f^{(2)})^{-1} A^{(2)} f^{(2)} g^{(2)T} A^{(2)}\|_2$$

is obtained with $f^{(2)} = v_2$ and $g^{(2)} = u_2$, since

$$A^{(1)} - \sigma_1 u_1 v_1^T = \sum_{i=2}^r \sigma_i u_i v_i^T.$$

On the other hand, the rank 2 reduction problem,

$$\min \|E_2\|_2 = \min_{F^{(1)} \in \mathbb{R}^{n \times 2}, G^{(1)} \in \mathbb{R}^{m \times 2}} \|A - AF^{(1)}(G^{(1)T} AF^{(1)})^{-1} G^{(1)T} A\|_2,$$

has the solution $F^{(1)} = (v_1 \ v_2)$ and $G^{(1)} = (u_1 \ u_2)$, again due to Theorem 2.2. Therefore, reducing the rank k times by finding the rank one reducing vectors that give minimum two norm distance between the matrix and the reduced rank matrix is equivalent to reducing the rank by k at once by finding the rank k reducing matrices that satisfy the same criterion.

Similar results can be obtained from Theorem 2.3 when the matrix Frobenius norm is used.

2.2 Latent Semantic Indexing and the SVD

In the vector space based information retrieval system [38, 39], the document collection is represented as vectors in a matrix. Specifically, a term-document matrix

$$A = \begin{bmatrix} a_1 & a_2 & \cdots & a_n \end{bmatrix} \in \mathbb{R}^{m \times n} \quad (12)$$

is formed based on the collection of documents, where m is the total number of terms in the document collection and n is the number of documents. Each column of A represents a document, and in the matrix A the elements a_{ij} are weighted frequencies of each word in a specific document representing the importance of term i in document j . The simplest a_{ij} is binary, but to improve retrieval performance, various weighting methods have been developed [27, 39]. For other related topics such as stemming and removing stop lists, see [11, 39, 4]. The SMART system [39] is one of the most influential test-beds where the vector space based method is successfully implemented. One major advantage of a vector space based method is that the algebraic structures of the term-document matrix can be exploited using the techniques developed in linear algebra.

For achieving higher efficiency and effectiveness in manipulating these data, it will be necessary to find a lower dimensional representation of the data [30]. A vector space based information retrieval system needs to solve the following three problems frequently: document retrieval, classification, and clustering. Document retrieval is used to extract relevant documents from a text database given a query. Classification is the process of assigning new data to its proper group. The group is also called class or category [1]. Clustering is the process to find homogeneous groups (clusters) of data based on the values of their vector components and a predefined measure [31]. While the category structure is known in classification, in cluster

analysis little or nothing is known about the category structure. All that is available is a collection of data whose category memberships are unknown. The objective is to discover a category structure in the data set.

In document retrieval, classification, and in some clustering algorithms, the major computation involves comparison of two vectors, which will be affected by different weighting schemes and the similarity measures [27, 39]. With dimension reduction of the given text collection, the complexity of subsequent computations involved in these problems can be substantially reduced. To achieve higher efficiency in computation, often it is necessary to reduce the dimension severely, and in the process, one may lose much information which was available in the original data. Therefore, it is important to achieve a *better representation* of data in the lower dimensional space according to specific tasks to be performed after the dimension reduction, such as classification, rather than simply reducing the dimension of the data to best approximate the full term-document matrix. This is discussed later in detail in Section 3.3. The significance of this has been recognized by Hubert, et.al. [25], for example. The difficulty involved is that it is not easy to measure how well a certain dimension reduction method provides a good representation of the original data. It seems that this can only be estimated using experimental results. The dimension reduction/feature extraction methods that will be discussed in this chapter are based on the vector subspace computation in linear algebra.

In Latent semantic indexing (LSI) [3, 8, 17, 36], the SVD of the term-document matrix is utilized for conceptual retrieval and lower dimensional representation. The dimension reduction by the optimal lower rank approximation from the SVD has been successfully applied in numerous other applications, e.g. in signal processing. In these applications, often what the dimension reduction achieves is the effect of removing noise in the data. In the case of information retrieval, often the term-document matrix has either full rank or close-to full rank. Also the meaning of *noise* in the text data collection is not well understood, unlike in other applications such as signal processing [37] or image processing.

LSI is based on the assumption that there is some underlying latent semantic structure in the data of the term-document matrix that is corrupted by the wide variety of words used in documents and queries for the same objects (the problem of polysemy and synonymy, see [8]). The basic idea of the LSI/SVD is that if two document vectors represent the same topic, they will share many associating words with a keyword, and they will have very close semantic structures after dimension reduction via the SVD.

In classification, clustering, and document retrieval, the fundamental operation is to compare a document to another document. The choice of similarity measure plays an important role [27]. In vector space based information retrieval, the most commonly used similarity measures are, L_2 norm (Euclidean distance), inner product, cosine, or variations of these [27]. When the inner product is used as a measure, using the rank k approximation A_k for A , the documents are compared as,

$$A^T A \approx A_k^T A_k = V_k \Sigma_k^T U_k^T U_k \Sigma_k V_k^T = (V_k \Sigma_k^T)(\Sigma_k V_k^T), \quad (13)$$

i.e., the inner product between a pair of columns of A can be approximated by the inner product between a pair of columns of $\Sigma_k V_k^T$. Accordingly, $V_k \Sigma_k^T \in \mathbb{R}^{k \times n}$ is considered a representation of the document vectors in the reduced k dimensional space.

Let $q \in \mathbb{R}^{m \times 1}$ represent a query vector, i.e. a document that has been preprocessed in the same way as the documents represented by the columns of A . Consider the inner product between a q and the document vectors in A :

$$q^T A \approx q^T A_k = (q^T)(U_k \Sigma_k V_k^T) = (q^T U_k)(\Sigma_k V_k^T). \quad (14)$$

Eqn. (14) shows that a new vector $q \in \mathbb{R}^{m \times 1}$ can be represented as

$$\hat{q} = U_k^T q \quad (15)$$

in the k dimensional space, since the columns of $\Sigma_k V_k^T$ represent the columns of A in the k dimensional space. In LSI/SVD, it has also been proposed that q be reduced to a vector in $\mathbb{R}^{k \times 1}$ as

$$\hat{q} = \Sigma_k^{-1} U_k q. \quad (16)$$

2.3 Principal Component Analysis and SVD

A common task in data analysis is to project high-dimensional numerical data on a low-dimensional subspace. The purpose of the projection is different depending on the application: In some cases the low-dimensional representation of the data is used for visualization of the data set. In other cases the computations using the high-dimensional representation would be too costly (e.g. in a real time application), so that a dimension reduction is needed for numerical efficiency. In data analysis applications this dimensionality reduction is routinely done using principal component analysis (PCA). See e.g. [32] and [20, Chapter 14.5].

Often in PCA it is assumed that the data are preprocessed so that each column of A has mean zero and Euclidean length one. The reader may assume that this has been done in the sequel (however, it is not essential for the description). PCA is usually presented and its solution is computed in terms of the eigenvalues and eigenvectors of AA^T and $A^T A$. It turns out that the derivation and computation of solution can also be done, with certain advantages, in terms of the SVD.

From Theorem 2.3 one has

$$\min_{\text{rank}(B)=k} \|A - B\|_F^2 = \|A - A_k\|_F^2 = \|A - U_k \Sigma_k V_k^T\|_F^2 = \|A - \sum_{i=1}^k \sigma_i u_i v_i^T\|_F^2.$$

Defining

$$L_k = \Sigma_k V_k^T = \begin{pmatrix} \sigma_1 v_1^T \\ \sigma_2 v_2^T \\ \vdots \\ \sigma_k v_k^T \end{pmatrix} = \begin{pmatrix} l_1^{(k)} & l_2^{(k)} & \dots & l_n^{(k)} \end{pmatrix},$$

where $l_i^{(k)}$ denotes the i th column vector of L_k , one can write

$$\|A - A_k\|_F^2 = \|A - U_k L_k\|_F^2 = \sum_{i=1}^n \|a_i - U_k l_i^{(k)}\|_2^2,$$

where a_i is a column vector of A . This means that there are n linear models for the columns of A ,

$$a_i \sim U_k l_i^{(k)}, \quad i = 1, 2, \dots, n, \quad (17)$$

with the same orthogonal basis vectors $u_1, u_2, \dots, u_k$.

The SVD also gives linear models for the row vectors. Assume that the i th row of A is denoted as $\hat{a}_i^T$, i.e.,

$$A = \begin{pmatrix} \hat{a}_1^T \\ \hat{a}_2^T \\ \vdots \\ \hat{a}_m^T \end{pmatrix}.$$

Defining

$$\hat{L}_k = \Sigma_k U_k^T = \begin{pmatrix} \hat{l}_1^{(k)} & \hat{l}_2^{(k)} & \dots & \hat{l}_m^{(k)} \end{pmatrix},$$

one obtains

$$\hat{a}_i \sim V_k \hat{l}_i^{(k)}, \quad i = 1, 2, \dots, m. \quad (18)$$

The linear models (17) and (18) are often referred to as *principal component analyses (PCA)* of the data matrix; the vectors u_i are called the principal components of the column space, and the v_i the principal components of the row space. It is common to choose $k = 2$ or 3 , and plot the coordinates $l_i^{(k)}$ ($\hat{l}_i^{(k)}$) of the columns (rows) in a Cartesian coordinate system, in order to visualize or cluster the data. According to Theorems 2.2 and 2.3 the principal components are ordered so that the first one reflects the direction of largest variation in the data matrix, the second one the second largest, and so on. In the next example, the principal components are shown to have a concrete interpretation, and the directions of largest variation are illustrated.

EXAMPLE 2.4 *In an application of handwritten digit classification [40], the singular value decomposition of a collection of a couple of hundred handwritten “4” was computed. Each image of the digit was discretized on a 16×16 grid (256 pixels), and then made into a vector of length 256. All the vectors were collected in a data matrix, and its SVD was computed. Then the first singular vectors (each of length 256) were reshaped as 16×16 matrices, which can then be printed as images. In Figure 1 the first three left singular vectors are illustrated. The first left singular vector, the dominating direction of variation of the*

Figure 1: The first three singular vectors of a collection of handwritten “4”.

data matrix, represents a rather well written digit “4”. The following two illustrate the two most important directions of variation of the set of digits, in the sense that many of the digits in the collection can be represented quite well as linear combinations of these first three basis vectors (principal components). ■

Sometimes in the literature it is recommended to compute the principal components u_i and v_i as the eigenvectors of AA^T and $A^T A$, respectively. However, from the point of view of accuracy it is better to compute the SVD, since the forming (explicitly or implicitly) of AA^T ($A^T A$) causes loss of information. For small and medium sized matrices¹, there are efficient and accurate algorithms for computing the SVD, implemented in numerical software libraries such as LAPACK [2] and problem solving environments such as MATLAB, Mathematica, SAS, SPSS, etc.

¹Today (2003) a matrix with $\max(m, n) \leq C10^3$, say, might be considered as medium sized. The value of C depends on the available computer power.

2.4 Pivoted Orthogonal Decomposition

The power of the SVD lies in the fact that one can determine the rank of the matrix by simply reading off the diagonal elements of the matrix Σ . Due to this fact, orthogonal basis for the four fundamental spaces can also be obtained simply from the columns of the matrices U and V . However, a SVD is rather expensive to compute. Therefore, much research has been devoted to approximating the SVD by computationally less expensive methods. One group of approaches for achieving this is to relax the condition for diagonalization, and compute a triangularization, from which the approximate rank of the matrix is revealed.

THEOREM 2.5 (ORTHOGONAL (QR) DECOMPOSITION) *For any matrix $A \in \mathbb{R}^{m \times n}$, there exist a matrix $Q \in \mathbb{R}^{m \times m}$ with $Q^T Q = I$, such that*

$$Q^T A = R \in \mathbb{R}^{m \times n} \quad (19)$$

where R is upper triangular. ■

When $m \geq n$ and $\text{rank}(A) = n$, the first n columns of the matrix Q is an orthonormal basis for $\text{Range}(A)$. However, this is not true in general. The following example [16] illustrates that the QR decomposition does not provide bases for all four fundamental spaces.

EXAMPLE 2.6 *Consider the QR decomposition of the matrix*

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}.$$

Clearly $\text{rank}(A) = 2$ but no two columns of the matrix Q form a basis for $\text{Range}(A)$. ■

In addition, the following example [16] shows that one cannot obtain a reliable numerical rank of a triangular matrix based on its diagonal elements. Therefore, the QR decomposition does not provide the rank of the matrix in general.

EXAMPLE 2.7 *Consider the matrix R_n defined as*

$$R_n = \begin{pmatrix} 1 & -1 & -1 & \cdots & -1 \\ 0 & 1 & -1 & \cdots & -1 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & & & 1 & -1 \\ 0 & \cdots & 0 & 1 \end{pmatrix} \in \mathbb{R}^{n \times n}. \quad (20)$$

Although there is no small diagonal component, the matrix R_n gets closer to be rank deficient as n becomes larger. For example, $\sigma_{50}(R_{50}) \approx 2.7 \times 10^{-15}$ and $\sigma_{100}(R_{100}) \approx 3.1 \times 10^{-18}$. ■

The QR decomposition provides an orthogonal basis for the matrix only when the matrix has full rank, in general. Therefore, the SVD is needed when the matrix is rank deficient or the rank of the matrix is unknown, and any information on the four fundamental subspace is required. One of the simpler methods for approximating the SVD is the QR decomposition with column pivoting, introduced by Golub [14].

Assume that $A \in \mathbb{R}^{m \times n}$ is rank deficient with rank γ . The *exact* QR decomposition with column pivoting of A is then given by

$$Q^T A \Pi = R = \begin{pmatrix} \tilde{R}_\gamma & \tilde{S}_\gamma \\ 0 & 0 \end{pmatrix},$$

where $\tilde{R}_\gamma \in \mathbb{R}^{\gamma \times \gamma}$ is upper triangular with $\text{rank}(\tilde{R}_\gamma) = \gamma$, $Q \in \mathbb{R}^{m \times m}$ is orthogonal, and $\Pi \in \mathbb{R}^{n \times n}$ is a permutation matrix. In floating point arithmetic, one may obtain

$$Q^T A \Pi = R = \begin{pmatrix} R_k & S_k \\ 0 & X_k \end{pmatrix}, \quad (21)$$

where $R_k \in \mathbb{R}^{k \times k}$, $\|X_k\|$ is small and k is an estimated value for γ .

Assume that the algorithm for computing (21) has advanced i steps. The reduced matrix then has the form

$$\begin{pmatrix} R_i & S_i \\ 0 & X_i \end{pmatrix},$$

where $R_i \in \mathbb{R}^{i \times i}$ is upper triangular with $\text{rank}(R_i) = i$. The next permutation matrix is chosen so that when it is applied to X_i it moves the column of largest norm to the front of that matrix. Then that column is reduced by an orthogonal transformation so that the dimension of the upper triangular matrix is increased by one.

Suppose the QR decomposition with column pivoting is terminated after k steps satisfying the stopping criterion that $\|X_k\| \leq \epsilon$, for a given tolerance ϵ . Then since

$$Q^{(k)T} Q^{(k-1)T} \cdots A \cdots \Pi_{k-1} \Pi_k = \begin{pmatrix} R_k & S_k \\ 0 & X_k \end{pmatrix},$$

with $Q \equiv Q^{(1)} Q^{(2)} \cdots Q^{(k)} = (Q_1 \ Q_2)$ where $Q_1 \in \mathbb{R}^{m \times k}$, Q_1 provides an approximate orthogonal basis for $\text{Range}(A)$ and Q_2 for $\text{Null}(A^T)$. Some further computation is needed to obtain the bases for the other two fundamental subspaces.

Although the diagonal values of the resulting matrix R_k reveal gaps in the singular values in most practical problems, the following well known example shows that the orthogonal decomposition with column pivoting may fail to detect rank deficiency since a matrix may be rank deficient with no small submatrix X_k .

EXAMPLE 2.8 (KAHAN [28]) Let $T_n(c) \in \mathbb{R}^{n \times n}$ be defined as

$$T_n(c) = \text{diag}(1, s, \dots, s^{n-1}) \begin{pmatrix} 1 & -c & -c & \cdots & -c \\ 0 & 1 & -c & \cdots & -c \\ & \ddots & & \ddots & \vdots \\ \vdots & & & 1 & -c \\ 0 & \cdots & & & 1 \end{pmatrix}, \quad (22)$$

where $c > 0$, $s > 0$, and $c^2 + s^2 = 1$. The QR decomposition with column pivoting applied to $T_n(c)$ does not change the matrix, i.e., $T_n(c) = R$, $Q = I$ and $\Pi = I$. Therefore, $\|X_k\| \geq s^{n-1}$, whatever the value of k is. However, the smallest singular value of $T_n(c)$ is often much smaller than $\|X_k\|$ for any k . For example, for $T_{10}(0.8)$, $\|X_k\|_2 \geq s^9 \approx 8.96 \times 10^{-5}$ but $\sigma_{10}(T_{10}(0.8)) \approx 0.01$. And for $T_{100}(0.2)$, $\|X_k\|_2 \geq s^{99} \approx 0.13$, but $\sigma_{100}(T_{100}(0.2)) \approx 3.68 \times 10^{-9}$. ■

The singular value gap detecting property of the decomposition can be improved by a simple modification. In his QLP decomposition, Stewart [43] suggested applying the QR decomposition with column pivoting one more time to the matrix R^T where R is obtained from the QR decomposition with column pivoting. Specifically, suppose the QR decomposition with column pivoting applied to A gives

$$Q_A^T A \Pi_A = R \quad (23)$$

and applied to R^T gives

$$Q_R^T R^T \Pi_R = U. \quad (24)$$

Then combining (23) and (24),

$$A = (Q_A \Pi_R) U^T (Q_R^T \Pi_A^T) = QLP. \quad (25)$$

The following examples show that this simple idea of applying the QR decomposition with column pivoting to the triangular matrix R^T one more time greatly improves the singular value tracking capability of the algorithm.

EXAMPLE 2.9 Consider Kahan's example shown in Example 2.8. The matrix $T_{10}(0.8)$ has the diagonal elements (which are the diagonal elements of the R factor in the QR decomposition with column pivoting)

$$[1.00, 0.600, 0.360, 0.216, 0.130, 0.0778, 0.0467, 0.0280, 0.0168, 0.0101],$$

and $T_{10}(0.8)$ has the singular values

$$[2.889, 1.04, 0.608, 0.359, 0.212, 0.124, 0.0722, 0.0412, 0.0225, 8.96 \times 10^{-5}].$$

Therefore, the diagonal elements of R are not very accurate approximations of the singular values. On the other hand, the diagonal elements of the triangular factor U from the QLP algorithm applied to $T_{10}(0.8)$ are

$$[2.60, 1.10, 0.619, 0.362, 0.213, 0.125, 0.0727, 0.0411, 0.0214, 9.50 \times 10^{-5}].$$

Note that all the singular values are now approximated much better. ■

EXAMPLE 2.10 Consider the matrix

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix},$$

which has the singular values

$$[2.18, 1.13, 0].$$

The diagonal elements of the matrix R from its QR decomposition with column pivoting (computed using MATLAB) are

$$[2.00, 0.866, 0]$$

and the diagonal elements from the matrix U from its QLP decomposition are

$$[2.12, 1.15, 0].$$

■

In addition, with the QLP algorithm, one can have a good approximation for all four fundamental subspaces of the matrix A .

By allowing the orthogonal matrices applied from both sides of the matrix as in the QLP decomposition instead of simply permuting the columns, one can obtain a better approximation to the SVD. The next section covers this in a more general context.

2.5 Complete Orthogonal Decomposition

As seen earlier, the SVD is of great theoretical and practical importance. Since the computation of the SVD is expensive, there is a need for alternative, cheaper methods. Another important fact is that there are no fast methods to update the SVD when the data matrix is changed slightly, which is a serious drawback when the problem is of recursive nature. For instance, the updating of the SVD when a row is added to A requires $\mathcal{O}(n^3)$ flops (floating point operations) for a matrix of order n .

Among the most successful methods for approximating the SVD are the rank-revealing URV and ULV decompositions due to Stewart [41, 42]. They are effective in exhibiting the rank and give bases for the range and null spaces, and can be updated in $\mathcal{O}(n^2)$ flops. They are compromises between the SVD and a QR decomposition with some of the virtues of both. Here, for simplicity, only the URVD will be discussed. First the concept of a complete orthogonal decomposition [16, Chapter 5.4.2] is defined.

Let A have rank γ . Then a *complete orthogonal decomposition* is given by

$$U^T A V = \begin{pmatrix} \tilde{R}_\gamma & 0 \\ 0 & 0 \end{pmatrix},$$

where $\tilde{R}_\gamma \in \mathbb{R}^{\gamma \times \gamma}$ has rank γ , and $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices.

The SVD is a special case of a complete orthogonal decomposition. When the matrix $A \in \mathbb{R}^{m \times n}$, has *numerical rank* k , i.e. σ_k is *large* compared to σ_{k+1} , its URVD is given by

$$U^T A V = \begin{pmatrix} T \\ 0 \end{pmatrix} = \begin{pmatrix} R_k & S_k \\ 0 & \hat{R}_k \\ 0 & 0 \end{pmatrix}, \quad (26)$$

where $U \in \mathbb{R}^{m \times m}$ and $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices, $T \in \mathbb{R}^{n \times n}$, $R_k \in \mathbb{R}^{k \times k}$, and $\hat{R}_k \in \mathbb{R}^{(n-k) \times (n-k)}$ are upper triangular, and

$$\sigma_k(R_k) \approx \sigma_k, \quad \|S_k\|_F^2 + \|\hat{R}_k\|_F^2 \approx \sigma_{k+1}^2 + \cdots + \sigma_n^2.$$

The algorithm for computing the URVD is based on the following observation: Given any upper triangular matrix $R \in \mathbb{R}^{n \times n}$, let w be a unit norm approximate right singular vector of R corresponding to the smallest singular value σ_n (such an approximation can be obtained using a standard condition estimator, at a cost of $\mathcal{O}(n^2)$ flops). With an orthogonal matrix $Q \in \mathbb{R}^{n \times n}$ such that

$$Q^T w = e_n$$

and the QR decomposition of the product RQ , which is

$$RQ = Q^{(1)} R^{(1)},$$

where $Q^{(1)} \in \mathbb{R}^{n \times n}$ is orthogonal and $R^{(1)} \in \mathbb{R}^{n \times n}$ is upper triangular,

$$\sigma_n \approx \|Rw\|_2 = \|Q^{(1)T} R Q Q^T w\|_2 = \|R^{(1)} e_n\|_2.$$

Thus the last column of $R^{(1)}$ is small, and can be *deflated* in the sense that it is considered to belong to the right “small part” of the matrix. The computation of $R^{(1)}$ can be organized so that $\mathcal{O}(n^2)$ flops are required. Then the analogous procedure is applied to the $(n-1) \times (n-1)$ principal submatrix. After $n-k$ such steps the rank revealing URVD is obtained.

When the orthogonal matrix V is partitioned according to the numerical rank as

$$V = (V_1 \ V_2),$$

where $V_1 \in \mathbb{R}^{n \times k}$ and $V_2 \in \mathbb{R}^{n \times (n-k)}$, V_2 is an orthogonal basis for the null space of A . Similarly a basis of the range space can be obtained.

The decompositions shown in this chapter that approximate the SVD, can be used in LSI and PCA to obtain faster methods.

3 Constrained Rank Reduction

In Section 2 the rank reduction was based on a criterion which gives the directions of largest variation in the data matrix A . In the linear least squares problem

$$\min_x \|Ax - b\|_2, \tag{27}$$

where $A \in \mathbb{R}^{m \times n}$, the criterion emphasizes the “covariation” between the data matrix and the right hand side b . This problem occurs in many problems in data analysis, see e.g. [20]. In statistical terminology this is the *multiple linear regression model*: the right hand side b contains observations of a response variable, whose variation is to be explained by a linear combination of the explanatory variables in the columns of the data matrix A .

Often the data matrix is large and the columns of A are almost linearly dependent (the explanatory variables are almost collinear). Then one would like to find a hopefully small subset of the explanatory variables that account for as much of the variation in b as possible. In other words, a linear combination of a subset of the columns is to be found such that the corresponding residual is as small as possible. However, this *subset selection problem* [16, Chapter 12.2] is quite difficult, and often one accepts a projection of the solution onto a subspace of small dimension. Such a projection can be computed using the SVD of A or the *Partial Least Squares (PLS)* method. In both methods a rank reduction of A is performed, but here the purpose is not primarily to obtain a low rank matrix approximation but rather to reduce the norm of the residual as much as possible.

In many cases it is not the least squares solution x that is of interest in data analysis. Instead, x is an auxiliary variable that is used for *prediction*:

$$y_{\text{predicted}} := \hat{a}^T \hat{x},$$

where $\hat{a}^T$ is a new observation of the explanatory variables, and $\hat{x}$ is an estimate of the solution of (27).

Algorithm 1 PCR

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with its SVD, and a right hand side $b \in \mathbb{R}^m$, it computes a solution projected on the subspace corresponding to the k largest singular values.

1. $x_{pcr}^{(0)} = 0$,
2. **for** $i = 1, 2, \dots, k$
 - (a) $x_{pcr}^{(i)} = x_{pcr}^{(i-1)} + \frac{u_i^T b}{\sigma_i} v_i$.

3.1 Regression using SVD: Principal Component Regression

Using the SVD (9) the minimum norm least squares solution of (27) can be written

$$x = V \begin{bmatrix} \Sigma_\gamma^{-1} & 0 \\ 0 & 0 \end{bmatrix} U^T b = \sum_{i=1}^{\gamma} \frac{u_i^T b}{\sigma_i} v_i, \quad (28)$$

where $\gamma = \text{rank}(A)$. In statistical literature, regression using the SVD goes under the name *principal component regression (PCR)*. The procedure is equivalent to truncating the SVD solution (28), i.e. the solution is expressed in terms of the k first singular vectors, $k \leq \gamma$,

$$x_{pcr}^{(k)} = \sum_{i=1}^k \frac{u_i^T b}{\sigma_i} v_i. \quad (29)$$

This can be interpreted as a rank reduction procedure, where in each step the reduction is made along the direction of largest variation in the data matrix A . PCR is described in Algorithm 1.

The solution (29) is also a projection onto a lower dimensional subspace as is seen from the equivalent least squares problem:

$$\min_z \|(AV_k)z - b\|, \quad x_{pcr}^{(k)} = V_k z, \quad V_k = (v_1 \ v_2 \ \dots \ v_k).$$

Thus, the approximate solution of (27) is obtained by projecting the solution onto the subspace spanned by the singular vectors $v_1, v_2, \dots, v_k$. The value of the truncation index k can be chosen, e.g. using cross validation [12, 51]. The purpose is to choose k such that there is as large a reduction of the norm of the residual for as small a value of k as possible.

It is not uncommon that k must be chosen inconveniently large to obtain the required reduction of the norm of the residual [10, 44]. This happens if the right hand side b does not have significant components along the first left singular vectors u_i . To explain this, use the SVD to write the residual of the least squares problem,

$$r_k^2 := \|Ax_{pcr}^{(k)} - b\|^2 = \left\| \begin{bmatrix} \Sigma_\gamma & 0 \\ 0 & 0 \end{bmatrix} z^k - U^T b \right\|^2, \quad z^k = \begin{pmatrix} u_1^T b / \sigma_1 \\ \vdots \\ u_k^T b / \sigma_k \\ 0 \\ \vdots \\ 0 \end{pmatrix}.$$

Algorithm 2 MPCR (Modified PCR)

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with its SVD, and a right hand side $b \in \mathbb{R}^m$, it computes a solution projected on the subspace corresponding to the k largest components of $U^T b$.

1. $x_{kcr}^{(0)} = 0$, $c = U^T b$,
2. **for** $i = 1, 2, \dots, k$
 - (a) Choose j_i such that $|c_{j_i}| = \max_i |c_i|$,
 - (b) $x_{mPCR}^{(i)} = x_{mPCR}^{(i-1)} + \frac{u_{j_i}^T b}{\sigma_{j_i}} v_{j_i}$,
 - (c) $c_{j_i} = 0$.

Thus the residual becomes

$$r_k^2 = \sum_{i=k+1}^m (u_i^T b)^2.$$

The singular values are ordered in such a way that they reflect the directions of largest variation in the data matrix A . However, if in regression the purpose is to reduce the residual significantly using a projection onto a subspace of as small a dimension as possible, then, naturally, the right hand side must influence the choice of directions. This can be done simply by choosing the equations to satisfy exactly (i.e. the non-zero components of z) *in the order of the largest magnitudes of the vector* $U^T b$. The rank reduction process, here denoted MPCR (Modified PCR), is described in Algorithm 2.

The Partial Least Squares method, described in the next section, automatically takes into account *both the right hand side and the data matrix*.

3.2 Regression using Partial Least Squares

The Partial Least Squares (PLS) method due to Wold [49] is commonly used in data analysis and chemometrics [51]. It has been known for quite some time [50, 10], that PLS is closely related to Lanczos (Golub-Kahan [15]) bidiagonalization. The so called NIPALS version of the method, given in Algorithm 3, is clearly a rank reduction process.

Defining

$$\begin{aligned} W_k &= \begin{bmatrix} w_1 & w_2 & \dots & w_k \end{bmatrix}, \\ T_k &= \begin{bmatrix} t_1 & t_2 & \dots & t_k \end{bmatrix}, \\ S_k &= \begin{bmatrix} s_1 & s_2 & \dots & s_k \end{bmatrix}, \end{aligned}$$

where the columns w_i , t_i , and s_i are computed as in Algorithm 3, it is straightforward to show that the columns of W_k and T_k are orthonormal, see e.g. [50, 10]. In the NIPALS context the approximate solution $x_{pls}^{(k)}$ of (27) is usually obtained from

$$x_{pls}^{(k)} = W_k (S_k^T W_k)^{-1} T_k^T b. \quad (30)$$

From the equivalence of PLS and Lanczos bidiagonalization [50, 10], one can easily show that

$$A W_k = T_k B_k,$$

Algorithm 3 PLS

Given a data matrix $A \in \mathbb{R}^{m \times n}$, and a right hand side $b \in \mathbb{R}^m$, it performs k steps of the partial least squares algorithm.

1. $A^{(1)} = A$,
2. **for** $i = 1, 2, \dots, k$
 - (a) $w_i = \frac{1}{\|(A^{(i)})^T b\|_2} (A^{(i)})^T b$,
 - (b) $t_i = \frac{1}{\|A^{(i)} w_i\|_2} A^{(i)} w_i$,
 - (c) $s_i = (A^{(i)})^T t_i$,
 - (d) $A^{(i+1)} = A^{(i)} - t_i s_i^T$.

where B_k is a bidiagonal matrix, the entries of which are normalizing constants in the algorithm [10]. From this identity, it is seen that the PLS solution (after k steps) is the same as the solution of

$$\begin{aligned} \min_{x \in \text{span}(W_k)} \|Ax - b\|_2^2 &= \min_z \|AW_k z - b\|_2^2 = \min_z \|T_k B_k z - b\|_2^2 \\ &= \min_z \|B_k z - T_k^T b\|_2^2 + \|(T_k^\perp)^T b\|_2^2, \end{aligned}$$

where $T = (T_k \ T_k^\perp) \in \mathbb{R}^{m \times m}$ is an orthogonal matrix. Thus PLS is a projection method, where the solution is expressed as a linear combination of the w_i vectors. The expression for the computation of these vectors,

$$w_i = \frac{1}{\|(A^{(i)})^T b\|_2} (A^{(i)})^T b,$$

shows that w_i is the covariance between the data matrix $A^{(i)}$ and the right hand side. This direction is chosen to *obtain a high correlation between t_i and the right hand side* [23]. Thus, one can interpret the PLS algorithm as follows:

In each step, find the new basis vector for the projection of the solution as the covariance vector between the present data matrix and the right hand side. After the projection has been computed, remove the corresponding information from the data matrix by performing a Wedderburn rank reduction.

For understanding the PLS, it is essential to know that the columns of W_p and T_p are orthogonal bases for Krylov subspaces. Let

$$\mathcal{K}_i(C, b) = \text{span}\{b, Cb, \dots, C^{i-1}b\}$$

denote the linear subspace spanned by the the vectors $b, Cb, \dots, C^{i-1}b$. The following proposition is well known and is easily proved using the recursion for $A^{(i)}$.

PROPOSITION 3.1 *The vectors $w_1, w_2, \dots, w_i$ and $t_1, t_2, \dots, t_i$ constitute orthogonal bases of the Krylov subspaces $\mathcal{K}_i(A^T A, A^T b)$ and $\mathcal{K}_i(AA^T, AA^T b)$, respectively.*

From the above description of PLS one may get the impression that there is always a quick reduction of the residual in the first steps of the algorithm. However, the actual reduction depends in a rather complicated way on the distribution of singular values and the magnitude of the elements of the vector $U^T b$ [10]. The Krylov spaces generated are the key to understanding the behavior of PLS. Consider the space

$$\mathcal{K}_i(A^T A, A^T b) = \text{span}\{A^T b, (A^T A)A^T b, \dots, (A^T A)^{i-1} A^T b\}.$$

It is easy to show, using the theory of the power method [16, Section 8.2], that for (almost) any vector s ,

$$\lim_{j \rightarrow \infty} (A^T A)^j s = v_1, \quad (31)$$

i.e., the first right singular vector. In view of (31), the behavior of the Krylov sequence can be summarized as follows:

1. If the mass of b is concentrated along some intermediate singular vectors, and that mass is large enough (related to the magnitude of the singular values), then the first few Krylov basis vectors will contain large components along those singular vectors. If the mass is not large enough, then the first basis vectors will be dominated by directions corresponding to the largest singular values.
2. The next several Krylov basis vectors will be dominated by components along the singular vectors corresponding to the largest singular values.
3. Gradually, since a sequence of orthogonal basis vectors is generated, the influence of singular vectors corresponding to smaller singular values will be noticed. Unless there is a huge component in the right hand side along a singular vector corresponding to a small singular value, it will take long before directions corresponding to the small singular values play a role in the approximate solution.

The following example illustrates that the rate of decay of the residual depends on the right hand side.

EXAMPLE 3.2 *A problem with a diagonal matrix of dimension 50 was constructed, whose first 22 diagonal elements decay from 1 to 0.1, with the rest decaying from 0.1 to 0.001, approximately. Two right hand sides were used as shown in the first two graphs of Figure 2. The coordinates of the right hand sides in terms of the left singular vectors u_i (here: the standard basis vectors e_i) are plotted in the first two graphs of Figure 2. In the third graph of Figure 2, the residuals are given. For Case 1, PLS automatically reads off the distribution of the $u_i^T b$ (in a way similar to MPCR), and reduces the residual significantly in the first two steps. On the other hand, in Case 2, which has a considerably more complicated right hand side, PLS requires more steps to reduce the residual.* ■

3.3 Feature Extraction of Clustered Data

To achieve higher efficiency in manipulating the data represented in a high dimensional space, it is often necessary to reduce the dimension *dramatically*. In this section, several dimension reduction/feature extraction methods that preserve the cluster structure are discussed. The clustering of data are assumed to be performed already. Each method attempts to choose a projection to the reduced dimension that will capture the cluster structure of the data collection as well as possible. The dimension reduction is only a preprocessing stage and even if this stage is a little expensive, it may be worthwhile if it effectively reduces the cost of the postprocessing involved in classification and query processing, which will dominate computationally.

Figure 2: The upper two graphs show the coordinates of the right hand sides along the left singular vectors u_i . Cases 1 (top) and 2 (middle). The bottom graph shows the residual as a function of the number of PLS steps. The solid curve is for Case 1 and the dashed for Case 2.

Some experimental results are presented at the end of this chapter to illustrate the trade-off in effectiveness versus efficiency of the methods, so that their potential application can be evaluated.

Given a term-document matrix, the problem is to find a transformation that maps each document vector in the m -dimensional space to a vector in the k -dimensional space for some $k < m$. For this, either the dimension reducing transformation $T^T \in \mathbb{R}^{k \times m}$ is computed explicitly or the problem is formulated as that of approximation where the given matrix A is to be decomposed into two matrices B and Y as

$$A \approx BY \quad (32)$$

where $B \in \mathbb{R}^{m \times k}$ with $\text{rank}(B) = k$ and $Y \in \mathbb{R}^{k \times n}$ with $\text{rank}(Y) = k$. The matrix B is responsible for the dimension reducing transformation. However, often the dimension reducing transformation T^T itself is not explicitly needed since the dimension reduced representation Y can be computed without the transformation T^T . The lower rank approximation (32) is not unique since for any nonsingular matrix $Z \in \mathbb{R}^{k \times k}$,

$$A \approx BY = (BZ)(Z^{-1}Y),$$

where $\text{rank}(BZ) = k$ and $\text{rank}(Z^{-1}Y) = k$. The decomposition (32) is closely related to the rank reducing decomposition presented in (3).

The problem of approximate decomposition (32) can be recast in two different but related ways. The first is in terms of a matrix rank reduction formula and the second is as a minimization problem

$$\min_{B \in \mathbb{R}^{m \times k}, Y \in \mathbb{R}^{k \times n}} \|A - BY\|_p. \quad (33)$$

The incorporation of clustered structure can be translated into choosing the factors F and G in (7) or adding a constraint in the minimization problem (33). In the next section, ways to compute the factor Y are discussed where knowledge of the clusters from the full dimension is reflected in the dimension reduction.

3.3.1 Methods Based on Centroids

For simplicity of discussion, the columns a_j of A are assumed to be grouped into k clusters as

$$A = [\tilde{A}_1 \quad \tilde{A}_2 \quad \cdots \quad \tilde{A}_k] \quad \text{where} \quad \tilde{A}_i \in \mathbb{R}^{m \times n_i}, \quad \text{and} \quad \sum_{i=1}^k n_i = n. \quad (34)$$

Let N_i denote the set of column indices that belong to cluster i . The centroid $c^{(i)}$ of the i th cluster is the average of the columns in N_i , i.e.,

$$c^{(i)} = \frac{1}{n_i} \sum_{j \in N_i} a_j$$

and the global centroid c is defined as

$$c = \frac{1}{n} \sum_{j=1}^n a_j.$$

The centroid vector achieves the minimum variance since

$$\min_{j=1}^n \|a_j - c\|_2^2 = \min_{x \in \mathbb{R}^{n \times 1}} \sum_{j=1}^n \|a_j - x\|_2^2 = \min_{x \in \mathbb{R}^{n \times 1}} \|A - xe^T\|_F^2,$$

where $e = (1, \dots, 1)^T \in \mathbb{R}^{n \times 1}$. In the Centroid method for dimension reduction due to Park et.al. [34] and summarized in Algorithm 4, the columns of B in the minimization problem (33) are chosen as the centroids of the k clusters. Then the k dimensional representation Y of the matrix A is obtained by solving the least squares problem [5]

$$\min_{Y \in \mathbb{R}^{k \times n}} \|A - CY\|_F, \quad (35)$$

where C is the centroid matrix whose i th column is the centroid of the i th cluster. Note that the reduced dimension k is the same as the number of clusters. A k dimensional representation y of any data $q \in \mathbb{R}^{n \times 1}$ can be found from solving

$$\min_{y \in \mathbb{R}^{k \times 1}} \|q - Cy\|_2. \quad (36)$$

The Centroid dimension reduction method can be represented using the matrix rank reduction formula. Defining a grouping matrix $H \in \mathbb{R}^{n \times k}$ as

$$H = J \cdot (J^T J)^{-1} \quad (37)$$

where $J \in \mathbb{R}^{n \times k}$ and

$$J(i, j) = \begin{cases} 1 & \text{if document } i \text{ belongs to cluster } j, \\ 0 & \text{otherwise,} \end{cases}$$

Algorithm 4 Centroid dimension reduction method

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with k clusters, it computes a k -dimensional representation y of any given vector $q \in \mathbb{R}^{m \times 1}$.

1. Compute the centroid $c^{(i)}$ of the i th cluster, $1 \leq i \leq k$.
2. Set $C = [c^{(1)} \quad c^{(2)} \quad \dots \quad c^{(k)}]$.
3. Solve $\min_Y \|CY - q\|_2$

Algorithm 5 Orthogonal Centroid dimension reduction method

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with k clusters, it computes a k -dimensional representation y of any given vector $q \in \mathbb{R}^{m \times 1}$.

1. Compute the centroid $c^{(i)}$ of the i th cluster, $1 \leq i \leq k$.
2. Set $C = [c^{(1)} \quad c^{(2)} \quad \dots \quad c^{(k)}]$.
3. Compute the reduced QR decomposition of C , which is $C = Q_k R$.
4. Solve $\min_Y \|Q_k y - q\|_F$ (in fact, $y = Q_k^T q$).

the centroid matrix C can be written as $C = AH$. The solution for the problem (35) is $Y = (C^T C)^{-1} C^T A$, which in turn yields the matrix rank reduction expression

$$\begin{aligned} E &= A - CY \\ &= A - (AH)(C^T C)^{-1} C^T A \\ &= A - (AH)(H^T A^T AH)^{-1} (H^T A^T A). \end{aligned}$$

This shows that the rank reduction theorem 1.2 with $G = H^T A^T$ and $F = H$, produces the Centroid method.

In the Orthogonal Centroid method [34], an orthonormal basis Q_k for the centroid matrix C is chosen as the matrix B and the k dimensional representation of the matrix A is found by solving

$$\min_{Y \in \mathbb{R}^{k \times n}} \|A - Q_k Y\|_F. \quad (38)$$

Then a k dimensional representation of any data $q \in \mathbb{R}^{m \times 1}$ can be found from solving

$$\min_{y \in \mathbb{R}^{k \times 1}} \|q - Q_k y\|_F, \quad (39)$$

where the solution is $y = Q_k^T q$. The matrix Q_k can be obtained from the orthogonal decomposition of the matrix C when it has full rank. This Orthogonal Centroid method, summarized in Algorithm 5 provides a link between the methods of discriminant analysis and those based on the centroids. More detailed discussions can be found in [22].

The Orthogonal Centroid method is mathematically equivalent to the multiple group method known in applied statistics/psychometrics for more than 50 years. In his book on factor analysis [24], Horst attributes

the multiple group method to Thurstone [46]. For a description of the multiple group algorithm in modern numerical linear algebra terms, see [21].

3.3.2 Classical Linear Discriminant Analysis

The goal of linear discriminant analysis is to combine features of the original data in a way that most effectively discriminates between classes. Assuming that the given data are already clustered, a transformation $T^T \in \mathbb{R}^{k \times m}$ is to be found that optimally preserves this cluster structure in the reduced dimensional space. For this purpose, cluster quality needs to be measured. When cluster quality is high, each cluster is tightly grouped, but well separated from the other clusters. To quantify this, the within-cluster, between-cluster, and mixture scatter matrices are defined [13] as

$$\begin{aligned} S_w &= \sum_{i=1}^k \sum_{j \in N_i} (a_j - c^{(i)})(a_j - c^{(i)})^T, \\ S_b &= \sum_{i=1}^k \sum_{j \in N_i} (c^{(i)} - c)(c^{(i)} - c)^T = \sum_{i=1}^k n_i (c^{(i)} - c)(c^{(i)} - c)^T, \\ S_m &= \sum_{j=1}^n (a_j - c)(a_j - c)^T, \end{aligned}$$

respectively. The scatter matrices have the relationship [26]

$$S_m = S_w + S_b. \quad (40)$$

Since

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

measures the closeness of the columns within the clusters, and

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

measures the separation between clusters, an optimal transformation that preserves the given cluster structure would maximize $\text{trace}(T^T S_b T)$ and minimize $\text{trace}(T^T S_w T)$.

One of the most commonly used criteria in discriminant analysis is that of optimizing

$$J_1(T) = \text{trace}((T^T S_2 T)^{-1} (T^T S_1 T)), \quad (41)$$

where S_1 and S_2 are chosen from S_w , S_b , and S_m . When S_2 is assumed to be nonsingular, it is symmetric positive definite. According to results from the symmetric-definite generalized eigenvalue problem [16], there exists a nonsingular matrix $X \in \mathbb{R}^{m \times m}$ such that

$$X^T S_1 X = \Lambda = \text{diag}(\lambda_1 \dots \lambda_m) \quad \text{and} \quad X^T S_2 X = I_m,$$

where without loss of generality, the following can be assumed:

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

Since S_1 is positive semidefinite and $x_i^T S_1 x_i = \lambda_i$, each λ_i is nonnegative and only the largest $q = \text{rank}(S_1)$ λ_i 's are nonzero. Letting x_i denote the i th column of X ,

$$S_1 x_i = \lambda_i S_2 x_i, \quad (42)$$

which means that λ_i and x_i are an eigenvalue-eigenvector pair of $S_2^{-1} S_1$. Then

$$\begin{aligned} J_1(T) &= \text{trace}((T^T S_2 T)^{-1} (T^T S_1 T)) \\ &= \text{trace}((T^T X^{-T} X^{-1} T)^{-1} T^T X^{-T} \Lambda X^{-1} T) \\ &= \text{trace}((\tilde{T}^T \tilde{T})^{-1} \tilde{T}^T \Lambda \tilde{T}), \end{aligned}$$

where $\tilde{T} = X^{-1} T$. The matrix $\tilde{T}$ has full column rank provided T does, so it has the reduced QR factorization $\tilde{T} = QR$, where $Q \in \mathbb{R}^{m \times l}$ has orthonormal columns and R is nonsingular. Hence

$$\begin{aligned} J_1(T) &= \text{trace}((R^T R)^{-1} R^T Q^T \Lambda Q R) \\ &= \text{trace}(Q^T \Lambda Q). \end{aligned}$$

This shows that

$$\max_T J_1(T) = \max_{Q^T Q = I} \text{trace}(Q^T \Lambda Q) \leq \lambda_1 + \cdots + \lambda_q = \text{trace}(S_2^{-1} S_1).$$

(Here only maximization is considered. However, J_1 may need to be minimized for some other choices of S_1 and S_2 .) When the reduced dimension k is at least as large as q , i.e., $k \geq q$, this upper bound on $J_1(T)$ is achieved with

$$Q = \begin{pmatrix} I_k \\ 0 \end{pmatrix} \quad \text{or} \quad T = X \begin{pmatrix} I_k \\ 0 \end{pmatrix} R.$$

The transformation T is not unique. That is, J_1 satisfies the invariance property

$$J_1(T) = J_1(TW)$$

for any nonsingular matrix $W \in \mathbb{R}^{k \times k}$, since

$$\begin{aligned} J_1(TW) &= \text{trace}((W^T T^T S_2 T W)^{-1} (W^T T^T S_1 T W)) \\ &= J_1(T). \end{aligned}$$

Hence, the maximum $J_1(T)$ is also achieved with

$$T = X \begin{pmatrix} I_l \\ 0 \end{pmatrix},$$

and

$$\text{trace}((T^T S_2 T)^{-1} T^T S_1 T) = \text{trace}(S_2^{-1} S_1) \quad (43)$$

whenever $T \in \mathbb{R}^{m \times k}$ consists of k eigenvectors of $S_2^{-1} S_1$ corresponding to the l largest eigenvalues.

3.3.3 Improved Linear Discriminant Analysis and GSVD

Assume the cluster structure 34. Defining the $m \times n$ matrices,

$$H_w = [A_1 - c^{(1)}e^{(1)T}, A_2 - c^{(2)}e^{(2)T}, \dots, A_k - c^{(k)}e^{(k)T}], \quad (44)$$

$$H_b = [(c^{(1)} - c)e^{(1)T}, (c^{(2)} - c)e^{(2)T}, \dots, (c^{(k)} - c)e^{(k)T}], \quad (45)$$

$$H_m = [a_1 - c, \dots, a_n - c] = A - ce^T, \quad (46)$$

where $e^{(i)} = (1, \dots, 1)^T \in \mathbb{R}^{n_i \times 1}$, the scatter matrices can be expressed as

$$S_w = H_w H_w^T, \quad S_b = H_b H_b^T, \quad \text{and} \quad S_m = H_m H_m^T. \quad (47)$$

Now, a limitation of the J_1 criteria in many applications, including text processing in information retrieval, is that the matrix S_2 must be nonsingular. The matrix S_b is always singular since $\text{rank}(S_b) \leq k - 1$. The matrices S_w and S_m are singular in many applications: when the number of terms in the document collection is larger than the total number of documents (i.e., $m > n$ in the term-document matrix A) and when the data are represented in a high dimensional space but collecting many samples is expensive. However, a solution based on the GSVD does not impose this restriction, and can be found without explicitly forming S_1 and S_2 from H_w , H_b , and H_m . In addition, the improved LDA/GSVD method is numerically superior to the classical LDA which uses the generalized eigenvalue decomposition, when the matrices involved are ill conditioned [22]. The generalized singular value decomposition (GSVD) [47, 35, 16], as defined in the following theorem is used as a key tool.

THEOREM 3.3 (GENERALIZED SINGULAR VALUE DECOMPOSITION [35]) *For any two matrices $K_A \in \mathbb{R}^{n \times m}$ and $K_B \in \mathbb{R}^{p \times m}$ with the same number of columns, there exist orthogonal matrices $U \in \mathbb{R}^{n \times n}$, $V \in \mathbb{R}^{p \times p}$, $W \in \mathbb{R}^{t \times t}$, and $Q \in \mathbb{R}^{m \times m}$ such that*

$$U^T K_A Q = \Sigma_A (\underbrace{W^T R}_t, \underbrace{0}_{m-t}) \quad \text{and} \quad V^T K_B Q = \Sigma_B (\underbrace{W^T R}_t, \underbrace{0}_{m-t}),$$

where

$$K = \begin{pmatrix} K_A \\ K_B \end{pmatrix} \quad \text{and} \quad t = \text{rank}(K),$$

$$\Sigma_A = \begin{pmatrix} I_A & & \\ & D_A & \\ & & O_A \end{pmatrix}_{n \times t}, \quad \Sigma_B = \begin{pmatrix} O_B & & \\ & D_B & \\ & & I_B \end{pmatrix}_{p \times t},$$

and $R \in \mathbb{R}^{t \times t}$ is nonsingular with its singular values equal to the nonzero singular values of K . The matrices

$$I_A \in \mathbb{R}^{r \times r} \quad \text{and} \quad I_B \in \mathbb{R}^{(t-r-s) \times (t-r-s)}$$

are identity matrices, where

$$r = \text{rank} \begin{pmatrix} K_A \\ K_B \end{pmatrix} - \text{rank}(K_B) \quad \text{and} \quad s = \text{rank}(K_A) + \text{rank}(K_B) - \text{rank} \begin{pmatrix} K_A \\ K_B \end{pmatrix},$$

$$O_A \in \mathbb{R}^{(n-r-s) \times (t-r-s)} \quad \text{and} \quad O_B \in \mathbb{R}^{(p-t+r) \times r}$$

Algorithm 6 LDA/GSVD

Given a data matrix $A \in \mathbb{R}^{m \times n}$ with k clusters, it computes the columns of the matrix $T \in \mathbb{R}^{m \times (k-1)}$, which preserves the cluster structure in the reduced dimensional space, using

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

It also computes the $k - 1$ dimensional representation y of any given vector q .

1. Compute H_b and H_w from A according to

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

and (44), respectively.

2. Compute the complete orthogonal decomposition of

$$K = \begin{bmatrix} H_b^T \\ H_w^T \end{bmatrix} \in \mathbb{R}^{(k+n) \times m}, \text{ which is } P^T K Q = \begin{bmatrix} R & 0 \\ 0 & 0 \end{bmatrix}.$$

3. Let $t = \text{rank}(K)$.
4. Compute W from the SVD of $P(1 : k, 1 : t)$, which is $U^T P(1 : k, 1 : t) W = \Sigma_A$.

5. Compute the first $k - 1$ columns of $X = Q \begin{bmatrix} R^{-1} W & 0 \\ 0 & I \end{bmatrix}$, and assign them to T .

6. $y = T^T q$

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

$$D_A = \text{diag}(\alpha_{r+1}, \dots, \alpha_{r+s}) \quad \text{and} \quad D_B = \text{diag}(\beta_{r+1}, \dots, \beta_{r+s})$$

satisfy

$$1 > \alpha_{r+1} \geq \dots \geq \alpha_{r+s} > 0, \quad 0 < \beta_{r+1} \leq \dots \leq \beta_{r+s} < 1, \quad (48)$$

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

This form of GSVD is related to that of Van Loan [35] as

$$U^T K_A X = (\Sigma_A, 0) \quad \text{and} \quad V^T K_B X = (\Sigma_B, 0), \quad (49)$$

where $X = Q \begin{bmatrix} R^{-1} W & 0 \\ 0 & I \end{bmatrix}$ is a nonsingular matrix. This implies that

$$X^T K_A^T K_A X = \begin{bmatrix} \Sigma_A^T \Sigma_A & 0 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad X^T K_B^T K_B X = \begin{bmatrix} \Sigma_B^T \Sigma_B & 0 \\ 0 & 0 \end{bmatrix}.$$

Defining

$$\alpha_i = 1, \beta_i = 0 \text{ for } i = 1, \dots, r$$

and

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

one obtains, for $1 \leq i \leq t$,

$$\beta_i^2 K_A^T K_A x_i = \alpha_i^2 K_B^T K_B x_i, \quad (50)$$

where x_i represents the i th column of X . Since

$$K_A^T K_A x_i = 0 \quad \text{and} \quad K_B^T K_B x_i = 0$$

for x_i with $t + 1 \leq i \leq m$, the remaining $m - t$ columns of X , (50) is satisfied for arbitrary values of α_i and β_i when $t + 1 \leq i \leq m$. The columns of X are the generalized right singular vectors for the matrix pair (K_A, K_B) . 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.

Expressing λ_i as α_i^2/β_i^2 , (42) becomes

$$\beta_i^2 S_1 x_i = \alpha_i^2 S_2 x_i, \quad (51)$$

which has the form of a problem that can be solved using the GSVD.

Consider the case where

$$(S_1, S_2) = (S_b, S_w).$$

From (47) and the definition of H_b given in (45), $\text{rank}(S_b) \leq k - 1$. To approximate T that satisfies both

$$\max_T \text{trace}(T^T S_b T) \quad \text{and} \quad \min_T \text{trace}(T^T S_w T), \quad (52)$$

the x_i 's which correspond to the $k - 1$ largest λ_i 's are chosen, where $\lambda_i = \alpha_i^2/\beta_i^2$. When the GSVD construction orders the singular value pairs as in (48), the α_i/β_i quotients, are in nonincreasing order. Therefore, the first $k - 1$ columns of X are all that is needed. This method based on the GSVD is summarized in Algorithm LDA/GSVD [21]. In the algorithm, the matrices H_b and H_w from the data matrix A are first computed. Then a limited portion of the GSVD of the matrix pair (H_b^T, H_w^T) is computed. This solution is accomplished by following the construction in the proof of Theorem 3.3 [35]. The major steps are limited to the complete orthogonal decomposition [16, 33] of

$$K = \begin{pmatrix} H_b^T \\ H_w^T \end{pmatrix},$$

which produces orthogonal matrices P and Q and a nonsingular matrix R , followed by the singular value decomposition of a leading principal submatrix of P .

When $m > n$, the scatter matrix S_w is singular. Hence, the J_1 criterion cannot be defined, and discriminant analysis fails. Consider a generalized right singular vector x_i that lies in the null space of S_w . From Eqn. (51), it is seen that either x_i also lies in the null space of S_b , or the corresponding β_i equals zero.

Algorithm 7 Centroid-based Classification

Given a data matrix A with k clusters and k corresponding centroids, $c^{(i)}$ for $1 \leq i \leq k$, it finds the index j of the cluster to which a vector q belongs.

- find the index j such that $\text{sim}(q, c^{(i)})$, $1 \leq i \leq k$, is minimum (or maximum), where $\text{sim}(q, c^{(i)})$ is the similarity measure between q and $c^{(i)}$.

(For example, $\text{sim}(q, c^{(i)}) = \|q - c^{(i)}\|_2$ using the L_2 norm, and the index with the minimum value is taken. Using the cosine measure, $\text{sim}(q, c^{(i)}) = \cos(q, c^{(i)}) = \frac{q^T c^{(i)}}{\|q\|_2 \|c^{(i)}\|_2}$, and the index with the maximum value is taken.)

Algorithm 8 k Nearest Neighbor (knn) Classification

Given a data matrix $A = [a_1 \ \dots \ a_n]$ with k clusters, it finds the cluster to which a vector q belongs.

1. From the similarity measure $\text{sim}(q, a_j)$ for $1 \leq j \leq n$, find the k nearest neighbors of q . (k is used to distinguish the algorithm parameter from the number of clusters k .)
2. Among these k vectors, count the number belonging to each cluster.
3. Assign q to the cluster with the greatest count in the previous step.

When $x_i \in \text{null}(S_w) \cap \text{null}(S_b)$, Eqn. (51) is satisfied for arbitrary values of α_i and β_i . This will be the case for the rightmost $m - t$ columns of X . To determine whether these columns should be included in T , consider

$$\text{trace}(T^T S_b T) = t_j^T S_b t_j \quad \text{and} \quad \text{trace}(T^T S_w T) = t_j^T S_w t_j,$$

where t_j represents the j th column of T . Since $x_i^T S_w x_i = 0$ and $x_i^T S_b x_i = 0$, adding the column x_i to T does not contribute to either maximization or minimization in (52). For this reason, these columns of X are not included in the solution.

When $x_i \in \text{null}(S_w) - \text{null}(S_b)$, then $\beta_i = 0$. This implies that $\alpha_i = 1$, and hence that the generalized singular value α_i/β_i is infinite. The leftmost columns of X will correspond to these. Including these columns in T increases $\text{trace}(T^T S_b T)$, while leaving $\text{trace}(T^T S_w T)$ unchanged. It follows that, even when S_w is singular, the rule regarding which columns of X to include in T remains the same as for the nonsingular case. The experiments in Section 3.4 demonstrate that Algorithm LDA/GSVD works well even when S_w is singular, thus extending its applicability beyond that of classical discriminant analysis, and producing more accurate solutions even when S_w is nonsingular and thus the classical LDA can be applied. For equivalence proof of the solutions from various choices of (S_1, S_2) , see [22].

3.4 Experiments in Classification

In this section, the effectiveness of the three dimension reduction methods, Centroid, Orthogonal Centroid, and LDA/GSVD algorithms, are illustrated using experimental results. For LDA/GSVD, its mathematical equivalence to J_1 using an alternative choice of (S_1, S_2) are confirmed. For Orthogonal Centroid, its preser-

Table 1: Trace values and self-classification accuracy (in %) with L_2 norm similarity on a 150×2000 artificially generated data set with 7 clusters.

Method	Full	Orthogonal Centroid	Centroid	max $\text{trace}(S_w^{-1}S_b)$	max $\text{trace}(S_w^{-1}S_m)$
Dim	150×2000	7×2000	7×2000	6×2000	6×2000
$\text{trace}(S_w)$	299705	13885	907.75	1.97	1.48
$\text{trace}(S_b)$	22925	22925	1706.8	4.03	3.04
$\text{trace}(S_m)$	322630	36810	2614.55	6.00	4.52
$\text{trace}(S_w^{-1}S_b)$	12.6	-	-	12.6	12.6
$\text{trace}(S_w^{-1}S_m)$	162.6	-	-	18.6	18.6
centroid	97.4 %	97.4 %	97.1 %	97.8 %	98.0 %
5nn	81.3 %	96.7%	96.8 %	97.8 %	97.8 %
15nn	89.9 %	97.3 %	97.2 %	98.2 %	98.1 %

variation of $\text{trace}(S_b)$ is shown to be a very effective compromise for the simultaneous optimization of two traces approximated by J_1 .

In Tables 1 and 2, a clustered data set generated by an algorithm adapted from [26, Appendix H] was used. The data consist of 2000 vectors in a space of dimension 150, with $k = 7$ clusters and these data are reclassified in the full space and in the reduced dimensional spaces in Table 1. The Centroid and Orthogonal Centroid methods reduce the dimension from 150 to $k = 7$, and the LDA/GSVD reduces the dimension $k - 1 = 6$. In Table 1, the LDA/GSVD criterion, $J_1 = \max \text{trace}(S_w^{-1}S_b)$ was compared with the alternative J_1 criterion, $\max \text{trace}(S_w^{-1}S_m)$. The trace values confirm the theoretical finding that the generalized eigenvectors that optimize the alternative J_1 also optimize LDA/GSVD's J_1 . The values for $\text{trace}(S_w^{-1}S_b)$ and $\text{trace}(S_w^{-1}S_m)$ were not computed for the centroid-based dimension reduction methods.

The classification accuracies for a centroid-based classification method [21] and the k nearest neighbor (knn) classification method [45] (see Algorithms 7 and 8), are presented. Note that the classification parameter k of knn differs from the number of clusters k . These are obtained using the L_2 norm similarity measure. In Table 2, the results with the same data are shown except that now the 2000 data items are randomly split into 1002 training data items and 998 test data items.

Another set of experiments validates the effectiveness of the dimension reduction methods and the extension of J_1 to the singular case. For this purpose, five categories of abstracts from the MEDLINE database [22] was used. Each category has 40 documents. There are 7519 terms after preprocessing with stemming and removal of stop words [29]. Since 7519 exceeds the number of documents (200), S_w is singular and classical discriminant analysis breaks down. However, the LDA/GSVD method circumvents this singularity problem.

The centroid-based dimension reduction algorithms dramatically reduce the dimension 7519 to 5, and the LDA/GSVD reduces the dimension 7519 to 4, which is one less than the number of clusters. Table 4 shows classification results using the L_2 norm similarity measure. For the top half of the table where the full problem size is 7519×200 , all 200 data items were used for dimension reduction, and the same data items are reclassified in the full space and in the reduced space. Then 200 data points were divided into 100 training data points that generate the dimension reducing transformations, and 100 test data points that were used in prediction. The bottom half of the table shows the prediction accuracy for the test data.

Table 2: Trace values and classification accuracy (in %) with L_2 norm similarity. The data set is the same as that for Table 1 except that out of 2000 data points, 1002 were used for training and 998 as test data.

Method	Full	Orthogonal Centroid	Centroid	LDA/GSVD
Dim	150×998	7×998	7×998	6×998
$\text{trace}(S_w)$	149821	6918.1	436.25	1.81
$\text{trace}(S_b)$	12025	12025	855.15	4.19
$\text{trace}(S_w^{-1}S_b)$	14.358	12.226	12.226	14.358
centroid	95.8 %	95.9%	96.0%	94.1 %
5nn	75.6 %	95.6%	95.6%	94.4 %
15nn	87.3 %	96.3%	96.5%	94.3 %
50nn	92.4 %	96.1%	96.1%	93.8 %

Table 3: MEDLINE data set, clusters and the number of documents from each cluster

class	category	no. of documents
1	heart attack	40
2	colon cancer	40
3	diabetes	40
4	oral cancer	40
5	tooth decay	40
	dimension	7519×200

Because the J_1 criterion is not defined in this case due to the fact that S_w is singular, the ratio $\frac{\text{trace}(S_b)}{\text{trace}(S_w)}$ is computed as a rough optimality measure. The ratio is much higher for LDA/GSVD reduction than for the other methods. These experimental results confirm that the LDA/GSVD algorithm effectively extends the applicability of the J_1 criterion to cases that classical discriminant analysis cannot handle. In addition, the Orthogonal Centroid algorithm preserves $\text{trace}(S_b)$ from the full dimension without the expense of computing eigenvectors. Taken together, the results for these methods demonstrate the potential for dramatic and efficient dimension reduction without compromising cluster structure.

The experimental results verify that the J_1 criterion, when applicable, effectively optimizes classification in the reduced dimensional space, while the LDA/GSVD extends the applicability to cases that classical discriminant analysis cannot handle. In addition, the LDA/GSVD algorithm avoids the numerical problems inherent in explicitly forming the scatter matrices.

In terms of computational complexity, the most costly part of Algorithm LDA/GSVD is Step 2, where a complete orthogonal decomposition is needed, and a fast algorithm needs to be developed for Step 2. For Orthogonal Centroid, the most expensive step is the reduced QR decomposition of C [16]. By solving a simpler eigenvalue problem and avoiding the computation of eigenvectors, Orthogonal Centroid is significantly cheaper than LDA/GSVD. The experiments show it to be a very reasonable compromise.

Finally, it bears repeating that dimension reduction is only a preprocessing stage. Since classification

Table 4: Trace values and classification accuracy (in %) with L_2 norm similarity on the 7519×200 MED-LINE data. The top half of the table shows the self-classification accuracy, where all 200 data points were used in obtaining the dimension reducing transformations and then reclassified. For the bottom half, 100 data items were used for obtaining the dimension reducing transformations, and the other 100 points were used as a test data set to obtain prediction accuracy.

Method	Full	Orthogonal Centroid	Centroid	LDA/GSVD
Dim	7519×200	5×200	5×200	4×200
$\text{trace}(S_w)$	73048	4210.1	90.808	0.05
$\text{trace}(S_b)$	6228.7	6228.7	160.0	3.95
$\text{trace}(S_b)/\text{trace}(S_w)$	0.0853	1.4795	1.7620	79
centroid	95.0 %	95.0 %	98.0 %	99.0 %
1nn	58.5 %	96.0 %	96.5 %	98.0 %
Dim	7519×100	5×100	5×100	4×100
$\text{trace}(S_w)$	36599	2538.5	39.813	6e-16
$\text{trace}(S_b)$	4329.6	4329.3	80.0	4.0
$\text{trace}(S_b)/\text{trace}(S_w)$	0.1183	1.7056	2.0094	6.7e15
centroid	77.0 %	77.0%	90.0%	87.0 %
1nn	39.0 %	87.0%	85.0%	87.0 %

and document retrieval will be the dominating parts computationally, the expense of dimension reduction should be weighed against its effectiveness in reducing the cost involved in those processes.

3.5 Two Dimensional Visualization of Data

Using the same data as in Section 3.4, the 2D visualization capability of some dimension reduction methods for clustered data is shown in this section. To illustrate the discriminatory power of LDA/GSVD, the J_1 criterion was used to reduce the dimension from 150 to two. Even though the optimal reduced dimension is six, J_1 does surprisingly well at discriminating among seven classes, as seen in Figure 3. As expected, the alternative J_1 does equally well in Figure 3. In contrast, Figure 3 shows that the truncated SVD is not the best discriminatory for clustered data sets.

Acknowledgement

The authors are grateful to Gene Golub for helpful discussions on the Wedderburn rank reduction formula, and to Jimmy Wang for the implementation results presented in the tables.

References

- [1] M. R. Anderberg. Cluster analysis for applications. *Academic Press, New York and London*, 1973.

Figure 3: Two dimensional representation of 150×2000 data with seven clusters. The first graph is obtained based on the transformation T from $\max \text{trace}(T^T S_w^{-1} S_b T)$, the second by $\max \text{trace}(T^T S_w^{-1} S_m T)$, and the third by using $\Sigma_2 V_2^T$ from truncated SVD.

- [2] E. Anderson, Z. Bai, C. H. Bischof, J. W. Demmel, J. J. Dongarra, J. J. Du Croz, A. Greenbaum, S. J. Hammarling, A. McKenney, S. Ostrouchov, and D. C. Sorensen. *LAPACK Users' Guide, Release 2.0*. Soc. for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 1995.
- [3] M.W. Berry, S.T. Dumais, and G.W. O'Brien. Using linear algebra for intelligent information retrieval. *SIAM Review*, 37(4):573–595, 1995.
- [4] M.W. Berry and M. Browne. *Understanding Search Engines. Mathematical Modeling and Text Retrieval*. SIAM, Philadelphia, PA, 1999.
- [5] A. Björck. *Numerical Methods for Least Squares Problems*. SIAM, Philadelphia, PA, 1996.
- [6] M.T. Chu, R.E. Funderlic, and G.H. Golub. A rank-one reduction formula and its applications to matrix factorization. *SIAM Review*, 37:512–530, 1995.
- [7] R.E. Cline and R.E. Funderlic. The rank of a difference of matrices and associated generalized inverses. *Linear Algebra Appl.*, 24:185–215, 1979.
- [8] 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*, 41:391–407, 1990.
- [9] C. Eckart and G. Young. The approximation of one matrix by another lower rank. *Psychometrika*, 1:211–218, 1936.
- [10] L. Eldén. Partial least squares vs. Lanczos bidiagonalization I: Analysis of a projection method for multiple regression. Technical Report LiTH-MAT-R-2002-24, Linköping University, Department of Mathematics, 2002.
- [11] W.B. Frakes and R. Baeza-Yates. *Information Retrieval: Data Structures and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey, 1992.
- [12] I.E. Frank and J.H. Friedman. A statistical view of some chemometrics regression tools. *Technometrics*, 35:109–135, 1993.
- [13] K. Fukunaga. *Introduction to Statistical Pattern Recognition*. Academic Press, second edition, 1990.
- [14] G.H. Golub. Numerical methods for solving least squares problems. *Numerische Mathematik*, 7:206–216, 1965.
- [15] G. H. Golub and W. Kahan. Calculating the singular values and pseudo-inverse of a matrix. *SIAM J. Numer. Anal. Ser. B*, 2:205–224, 1965.
- [16] G.H. Golub and C.F. Van Loan. *Matrix Computations*, third edition. Johns Hopkins University Press, Baltimore, 1996.
- [17] M.D. Gordon. Using Latent Semantic Indexing for Literature Based Discovery. *J. of the American Society for Information Science*, 49(8):674–685, 1998.
- [18] E. Gose, R. Johnsonbaugh and S. Jost. *Pattern Recognition and Image Analysis*. Prentice Hall, Upper Saddle River, New Jersey, 1996.

- [19] L. Guttman. A necessary and sufficient formula for matrix factoring. *Psychometrika*, 22:79-81, 1957.
- [20] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning. Data mining, Inference and Prediction*. Springer Verlag, New York, 2001.
- [21] 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.
- [22] P. Howland and H. Park. Generalization of discriminant analysis based on the generalized singular value decomposition, submitted for a publication, January, 2003.
- [23] I.S. Helland. On the structure of partial least squares regression. *Commun. Statist. Simulation*, 17:581–607, 1988.
- [24] P. Horst. *Factor Analysis of Data Matrices*. Holt, Rinehart and Winston, Inc., 1965.
- [25] L. Hubert, J. Meulman, and W. Heiser. Two purposes for matrix factorization: A historical appraisal. *SIAM Review*, 42:68–82, 2000.
- [26] A.K. Jain and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [27] Y. Jung, H. Park, and D. Du. A Balanced Term-Weighting Scheme for Improved Document Comparison and Classification, *preprint*, 2001.
- [28] W. Kahan. Numerical linear algebra. *Canad. Math. Bull.*, 9:757–801, 1966.
- [29] G. Kowalski. *Information Retrieval Systems : Theory and Implementation*. Kluwer Academic Publishers, 1997.
- [30] H. Kim, P. Howland, and H. Park. Text Categorization using Support Vector Machines with Dimension Reduction, in preparation.
- [31] A.K. Jain, and R.C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, 1988.
- [32] I. Joliffe. *Principal Component Analysis*. Springer, New York, NY, 1986.
- [33] C. L. Lawson and R. J. Hanson. *Solving Least Squares Problems*. SIAM, Classics in Applied Mathematics, Philadelphia, 1995.
- [34] H. Park, M. Jeon, and J.B. Rosen. Lower dimensional representation of text data based on centroids and least squares, *BIT*, to appear.
- [35] C.C. Paige and M.A. Saunders. Towards a generalized singular value decomposition. *SIAM J. Numer. Anal.*, 18(3):398–405, 1981.
- [36] A.M. Pejtersen. Semantic Information Retrieval. *Communications of the ACM*, 41(4):90-92, 1998.
- [37] J.B. Rosen, H. Park, and J. Glick. Total least norm formulation and solution for structured problems. *SIAM Journal on Matrix Anal. Appl.*, 17:110-128, 1996.
- [38] G. Salton. The SMART retrieval system, *Prentice Hall*, 1971.

- [39] G. Salton, and M.J. McGill. Introduction to Modern Information Retrieval, *McGraw-Hill*, 1983.
- [40] B. Savas. Analyses and tests of handwritten digit recognition algorithms. LiTH-MAT-EX-2003-01, Linköping University, Department of Mathematics, 2003.
- [41] G.W. Stewart. An updating algorithm for subspace tracking. *IEEE Trans. Signal Proc.* 40(1992), 1535–1541.
- [42] G.W. Stewart. Updating a rank-revealing ULV decomposition. *SIAM J. Matrix Anal. Appl.* 14(1993), 494–499.
- [43] G.W. Stewart. The QLP approximation to the singular value decomposition,
- [44] O. Svensson, T. Kourti, and J.F. MacGregor. An investigation of orthogonal signal correction algorithms and their characteristics. *J. Chemometrics*, 16:176–188, 2002.
- [45] S. Theodoridis and K. Koutroumbas. Pattern Recognition, *Academic Press*, 1999
- [46] L.L. Thurstone. A multiple group method of factoring the correlation matrix. *Psychometrika*, 10(2):73–78, 1945.
- [47] C.F. Van Loan. Generalizing the singular value decomposition. *SIAM J. Numer. Anal.*, 13(1):76–83, 1976.
- [48] J.H.M Wedderburn. *Lectures on Matrices*. Colloquium Publications. Amer. Math. Soc, New York, 1934.
- [49] H. Wold. Soft modeling by latent variables; the nonlinear iterative partial least squares approach. In J. Gani, editor, *Perspectives in Probability and Statistics, Papers in honour of M.S. Bartlett*, chapter London. Academic Press, 1975.
- [50] S. Wold, A. Ruhe, H. Wold, and W.J. Dunn. The collinearity problem in linear regression. the partial least squares (PLS) approach to generalized inverses. *SIAM J. Sci. Stat. Comput.*, 5:735–743, 1984.
- [51] S. Wold, M. Sjöström, and L. Eriksson. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intell. Lab. Systems*, 58:109–130, 2001.